# Hop-Constrained Oblivious Routing

Mohsen Ghaffari
ETH Zurich
Switzerland
ghaffari@inf.ethz.ch

Bernhard Haeupler
Carnegie Mellon University & ETH Zurich
USA & Switzerland
haeupler@cs.cmu.edu

Goran Zuzic
ETH Zurich
Switzerland
goran.zuzic@inf.ethz.ch

## ABSTRACT

We prove the existence of an oblivious routing scheme that is poly$(\log n)$-competitive in terms of $(congestion + dilation)$, thus resolving a well-known question in oblivious routing.

Concretely, consider an undirected network and a set of packets each with its own source and destination. The objective is to choose a path for each packet, from its source to its destination, so as to minimize $(congestion+dilation)$, defined as follows: The dilation is the maximum path hop-length, and the congestion is the maximum number of paths that include any single edge. The routing scheme obliviously and randomly selects a path for each packet independent of (the existence of) the other packets. Despite this obliviousness, the selected paths have $(congestion+dilation)$ within a poly$(\log n)$ factor of the best possible value. More precisely, for any integer hop-bound $h$, this oblivious routing scheme selects paths of length at most $h \cdot$ poly$(\log n)$ and is poly$(\log n)$-competitive in terms of $congestion$ in comparison to the best possible $congestion$ achievable via paths of length at most $h$ hops. These paths can be sampled in polynomial time.

This result can be viewed as an analogue of the celebrated oblivious routing results of Räcke [FOCS 2002, STOC 2008], which are $O(\log n)$-competitive in terms of $congestion$, but are not competitive in terms of $dilation$.

## CCS CONCEPTS

• **Mathematics of computing → Graph theory**; *Network flows*; • **Theory of computation → Routing and network design problems**; Random projections and metric embeddings.

## KEYWORDS

hop-constrained oblivious routing, oblivious routing, partial tree embeddings, congestion, dilation, hop constraints

**ACM Reference Format:**
Mohsen Ghaffari, Bernhard Haeupler, and Goran Zuzic. 2021. Hop-Constrained Oblivious Routing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3406325.3451098

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
*STOC '21, June 21–25, 2021, Virtual, Italy*
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8053-9/21/06…$15.00
https://doi.org/10.1145/3406325.3451098

## 1 INTRODUCTION

Routing packets in computer networks is a fundamental task and a widely studied problem. Consider the following prototypical scenario:

*The network is abstracted as an $n$-node undirected graph. Each edge $e$ has the capacity to transfer $c_e$ packet per time unit, and each packet traversing each edge takes one time unit. The network receives a number of packet delivery requests, where the $i^{th}$ packet should be transmitted from source $s_i$ to destination $t_i$. The objective is to minimize the packet delivery completion time, i.e., to deliver all the packets to their destinations in the shortest span of time possible.*

This paper's contribution can be informally summarized as follows: we present the first *oblivious routing* scheme that is *competitive* in *completion time*. The scheme is oblivious in the sense that for each packet delivery request $i$, the path chosen for this packet[1] is decided independent of all the other packets $j \neq i$. This obliviousness property is strongly desirable in numerous networking settings, where packet delivery requests arrive at various points in the network and their routing has to be determined without any central control of the state of the network. The competitiveness guarantee is that, albeit this obliviousness restriction, all the packets are delivered to their destinations within a time that is at most a poly$(\log n)$ factor larger than the optimal time that is needed to deliver all the packets, i.e., the completion time in the fastest possible way to deliver all the packets.

In what follows, we describe this contribution in a more formal manner, putting it in the context of what has been known about packet routing algorithms in networks and especially the prior results on oblivious routing.

### 1.1 Background on Routing

*Route selection and scheduling:* The problem of routing a set of packets from their sources $s$ to destinations $t$ while minimizing the completion time involves two components: (I) *route selection*, i.e., choosing the path $p_{s,t}$ along which the packet is transferred from $s$ to $t$, (II) *scheduling* the timing of the packet traversing this path $p_{s,t}$, i.e., at which time unit the packet goes through each edge $e \in p_{s,t}$. A celebrated result of Leighton, Maggs, and Rao [16] shows that one can decouple these two issues with only a moderate loss—once the routes are selected, we can solve scheduling nearly optimally. Let us make this more precise. We focus on the setting where all edge capacities are uniform (e.g., by replacing higher capacity edges

[1] We note that besides this path selection obliviousness, even the timing schedule of how the packet traverses this path is essentially independent of all other packets in the following sense: we can break time into phases each involving $\Theta(\log n)$ time units, and the phase number in which the packet traverses through any edge on its chosen path is independent of all the other packets.

with edge multiplicities). Suppose that the routes are selected and consider a given set of paths $\{p_{s_i,t_i}\}_i$, one path for each packet. Let dilation denote the length of the longest path among these. Also, let the congestion of each edge $e$ be the number of paths that include it and let the overall congestion denote the maximum congestion over all edges $e$. While the best completion time depends on the exact model of how are the packets coordinated (e.g., centralized scheduling of packets vs. distributed scheduling), in these settings the completion time is near-optimally characterized by the quantity congestion + dilation. Clearly, delivering the packets along these paths requires at least $\max\{\text{congestion}, \text{dilation}\} \geq (\text{congestion} + \text{dilation})/2$ time units (i.e., the completion time is at least $(\text{congestion} + \text{dilation})/2$). Leighton, Maggs, and Rao show a centralized scheduling algorithm that would deliver all the packets to their destinations with completion time of $O(\text{congestion} + \text{dilation})$ time units. Moreover, just using their basic *random delays* idea, we can define a schedule that can implemented in a distributed setting with a near-optimal completion time of at most $O(\log n) \cdot (\text{congestion} + \text{dilation})$. Thanks to this simple *random delays* idea, if we ignore logarithmic factors, we can the routing question routing question boils down to the route selection problem while minimizing congestion + dilation, which is the question we focus on the in remainder of the paper.

*Oblivious route selection:* In the oblivious case, we are given a set of sources and destinations $\mathcal{D} = \{(s_i, t_i)\}_i$, which we call the **demand**. If the demand is known in advance (i.e., all $\{s_i\}_i$ and $\{t_i\}_i$ are known), the route selection problem can be solved in polynomial time, giving a set of routes that has congestion + dilation within a constant factor of the optimum, by a classic result of Srinivasan and Teo [22]. Put together with the aforementioned scheduling result of Leighton et al. [16], this gives a constant approximation algorithm for the completion time in packet routing. However, in this scheme, the routes selected by different requests heavily depend on each other, and devising these routes requires central control of the entire network. A much more common scenario in networking is that the packet delivery requests arrive at various points in the network. It is much more desirable if one can select the route of each packet just based on its source and destination, and in a manner *oblivious* to all the other packet routing requests. More formally, a (probabilistic) *routing scheme* can be summarized as follows.

*Definition 1.1.* A **routing scheme** $R$ for an undirected graph $G = (V, E)$ is a collection of $|V|^2$ distributions $R = \{R_{u,v}\}_{u,v \in V}$, where for each pair of nodes $u, v \in V$, we have one distribution $R_{u,v}$ over paths between $u$ and $v$.

A routing scheme can be used to *obliviously* route requests $\{(s_i, t_i)\}$ in the following straightforward way: Given a routing scheme $R$, the $i^{th}$ path is independently sampled from $R_{s_i,t_i}$. Note that each request is routed independently of (the existence of) other requests, hence the routing is *oblivious*.

*Quality measures:* Our goal is to find routing schemes which, for every demand, guarantee that the obliviously selected paths are competitive with the optimal (demand-dependent) set of paths in terms of some quality measure. We can measure the quality of the selected paths using various functions, including the maximum or average congestion, $\ell_p$ norm of edge congestion, the maximum dilation, etc. Given the discussions above, our primary measure of interest will be the summation congestion + dilation. As noted before, thanks to the random delays technique for scheduling [16], a routing scheme that has polylogarithmic competitiveness in terms of the congestion + dilation measure provides a routing scheme that has polylogarithmic competitiveness in terms of the completion time to deliver all packets.

## 1.2 Prior Work on Oblivious Routing

We next discuss the prior work on routing schemes that are obliviously competitive for other measures, and some of the known obstacles towards being competitive in congestion + dilation.

*Results on special graphs:* Valiant and Brebner [24] were the first to study oblivious packet routing. They focused on the case where the network is a hypercube and showed that any permutation can be routed with completion time $O(\log n)$. Their path selection is based on the "Valiant's trick" of routing from the source $s_i$ to a random node $q$ and from there to the destination $t_i$, where the paths from $s_i$ to $q$ and from $q$ to $t_i$ are greedy (fixing differing dimensions one by one). Following them, there have been a number of routing schemes that are obliviously competitive in terms of congestion + dilation in a range of special graphs, including expanders, Caley graphs, fat trees, meshes, etc. [7–9, 17, 21, 23] See the thesis of Scheideler [21] and the survey of Räcke [20] for more on related work.

*Congestion-competitive oblivious routing:* A prominent highlight in prior work is *congestion-competitive* oblivious routing, a topic which was initiated by Räcke's seminal paper [18], and through a beautiful line of work [3, 6, 15], culminated in the following celebrated result of Räcke's [19]: For every undirected graph, there is a polynomial-time algorithm to build a routing scheme such that for every demand $\mathcal{D}$, the routing obliviously produces a collection of routes which are $O(\log n)$-competitive in terms of congestion, compared to the optimal collection of (demand-dependent) routes for $\mathcal{D}$.

Theorem 1.2 (Räcke [19]). *For every undirected (multi)-graph $G = (V, E)$, there exists a routing scheme $R = R(G)$ such that for every demand $\mathcal{D} = \{(s_i, t_i)\}_{i=1}^k$, the maximum expected congestion of routing $\mathcal{D}$ using $R$ is at most a $O(\log n)$-factor larger than the optimal congestion of $\mathcal{D}$ in $G$.*

Räcke's routing scheme has the additional property of being *tree-based* (see Section 3.1 for a formal definition). Moreover, the current state-of-the-art approaches are all based on tree-based routing schemes, which greatly simplify the process of constructing competitive oblivious routings. A significant challenge this paper needed to overcome is the fact that tree-based routing schemes do not exist in our setting of jointly minimizing the congestion and dilation. To this end, we develop a theory of constructing routings using *partial trees* which allow for greater flexibility at the cost of increased intricacy of the construction.

*Oblivious routing for congestion and dilation:* Considering that both congestion and dilation impact packet delivery, it would be very desirable to be competitive in both, or just their summation. Unfortunately, it is well-known that Räcke's routing scheme is not competitive in terms of dilation and it can select paths that are

arbitrarily longer than the paths in the optimal collection. Because of this, while the routing is competitive in the congestion measure, it is not competitive in the completion time measure, or more concretely in terms of congestion + dilation. Of course, if we focus on dilation alone, it is trivial to be competitive by simply routing each packet along the shortest path between its source and destination.

*Oblivious routing for other measures.* Gupta, Hajiaghayi, and Räcke [11] study a range of rather general measures for oblivious routing. Suppose for each commodity $i \in \{1, ..., k\}$, we want a (fractional) flow from source $s_i$ to destination $t_i$. For a given set of flows for the commodities, for each edge $e$, let us use $f_i(e)$ to denote the amount of commodity-$i$ flow passed through $e$. Consider a *load function* $L : \mathbb{R}_+^k \to \mathbb{R}_+$ where $\ell(f_1(e), \ldots, f_k(e))$ defines the *load* of edge $e$. Gupta et al. provide two results: (A) an $O(\log^2 n)$ competitive oblivious routing algorithm for the summation of loads of different edges, assuming that the load function $\ell$ is the class of monotone sub-additive functions, (B) an $O(\log^2 n \log\log n)$ competitive oblivious algorithm for the maximum of the loads of different edges assuming that the load function is a norm. In either case, their oblivious routing does not need to know the load function $\ell$. Englert and Räcke [11] extended this framework and presented an $O(\log n)$-competitive oblivious algorithm for the case that the load function is a monotone norm, and we take the $\ell_p$ norm of the loads of different edges as our measure for competitiveness.

*An impossibility?* The goal of being competitive in both dilation and congestion has been discussed in the literature of oblivious routing as an ultimate goal [2, 20]. However, the discussion often soon concludes in an "impossibility": Suppose that we interpret the goal as being competitive in terms of congestion and also in terms of dilation, simultaneously. This is not possible. Consider two neighboring nodes $u$ and $v$ that, besides the edge between them, are also connected with $\Theta(\sqrt{n})$ disjoint paths of length $\Theta(\sqrt{n})$. If there are $\sqrt{n}$ that should go from $u$ to $v$, to be poly$(\log n)$-competitive in congestion, at least $\sqrt{n} - \text{poly}(\log n)$ of packets should be routed through those long paths. In an obliviously-competitive routing scheme, that means each packet should be routed through that long path with probability $1 - o(1)$. But then, if we consider a demand with just one packet, the optimal dilation is simply 1, while this routing scheme will have dilation $= \Theta(\sqrt{n})$ with probability $1-o(1)$. We note that this example also shows that, even if we make the summation congestion + dilation as our measure for competitiveness, poly$(\log n)$-competitive oblivious routing appears impossible. Indeed, noting this apparent impossibility for general graphs, Aspnes et al. [2] asked for a workaround in their list of prominent open problems in distributed computing. Their suggestion was that it might be still feasible for special networks: "*Another important open problem is to find classes of networks in which oblivious routing gives C+D [i.e.,* congestion + dilation*] close to the off-line optimal... Such a result have immediate consequences in packet scheduling algorithms.*"

Our aim in this paper is to have a solution for *all* graphs. This calls for re-examining the above impossibility argument. The astute reader notices that, in this simple example, changing the requirement slightly makes the problem possible and still perfectly useful: if we are given a target dilation upper bound $h$, which is within poly$(\log n)$ factor of the optimal dilation for the input instance, then the routing scheme can select the routes so as to remain below this target $h$ and still be poly$(\log n)$-competitive in terms of the congestion. Similarly, when congestion + dilation is the measure, given a poly$(\log n)$ factor upper bound on the value of the optimum, we can get a poly$(\log n)$-competitive oblivious routing in terms of congestion + dilation. Of course, in these definitions, the routes may depend on the given dilation upper bound, or the upper bound on congestion + dilation. The assumption of having these upper bounds can be removed by standard ideas such as guessing and doubling. Furthermore, this choice also has the added flexibility of exploring the optimal trade-off between feasible values for congestion and dilation. We note that even though in this simple example the problem becomes possible and easy, achieving such a result for general graphs is far from trivial. Indeed, since we can set the upper bound arbitrarily high, this is a strictly stronger requirement than requirement competitiveness in congestion alone (as in results of [18, 19] mentioned above). This is exactly the goal that we achieve in this paper.

## 1.3 Our Contribution

We show a routing scheme that, given a dilation bound $h$, is obliviously poly$(\log n)$-competitive in terms of congestion compared to the best congestion achievable via paths of length $h$, and our routing scheme uses paths of length at most $h \cdot \text{poly}(\log n)$.

Theorem 1.3. ***(Informal)*** *For every undirected (multi)-graph $G = (V, E)$ and every dilation bound $h \geq 1$, there exists a routing scheme $R = R(G, h)$ such that for every demand $\mathcal{D} = \{(s_i, t_i)\}_{i=1}^k$, the selected paths have a hop-length of at most $h \cdot O(\log^7 n)$, and in which the expected maximum* congestion *is within an $O(\log^2 n \cdot \log^2(h \log n))$ factor of the optimum* congestion *when routing along paths of length at most $h$.*

See Theorem 3.1 for the formal statement. Theorem 1.3 directly gives us a way of obtaining a routing scheme that is obliviously competitive in terms of the completion time. As explained above, thanks to known scheduling results, we can simply focus on the summation congestion + dilation. If we set $h$ to be the optimal value of congestion + dilation, or a constant-factor approximation of it, the produced routing is poly$(\log n)$-competitive in congestion + dilation, and thus also poly$(\log n)$-competitive in terms of the completion time, via random delays [16].

# 2 PRELIMINARIES

We first give some common notation we use throughout the paper.

*General:* We denote by $[k] = \{1, 2, \ldots, k\}$ for some non-negative integer $k$ and $A \sqcup B$ denotes the disjoint union of $A$ and $B$. We often use the Iverson bracket notation $\mathbb{I}[\text{condition}]$ which evaluates to 1 when the condition is true and 0 otherwise. We assume that all graphs are undirected and we typically assume the existence of an underlying graph $G = (V, E)$ with $n := |V|$.

*Weighted graphs:* A **weighted graph** $G = (V, E, w)$ is an undirected graph $(V, E)$ along with **weights** $w : E \to [1/L, L]$. Equivalently, a **capacitated graph** $G = (V, E, c)$ is an undirected graph along with **capacities** $c : E \to [1/L, L]$. Here, the value $L$ is the **aspect ratio** and throughout this paper we assume that $L = \text{poly}(n)$.

Many intermediate results only apply to complete graphs, i.e., where $E = \binom{V}{2}$, hence we will explicitly disambiguate between general and complete weighted graphs. In the context of this paper, assuming graphs are complete is mostly without loss of generality. In particular, one can often transform any non-complete weighted graph $G = (V, E, w)$ with aspect ratio $L$ into an equivalent complete graph $G'$ with aspect ratio $L' = n^2 \cdot L$ which gives a weight of $L'$ (or $1/L'$ in maximization problems) to any edge not in $E$ without affecting the results. We use both $w_e$ and $w(e)$ to refer to the weights.

*Distances and hop-constraints:* Let $p = (p_0, p_1, \ldots, p_\ell)$ be a path in $G = (V, E, w)$. We denote the number of **hops** in $p$ with $\text{hop}(p) := \ell$ and the sum of weights of the edges in the path with $w(p)$. Paths are assumed to be non-simple unless explicitly stated otherwise. We define the distance between $u, v \in V$ as $d_G(u, v) := \min\{w(p) \mid \text{path } p \text{ between } u, v\}$. Furthermore, we define the **hop-constrained distance** with $h > 0$ as $d_G^{(h)}(u, v) := \min\{w_G(p) \mid \text{path } p \text{ between } u, v \text{ with } \text{hop}(w) \leq k\}$.

*Trees:* Trees can be either unweighted $T = (V, E)$ or weighted $T = (V, E, w)$. Trees are often rooted, in the sense that there is a special node $r \in V$. With $T_{u,v}$ we denote the (unique) path on $T$ between $u$ and $v$. For example, in a weighted tree $T$ it holds that $d_T(u, v) = w_T(T_{u,v})$ for all $u, v \in V$.

## 2.1 Approximating Hop-Constrained Distances

In this section, we describe an important ingredient from prior work: partial tree embeddings, and how to use them to approximate hop-constrained distances. To give some context, the seminal results of [4, 5, 12] have shown that any metric space can be approximately embedded into a (distribution over) weighted trees. This has led to major advances in the field of approximation algorithms because many NP-hard problems on general graphs can be exactly solved on trees in polynomial time (see, for instance, the survey [13]).

However, such tree embedding results are not applicable to problems involving hop-constrained distances $d_G^{(h)}$ since they do not form a metric space. To address this issue, very recent work [14] proposed using *partial* tree embeddings, where only a fraction of nodes are embedded in any single tree. Their paper also shows that using standard (non-partial) trees in such a setting necessarily leads to unsatisfactory polynomial losses in approximation. We formalize the notion of embedding a graph into a tree.

*Definition 2.1.* A **partial tree embedding** $(T, T^G)$ on a graph $G = (V(G), E(G))$ consists of a rooted and tree $T = (V(T), E(T))$ with $V(T) \subseteq V(G)$, and a mapping $T^G$ which maps every tree edge $\{u, v\} = e \in E(T)$ to paths $T_e^G$ in $G$ that are between $u$ and $v$.

We extend the notation from Definition 2.1 to nodes in $T$ which are not adjacent: For any two vertices $u, v \in V(T)$, if $e_i$ is the $i^{th}$ edge in $T_{uv}$ (ordered from $u$ to $v$) then $T_{uv}^G := T_{e_1}^G \oplus T_{e_2}^G \oplus \ldots$ where $\oplus$ is concatenation. Finally, we extend the notion to weighted graphs $G$ where the distances in the embedding must "dominate" the distances in $G$.

*Definition 2.2.* A partial tree embedding $(T, T^G)$ is **dominating** if (i) it is defined over a *weighted* graph $G = (V(G), E(G), w_G)$, (ii) $T = (V(T), E(T), w_T)$ is a weighted tree, and (iii) $w_G(T_{u,v}^G) \leq d_T(u, v)$ for all $\{u, v\} \in E(G)$.

Note that if $(T, T^G)$ is a dominating partial tree embedding, we have that $w_G(T_{u,v}^G) \leq d_T(u, v)$ for all $u, v \in V(T)$.

Distributions over partial tree embedding are sufficiently expressive to approximate hop-constrained distances $d_G^{(h)}$ in any graph $G$. However, the approximation here is bicriteria: the distances are stretched by a factor $\alpha \geq 1$ and the hop lengths are stretched by a factor $\beta \geq 1$. Furthermore, while some nodes are missing from any particular tree, each node $v \in V(G)$ must be embedded in at least $1 - \varepsilon$ fraction of the trees.

*Definition 2.3 (h-Hop Partial Embedding Distribution).* An $h$-**hop partial embedding distribution** is a distribution over dominating partial tree embeddings $\mathcal{T}$ on a weighted graph $G$. We annotate $\mathcal{T}$ with the following properties:

- $\mathcal{T}$ has **hop stretch** $\beta \geq 1$ if each partial tree embedding $(T, T^G) \in \text{supp}(\mathcal{T})$ has $\text{hop}(T_{u,v}^G) \leq \beta \cdot h$ for all $u, v \in V(T)$.
- $\mathcal{T}$ has **exclusion probability** $\varepsilon > 0$ if each node $v \in V(G)$ we have $\Pr[v \in V(T)] \geq 1 - \varepsilon$.
- $\mathcal{T}$ has **expected distance stretch** $\alpha \geq 1$ if
$$\mathbb{E}_{(T,\cdot)\sim\mathcal{T}}\left[d_T(u, v) \cdot \mathbb{I}[u, v \in V(T)]\right] \leq \alpha \cdot d_G^{(h)}(u, v)$$
for all $u, v \in V(G)$.

Theorem 2.4 ([14]). *For every (complete) weighted graph $G$ with polynomially-bounded weights, every $h \geq 1$, and every $0 < \varepsilon < 1/3$ there exists an $h$-hop partial embedding distribution with hop stretch $O(\frac{\log^3 n}{\varepsilon})$, expected distance stretch $O(\log n \cdot \log \frac{\log n}{\varepsilon})$, and exclusion probability $\varepsilon$. Moreover, the distribution can be sampled in polynomial time.*

We elaborate on the notion in which $d^{(h)}$ is approximated by $h$-hop partial embedding distributions. An alternative way to view the expected distance stretch $\alpha$ is to interpret it as a variant of the expected stretch of the conditional distribution, i.e, for $\varepsilon < 1/3$ we have

$$\mathbb{E}_{(T,\cdot)\sim\mathcal{T}}\left[d_T(u, v) \cdot \mathbb{I}[u, v \in V(T)]\right] = \\ = \Theta\left(\mathbb{E}_{(T,\cdot)\sim\mathcal{T}}\left[d_T(u, v) \mid u, v \in V(T)\right]\right).$$

With this in mind and the fact that partial embedding distributions $\mathcal{T}$ are over dominating embeddings, we can rewrite the guarantee of Definition 2.3 in the following way for every $u, v \in V(G)$:

$$d_G^{(\beta \cdot h)}(u, v) \leq \mathbb{E}_{(T,\cdot)\sim\mathcal{T}}\left[d_T(u, v) \mid u, v \in V(T)\right] \leq O(\alpha) \cdot d_G^{(h)}(u, v).$$

## 2.2 Hop-Constrained Oblivious Routings

In this section we give a formal definition of hop-constrained oblivious routings.

**Fractional demands and routings.** The discussion in the introduction mostly focused on *integral* demands and routings—where the demand was a sequence of $(s_i, t_i)$ pairs and the routes were single paths between $s_i$ and $t_i$. However, we express our technical results in the more general—fractional—setting (see further discussion about this later in this section). To this end, a set of requests $\{(s_i, t_i)\}_i$ are abstracted and generalized via the (fractional) demand matrix $\mathcal{D} : V \times V \to \mathbb{R}_{\geq 0}$ where $\mathcal{D}_{s,t}$ intuitively represents the number of requests between $s$ and $t$. In other words, $\{(s_i, t_i)\}_i$ would correspond to the demand (matrix) $\mathcal{D}_{u,v} := |\{i : s_i = u, t_i = v\}|$.

Similarly, a path between $s$ and $t$ will typically be replaced by a distribution of paths between $s$ and $t$. This requires us to slightly revisit the notions of dilation and congestion, which were previously defined only in the integral case. The dilation of a distribution of paths is $h$ if the distribution is supported over paths of at most $h$ hops. Next, we formally introduce the notion of flows, capacitated graphs and congestion.

*Definition 2.5 (Flows).* A **flow** $f$ is a non-negative vector indexed over the edges of the underlying graph $G$, i.e., $f \in \mathbb{R}_{\geq 0}^{E(G)}$. Each (possibly non-simple) path $p$ has a naturally associated flow which we denote as $\text{flow}_G(p) = \text{flow}_G(p) = (\text{flow}_G(p,e))_{e \in E(G)} \in \mathbb{R}_{\geq 0}^{E(G)}$ where $\text{flow}_G(p,e)$ is defined as the number of times $p$ goes through $e$.

We express our results on capacitated graphs, which generalize multi-graphs, in the sense that a multi-graph with $k_e$ copies of an edge $e$ corresponds to a capacitated graph with an edge capacity $c_e := k_e$. This allows us to define the congestion of a flow with respect to capacities.

*Definition 2.6 (Capacitated graphs and congestion).* Capacitated graphs $G = (V, E, c)$ are undirected graphs with **capacities** $c : E \to [1/L, L]$, where $L = \text{poly}(n)$ is the aspect ratio. Given a flow $f \in \mathbb{R}_{\geq 0}^{E}$, we define the **congestion** $\text{cong}_G(f) = \max_{e \in E} f_e/c_e$ to be the maximum ratio of flow over capacity, across all edges.

**Optimally routing a demand via a (hop-constrained) routing scheme.** Given a particular demand $\mathcal{D}$, we are interested in the best hop-constrained (fractional) routing $R = \{R_{s,t}\}_{s,t \in V}$ of $\mathcal{D}$. This is formalized in the following definition.

*Definition 2.7 (Optimal hop-constrained routings).* A **demand** is a matrix $\mathcal{D} : V \times V \to \mathbb{R}_{\geq 0}$. Given a demand $\mathcal{D}$ and a routing scheme $\{R_{s,t}\}_{s,t \in V \times V}$ for a capacitated graph $G$, we define the **congestion of routing $\mathcal{D}$ using** $R$ as

$$\text{cong}_G(\mathcal{D}, R) = \text{cong}_G\left(\sum_{s,t} \mathbb{E}_{p \sim P_{s,t}}[\text{flow}(p) \cdot \mathcal{D}_{s,t}]\right).$$

Finally, given a demand $\mathcal{D}$ and a hop constraint $h \geq 1$, we define the **optimal $h$-hop routing of** $\mathcal{D}$, denoted by $\text{opt}^{(h)}(\mathcal{D})$, as the minimum congestion $\text{cong}_G(\mathcal{D}, R)$ over all possible routing schemes $R = \{R_{s,t}\}_{s,t \in V}$ supported over paths of at most $h$ hops.

We reiterate that all routings and distributions in our paper are fractional, in the sense that a unit demand from $s$ to $t$ is carried over a distribution of paths, incurring fractional congestion on each one of these paths. Another important point of emphasis in the definition of $\text{opt}^{(h)}(\mathcal{D})$ is that the routing scheme $P$ can depend on the demand $\mathcal{D}$, i.e., is *adaptive* to the demand. In contrast, a good oblivious routing is a single routing scheme that is *obliviously competitive* with respect to *all demands* $\mathcal{D}$. We now define the principal concept in our paper.

*Definition 2.8.* An **$h$-hop oblivious routing** for a graph $G = (V, E)$ is a routing scheme $R$ that is additionally annotated in the following way:

(1) $R$ has **hop stretch** $\beta \geq 1$ if for all $s, t \in V$ all paths $p \in \text{supp}(R_{s,t})$ have $\text{hop}(p) \leq \beta \cdot h$.

(2) $R$ has **congestion approximation** $\alpha \geq 1$ if for all demands $\mathcal{D} : V \times V \to \mathbb{R}_{\geq 0}$ we have that $\text{cong}_G(\mathcal{D}, R) \leq \alpha \cdot \text{opt}^{(h)}(\mathcal{D})$.

**Integral vs. fractional routings.** Our choice to express our results in the fractional setting has multiple benefits. For one, the demand in our (fractional) setting is scale-invariant, in that a routing scheme that is competitive with respect to $\mathcal{D}$ will be competitive with respect to $\gamma \cdot \mathcal{D}$, for any $\gamma > 0$. Furthermore, one can easily recover the integral setting from the fractional one, making our choice more general. We elaborate on this. Suppose we are given an $h$-hop oblivious routing $R$ with hop stretch $\beta$ and congestion approximation $\alpha \gg \log n$. Given a set of requests $\{(s_i, t_i)\}_i$, suppose that some set of paths $\{p_i^*\}_i$ with at most $h$ hops connecting the source-sink pairs has optimal congestion $C^*$. For each request $i$ we independently randomly sample a path $p_i' \sim R_{s_i,t_i}$.

We now argue that $\{p_i'\}$ has dilation $\beta \cdot h$ and congestion at most $O(\alpha) \cdot C^*$ with high probability. The dilation bound follows from definition. We now argue that for each edge $e$ the expected number of times $\{p_i'\}$ crosses $e$ is $\alpha \cdot C^* \cdot c_e$. First, for all $u, v$ we set $\mathcal{D}_{u,v} := |\{i : s_i = u, t_i = v\}|$ and then let $P_{u,v}$ be a uniform distribution over the $\mathcal{D}_{u,v}$ paths with endpoints $u, v$. We note that $\text{opt}^{(h)}(\mathcal{D}) \leq \text{cong}_G(\mathcal{D}, P) \leq C^*$, where the first inequality is by definition and second is due to $\text{cong}_G(\mathcal{D}, P)$ being exactly equal to the congestion of a set of paths $\{p_i^*\}_i$. Therefore, the expected number of drawn paths $\{p_i'\}$ crossing an edge $e$ is at most $\alpha \cdot c_e \cdot \text{opt}^{(h)}(\mathcal{D}) \leq \alpha \cdot c_e \cdot C^*$, as required.

We now argue that the congestion of $\{p_i'\}_i$ is at most $\alpha \cdot C^*$ with high probability. Since each path was drawn independently at random and the paths can always assumed to be simple (simplying a path does not increase the congestion or the dilation), the number of paths $\{p_i'\}$ crossing an edge $e$ can be seen as a sum of independent $\{0, 1\}$-variables. We can apply a standard Chernoff bound and conclude that this number is at most $O(c_e \cdot \alpha \cdot C^* + \log n)$ with high probability. Since $C^* \cdot c_e \geq 1$ and $\alpha \gg \log n$, we have that $O(c_e \cdot \alpha \cdot C^* + \log n) = O(c_e \cdot \alpha \cdot C^*)$ with high probability. Union bounding over all edges, we conclude that the congestion of $\{p_i'\}_i$ is at most $O(\alpha) \cdot C^*$ with high probability, as required.

## 3 HOP-CONSTRAINED OBLIVIOUS ROUTING: A TECHNICAL OVERVIEW

We now formally state our main result.

Theorem 3.1. *For every (general) capacitated graph $G = (V, E, c)$ and every $h \geq 1$, there exists an $h$-hop oblivious routing with hop stretch $O(\log^7 n)$ and congestion approximation $O(\log^2 n \cdot \log^2(h \log n))$.*

**Remark.** Notice that there is a small discrepancy between the definition of congestion in Theorem 3.1 and its informal counterpart Theorem 1.3. The former talks about fractional routings (e.g., maximum expected congestion), while the later is about integral routings (e.g., expected maximum congestion). However, our formal statement (Theorem 3.1) implies the informal one, as argued in Section 2.2.

The rest of the paper is structures as follows. In Section 3.1 we explain why previous (tree-based) approaches fail to attain oblivious routings with hop constraints. In Section 3.2 we give an overview

of the routing scheme that achieves the guarantees of the hop-constrained oblivious routing. Finally, in Section 4 we prove the guarantees of our routing scheme.

## 3.1 Tree-Based Hop-Constrained Oblivious Routings Cannot Have Good Guarantees

In this section, we showcase a barrier that prevented prior approaches from achieving poly$(\log n)$-competitive hop-constrained oblivious routing. State-of-the-art oblivious routings are generally *tree-based routings*[2] [6, 11, 19, 20, 25], i.e., where a demand from $s$ to $t$ is routed by randomly sampling a tree $T$ from a fixed distribution $\mathcal{T}$ and then picking the tree-defined path $T^G_{u,v}$. We show that tree-based hop-constrained oblivious routings cannot have good guarantees.

*Definition 3.2.* A **complete tree embedding** $(T, T^G)$ of a graph $G = (V(G), E(G))$ consists of a tree $T = (V(T), E(T))$ where $V(G) = V(T)$, and a mapping $T^G$ which maps every edge $e \in E(G)$ to a path in $G$ between $e$'s endpoints.

We extend the definition of $T^G_{u,v}$ when $u, v$ are not adjacent in the natural way as in Definition 2.1. Note that the tree from the complete tree embedding is not necessarily a subtree of $G$, but it does contain *all* nodes of $G$.

*Definition 3.3.* A **tree-based routing scheme** $R^{\mathcal{T}}$ is a routing scheme that is induced by a distribution over complete tree embeddings $\mathcal{T}$ in the following way: we sample from $R^{\mathcal{T}}_{s,t}$ by sampling an embedding $(T, T^G) \sim \mathcal{T}$ and returning $T^G_{s,t}$.

*The $\mathcal{D}^{(1)}$ demand.* For a capacitated graph $G = (V, E, c)$ we define a special demand $\mathcal{D}^{(1)}$ which has a unit demand across every edge $e$ (for each unit of capacity), i.e., $\mathcal{D}^{(1)}_{s,t} := \mathbb{I}[\{s,t\} \in E] \cdot c_{\{s,t\}}$. This demand is particularly important for both the congestion-only and hop-constrained oblivious routings, as we shortly explain. Tree-based routings are especially suitable for constructing oblivious routings due to the following well-known fact. If a tree-based oblivious routing incurs congestion approximation $\alpha$ on the single demand $\mathcal{D}^{(1)}$, then it incurs congestion approximation at most $\alpha$ on all demands. This greatly simplifies the design of good tree-based routing schemes: one only needs to ensure the single $\mathcal{D}^{(1)}$ is routed in a good manner. On the other hand, one of the many aspects in which hop-constrained oblivious routing schemes are significantly more difficult than congestion-only routings is that one needs to explicitly ensure that other (non-$\mathcal{D}^{(1)}$) demands have good approximation guarantees when routed. The following statement formalizes the claim in the tree-based routing case (note that the discussion up to this point is for the unconstrained-hop setting, hence $h = \infty$).

Lemma 3.4. *Let $G = (V, E, c)$ be a capacitated graph and suppose that a tree-based routing scheme $R^{\mathcal{T}} = \{R^{\mathcal{T}}_{s,t}\}_{s,t \in V}$ achieves $\text{cong}_G(\mathcal{D}^{(1)}, R^{\mathcal{T}}) \le \alpha$, where $\mathcal{D}^{(1)} : V \times V \to \mathbb{R}_{\ge 0}$ with $\mathcal{D}^{(1)}_{s,t} := \mathbb{I}[\{s,t\} \in E] \cdot c_{\{s,t\}}$. Then for every demand $\mathcal{D} : V \times V \to \mathbb{R}_{\ge 0}$ we have $\text{cong}_G(\mathcal{D}, R^{\mathcal{T}}) \le \alpha \cdot \text{opt}^{(\infty)}(\mathcal{D})$.*

[2]We also note that routings supported on hierarchically separated trees or HSTs can be converted into tree-based routings with at most a constant loss in their guarantees.

Proof. The claim is implicit in, e.g., [19]. Claim 3 in Section 2 of [19] gives a definition of expected relative load $\alpha$ of a distribution over complete tree embeddings. The definition is equivalent to saying that for each edge $e$ the expected amount of flow routed over $e$ when routing the $\mathcal{D}^{(1)}$ demand is at most $\alpha \cdot c_e$. The Subsection titled "Oblivious Routing" of Section 3 that routing any set of demands $\mathcal{D}$ over a distribution with expected relative load $\alpha$ implies that the achieved routing has congestion approximation at most $\alpha$, as required. □

Unfortunately, hop-constrained oblivious routings with polylogarithmic hop stretch and congestion approximation cannot come from tree-based routings. This observation prevents all prior work for general graphs the authors are aware of from achieving good-quality routings that control both the congestion and dilation.

Lemma 3.5. *There exists an infinite family of graphs with unit capacities and diameter 4 such that for every graph $G$ in the family the following holds. For every $h \ge 1$, any tree-based $h$-hop oblivious routing $R^{\mathcal{T}}$ for $G$ with hop stretch $\beta$ and congestion approximation $\alpha$ has $\alpha \cdot \beta \cdot h \ge \Omega(\sqrt{n})$.*

Proof. By definition of hop stretch, we have that $R^{\mathcal{T}}$ is supported over paths of length at most $\beta h$. Furthermore, due to the congestion approximation being at most $\alpha$, the congestion of $R^{\mathcal{T}}$ on the demand $\mathcal{D}^{(1)}$ is $\alpha \cdot \text{opt}^{(h)}(\mathcal{D}) = \alpha$. Since $R^{\mathcal{T}}$ is a tree-based routing, we conclude via Lemma 3.4 that for all demands $\mathcal{D}$ the congestion of $R^{\mathcal{T}}$ on $\mathcal{D}$ is $\alpha$-competitive with $\text{opt}^{(\infty)}(\mathcal{D})$: $\text{cong}_G(\mathcal{D}, R^{\mathcal{T}}) \le \alpha \cdot \text{opt}^{(\infty)}(\mathcal{D})$. In other words, $R^{\mathcal{T}}$ is also a $\infty$-hop oblivious routing with congestion approximation $\alpha$ (in spite of being supported only on paths of length $\beta h$).

We now construct a graph $G$ with $n+1$ vertices which exhibits our bound. We take $\sqrt{n}$ paths $p_1, p_2, \ldots, p_{\sqrt{n}}$ of length $\text{hop}(p_i) = \sqrt{n} - 1$. Label the first and last node of $p_i$ with $s_i$ and $t_i$. For each path $p_i$ we connect each $j^{th}$ node to the $j^{th}$ node of $p_1$. Finally, we create a new node $r$ and connect it to all nodes on $p_1$ with edges we call "uplinks". The diameter of $G$ is clearly 4 (e.g., the hop distance between any node and $r$ is 2). We consider the demand $\mathcal{D}_{s,t} = \mathbb{I}[\exists i \text{ such that } (s,t) = (s_i, t_i)]$. Clearly, $\text{opt}^{(\infty)}(\mathcal{D}) \le 1$ since the demand between $s_i$ and $t_i$ can be sent across the $(\sqrt{n} - 1)$-hop path $p_i$, resulting in congestion 1. Furthermore, any path between the start and end of some $p_i$ of hop length at most $\beta h$ must cross one of the first $\beta h$ uplinks. Since there are $\sqrt{n}$ such demands, we conclude that the congestion of at least one of the uplinks is $\frac{\sqrt{n}}{\beta \cdot h} \le \alpha \cdot \text{opt}^{(h)}(\mathcal{D}) = \alpha$. □

**Remark.** The $\sqrt{n}$ bound can be improved to $\tilde{\Omega}(n)$ for unit-capacity graphs of diameter $O(\log n)$ using the well-known worst-case network family from [10].

## 3.2 An Overview of the Hop-Constrained Oblivious Routing

In this section, we give an overview of our hop-constrained oblivious routing.

While one cannot obtain hop-constrained oblivious routings via tree-based routings (i.e., complete tree embeddings, as argued in Section 3.1), we show that distributions over *partial tree embeddings*

yield useful results. We first introduce the notion of $\mathcal{D}^{(1)}$**-routers**, which are distributions over partial tree embeddings $\mathcal{T}$ and are a crucial building block of hop-constrained oblivious routings. We describe several important aspects of $\mathcal{D}^{(1)}$-routers before we formally define them in Definition 3.6.

(1) First, we explain how $\mathcal{T}$ induces an "routing scheme" $R^{\mathcal{T}} = \{R^{\mathcal{T}}_{s,t}\}_{s,t}$. For some $s, t \in V(G)$ we sample a partial tree embedding $(T, T^G) \in \mathcal{T}$; if $s, t \in V(T)$ we return $T^G_{s,t}$; otherwise, we simply return a special symbol $\perp$ which represents "failure". Clearly, $R^{\mathcal{T}}$ is not a valid routing scheme in the sense of Definition 1.1 since $\perp \in \mathrm{supp}(R_{s,t})$, but this is somewhat unavoidable when dealing with partial tree embeddings.

(2) $\mathcal{D}^{(1)}$-routers for a graph $G$ get their name from being able to route the $\mathcal{D}^{(1)}$ demand, defined as $\mathcal{D}^{(1)}_{s,t} := \mathbb{I}[\{s,t\} \in E(G)] \cdot c_{\{s,t\}}$. While this does not directly guarantee good congestion approximation on all demands (unlike tree-based routings, c.f. Lemma 3.4), it does lead to certain useful properties. First, we formalize what we mean by routing the $\mathcal{D}^{(1)}$ demand over routing scheme $R^{\mathcal{T}}$ which is induced by a distribution over partial tree embeddings $\mathcal{T}$. For a demand $\mathcal{D}$ we define $\mathrm{cong}_G(\mathcal{D}, R^{\mathcal{T}})$ as the maximum expected congestion while only routing non-failures:

$$\begin{aligned}
&\mathrm{cong}_G(\mathcal{D}, R^{\mathcal{T}}) \\
&= \mathrm{cong}_G\left(\sum_{s,t} \mathbb{E}_{p \sim R^{\mathcal{T}}_{s,t}}\left[\ \mathbb{I}[p \neq \perp] \cdot \mathrm{flow}(p) \cdot \mathcal{D}_{s,t}\ \right]\right) \\
&= \mathrm{cong}_G\left(\sum_{s,t} \mathbb{E}_{(T,T^G)\sim\mathcal{T}}\left[\ \mathbb{I}[s,t \in V(T)] \cdot \mathrm{flow}(T^G_{s,t}) \cdot \mathcal{D}_{s,t}\ \right]\right).
\end{aligned}$$

Finally, we say that a $\mathcal{D}^{(1)}$-router has **congestion approximation** $\alpha$ if $\mathrm{cong}_G(\mathcal{D}^{(1)}, R^{\mathcal{T}}) \leq \alpha$.

(3) In our hop-constrained setting, we need to control the hop length of the paths over which $\mathcal{T}$ routes. For this reason, we define the **dilation** of $\mathcal{T}$ to be $\beta$ if $\mathcal{T}$ is supported over partial tree embeddings $(T, T^G)$ where $\mathrm{hop}(T^G_{s,t}) \leq \beta$ for all $s, t \in V(T)$.

(4) When talking about distributions over partial tree embeddings, a new parameter called **exclusion probability** becomes important. The exclusion probability is $\varepsilon$ if for each node $v \in V(G)$ the probability that $v$ is excluded from the tree $(T, \cdot) \sim \mathcal{T}$ is at most $\varepsilon$. Note that this parameter also appears when talking about partial tree embeddings that approximate hop-constrained distances (Section 2.1).

We now give a formal definition equivalent to the above description and state its existence lemma.

*Definition 3.6.* A $\mathcal{D}^{(1)}$**-router** is a distribution over partial tree embeddings $\mathcal{T}$ on a capacitated graph $G$ that is additionally annotated in the following way:

(1) $\mathcal{T}$ has **dilation** $\beta \geq 1$ if each partial tree embedding $(T, T^G) \in \mathrm{supp}(\mathcal{T})$ has $\mathrm{hop}(T^G_{u,v}) \leq \beta$ for all $u, v \in V(T)$.

(2) $\mathcal{T}$ has **exclusion probability** $\varepsilon > 0$ if each node $v \in V(G)$ we have $\Pr[v \in V(T)] \geq 1 - \varepsilon$.

(3) $\mathcal{T}$ has **congestion** $\alpha \geq 1$ if

$$\mathrm{cong}_G\Big(\sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot \mathbb{E}_{(T,T^G)\sim\mathcal{T}}\left[\ \mathbb{I}[u,v \in V(T)] \cdot \mathrm{flow}(T^G_{u,v})\ \right]\Big) \leq \alpha.$$

Lemma 3.7. *For every (complete) capacitated graph $G$ and $0 < \varepsilon < 1/3$ there exists a $\mathcal{D}^{(1)}$-router with dilation $O(\frac{\log^3 n}{\varepsilon})$, exclusion probability $\varepsilon$, and congestion $O(\log n \cdot \log \frac{\log n}{\varepsilon})$.*

Proof deferred to Appendix A.

Having constructed $\mathcal{D}^{(1)}$-routers, the next and final step is to "lift" them into a proper hop-constrained oblivious routing. In this section, we aim only to give an overview, hence we will only show the sampling algorithm and defer arguing about its correctness to Section 4. Algorithm 1 shows how to sample the $h$-hop oblivious routing $R = \{R_{s,t}\}_{s,t\in V}$ that satisfies the constraints of our main result, Theorem 3.1.

**Algorithm 1** Sample a path $p \sim R_{s,t}$, given a graph $G$, $s, t \in V(G)$, and a hop constraint $h \geq 1$.

```
Create a "completion" H = (V, \binom{V}{2}, c_H) of G = (V, E, c_G).
    c_H(e) := c_G(e) if e ∈ E,
    c_H(e) := n^{-O(1)} · min_{e∈e} c_G(e) otherwise.
Let T_1 be a D^(1)-router on H with exclusion probability ε_1 = 1/(4h)
Sample r := O(log n) trees T_1, T_2, ..., T_r ~ T_1 conditioned on s, t ∈ V(T_i).
Assign q_1 := (T_1)^G_{s,t}, q_2 := (T_2)^G_{s,t}, ..., q_r := (T_r)^G_{s,t}.
Let T_2 be a D^(1)-router on H with exclusion probability ε_2 = 1/O(h log^4 n).
Sample a tree (F, F^G) ~ T_2 and let p := F^G_{s,t}.
    Simplify p by eliminating all cycles.
    Repeat the sampling of F and p if ⋃_{i=1}^r V(q_i) ⊈ V(T).
    Repeat the sampling of F and p if E(p) ⊈ E(G).
Return p.
```

We note that the final routing scheme $R$ is a conditional distribution induced by partial tree embeddings $\mathcal{T}_2$, conditioned on (1) the sampled tree $F$ containing the nodes of $O(\log n)$ sampled paths from $\mathcal{T}_1$, and (2) the sampled path using only edges in $G$ (i.e., not using virtual edges constructed during the completion of $G$).

## 4 LIFTING THE $\mathcal{D}^{(1)}$-ROUTER TO A HOP-CONSTRAINED OBLIVIOUS ROUTING

In this section, we describe and prove how to construct hop-constrained oblivious routings satisfying Theorem 3.1 from $\mathcal{D}^{(1)}$-routers. From a high-level, we first show that $\mathcal{D}^{(1)}$-routers can be used to route other demands $\mathcal{D} \neq \mathcal{D}^{(1)}$ if one allows for a constant fraction of failures (Section 4.1). Next, we show a method of "correcting" the number of failures down to $n^{-O(1)}$ with a polylogarithmic increase in congestion approximation and hop stretch guarantees (Section 4.2). Finally, we show how to eliminate failures entirely and extend our results to non-complete graphs (Section 4.3).

## 4.1 Hop-Constrained Subflow Routing

In this section, we argue that $\mathcal{D}^{(1)}$-routers are indeed useful for demands $\mathcal{D} \neq \mathcal{D}^{(1)}$. This is not unexpected: for tree-based routings in the congestion-only setting, a good $\mathcal{D}^{(1)}$-router immediately gives a good oblivious routing (Lemma 3.4). However, when dealing with distributions over partial tree embedding, one needs to take special care of failures $\perp$ that can arise when nodes are missing from the partial tree embeddings.

We introduce the concept of *subdistributions*. Suppose that we have a random variable $x$ which, sometimes, produces an unusable result. To model this, we introduce another random variable $y$, which can either be equal to $x$ (in case of success) or be $\perp$ in case of failures. Moreover, the probability of failure is controlled. The distributions of such variables $x$ and $y$ satisfy the following relation.

*Definition 4.1.* Let $X$ be a distribution over a set $U$. A distribution $Y$ over $U \cup \{\perp\}$ is an $\gamma$-**subdistribution** of $X$ if for all $u \in U$ it holds that $\Pr[Y = u] \leq \Pr[X = u]$ and $\Pr[Y = \perp] \leq \gamma$.

An equivalent definition of $Y$ being a subdistribution of $X$ is to say that we can construct a probability space with random variables $x \sim X$ and $y \sim Y$ such that $y \in \{x, \perp\}$.

To simplify notation, in this section we will often conflate a distribution $X$ and a random variable $x \sim X$. Naturally, one has to be careful in doing so since defining random variables requires defining a probability space. Sometimes there is no ambiguity about how to properly formalize the space (e.g., Definition 4.2, where linearity of expectation makes differences immaterial). However, in places where it matters, we will be careful to make the space clear from the context.

We now define the main concept of this section: subflow routing. Intuitively, a subflow routing is a routing scheme $R = \{R_{s,t}\}_{s,t \in V(G)}$ where we allow some paths to "fail". Similar to $\mathcal{D}^{(1)}$-routers, the paths that fail do not contribute to the congestion. Moreover, these failures can be adaptive to the demand, i.e., they are demand dependent, but the fraction of failures must be tightly controlled by a new parameter $0 < \gamma < 1$.

*Definition 4.2.* An $h$-**hop** $\gamma$-**subflow routing** with **congestion approximation** $\alpha \geq 1$ for a graph $G = (V, E)$ is a routing scheme $R = \{R_{u,v}\}_{u,v \in V}$ with the following property. For every demand $\mathcal{D} : V \times V \to \mathbb{R}_{\geq 0}$ there exists a "routing scheme with failures" $R' = R'(\mathcal{D}) = \{R'_{s,t}\}_{s,t \in V}$ where $R'_{s,t}$ is an $\gamma$-subdistribution of $R_{s,t}$ and

$$\text{cong}_G\Big(\sum_{s,t \in V} \mathbb{E}\left[\, \mathbb{I}[R'_{s,t} \neq \perp] \cdot \text{flow}(R'_{s,t}) \cdot \mathcal{D}_{s,t} \,\right]\Big) \leq \alpha \cdot \text{opt}^{(h)}(\mathcal{D}).$$

Additionally, we say that $R$ has **hop stretch** $\beta$ if for all $s, t \in V$ all paths $p \in \text{supp}(R_{s,t})$ have $\text{hop}(p) \leq \beta h$.

While in this paper we do not focus much on the computational aspects, we will note that the user of an ($h$-hop) $\gamma$-subflow routing cannot differentiate between failures and non-failures. They will be obliviously corrected down to $n^{-O(1)}$-fractions in Section 4.2 and then entirely eliminated in an oblivious manner in Section 4.3. We emphasize the oblivious part since the final hop-constrained oblivious routing makes no mention of failures (e.g., see Algorithm 1 or Theorem 3.1). We now show that $\mathcal{D}^{(1)}$-routers are indeed 1/2-subflow routings.

Lemma 4.3 ($\mathcal{D}^{(1)}$-routers are 1/2-subflow routings). *Let $\mathcal{T}$ be a $\mathcal{D}^{(1)}$-router on $G$ with dilation $\beta h$, exclusion probability at most $1/(4h)$, and congestion $\alpha \geq 1$. Let $R^{\mathcal{T}}$ be the routing scheme induced by $\mathcal{T}$ where the distribution $R^{\mathcal{T}}_{s,t}$ corresponds to sampling $(T, T^G) \sim \mathcal{T}$ and returning $T^G_{s,t}$ if $s, t \in V(T)$ and an arbitrary path otherwise. Then $R^{\mathcal{T}}$ is an $h$-hop $\frac{1}{2}$-subflow routing with congestion approximation $\alpha$ and hop stretch $\beta$.*

We remark that our definition of $R^{\mathcal{T}}_{s,t}$ sometimes samples a tree $T$ such that $s \notin V(T)$ or $t \notin V(T)$, in which case we return an arbitrary path. This is sensible because our analysis will draw a $\perp$ in such cases.

Proof. For simplicity of notation, we extend the definition of flow (which maps paths to their flows in $\mathbb{R}^{E(G)}_{\geq 0}$) to $\text{flow}(\perp) = \vec{0}$. Due to this, we can replace $\mathbb{I}[p \neq \perp] \cdot \text{flow}(p) = \text{flow}(p)$.

Clearly, the dilation property of $\mathcal{D}^{(1)}$-routers implies that each path $p \in \text{supp}(T^G_{s,t})$ has $\text{hop}(p) \leq \beta h$, hence the hop stretch property of $R^{\mathcal{T}}$ is immediate.

**Construction of subdistributions.** Fix a demand $\mathcal{D} : V(G) \times V(G) \to \mathbb{R}_{\geq 0}$. By definition, there exists a "witness" routing scheme $P^* = \{P^*_{s,t}\}_{s,t \in V(G)}$ that certifies the optimal $h$-hop routing solution. In other words, the support of $P^*_{s,t}$ is over $h$-hop paths connecting $s$ and $t$, and $\text{cong}_G(\sum_{s,t \in V(G)} \mathbb{E}[\mathcal{D}_{s,t} \cdot \text{flow}(P^*_{s,t})]) = \text{opt}^{(h)}(\mathcal{D})$. Equivalently, expanding the definition of $\text{cong}_G$, for every $\{u, v\} \in E(G)$:

$$\sum_{s,t \in V(G)} \mathbb{E}[\mathcal{D}_{s,t} \cdot \text{flow}(P^*_{s,t}, \{u, v\})] \leq \text{opt}^{(h)}(\mathcal{D}) \cdot c_{\{u,v\}}. \quad (1)$$

We now construct the collection of subdistributions $\{R'_{s,t}\}_{s,t}$. First, fix $s, t \in V(G)$. Then independently (of $P^*$) sample $(T, T^G) \sim \mathcal{T}$. Consider the event $V(P^*_{s,t}) \subseteq V(T)$, namely, that all nodes of a path $p = P^*_{s,t}$ are in $V(T)$, i.e., $V(p) \subseteq V(T)$. If $V(P^*_{s,t}) \subseteq V(T)$ then we assign $R'_{s,t} = T^G_{s,t}$ and otherwise $R'_{s,t} = \perp$. It is clear that $\Pr[R'_{s,t} = p] \leq \Pr[R_{s,t} = p]$ for every $p$ and every $s, t$. Furthermore, using the exclusion probability and a union bound, $\Pr[R'_{s,t} = \perp] = \Pr[V(P^*_{s,t}) \subseteq V(T)] \leq \frac{\text{hop}(P^*_{s,t})+1}{4h} \leq \frac{h+1}{4h} \leq \frac{1}{2}$. Therefore, $R'_{s,t}$ is a $\frac{1}{2}$-subdistribution of $R^{\mathcal{T}}_{s,t}$.

**Subdistribution congestion analysis.** For the sake of the analysis, for each $s, t \in V(G)$ we also introduce a random flow variable $C_{s,t} \in \mathbb{R}^{E(G)}_{\geq 0}$ (in the same probability space as above) as follows. In the event $V(P^*_{s,t}) \subseteq V(T)$ we consider the nodes on the path $P^*_{s,t} = (s = v_0, v_1, \ldots, v_{\ell-1}, v_\ell = t)$ and assign

$$\begin{aligned} C_{s,t} &:= \mathbb{I}[V(P^*_{s,t}) \subseteq V(T)] \cdot \sum_{i=0}^{\ell-1} \text{flow}(T^G_{v_i,v_{i+1}}) \\ &\geq \mathbb{I}[V(P^*_{s,t}) \subseteq V(T)] \cdot \text{flow}(T^G_{s,t}). \end{aligned}$$

Note that the right-hand side inequality holds because in any partial tree embedding $\text{flow}(T^G_{a,c}) \leq \text{flow}(T^G_{a,b}) + \text{flow}(T^G_{b,c})$ for all $a, b, c \in$

$V(T)$. Therefore,

$$\sum_{s,t} \mathbb{E}[\text{flow}(R'_{s,t})\mathcal{D}_{s,t}] = \sum_{s,t} \mathbb{E}\left[\mathbb{I}[V(P^*_{s,t}) \subseteq V(T)]\,\text{flow}(T^G_{s,t})\mathcal{D}_{s,t}\right] \le \sum_{s,t} \mathbb{E}[C_{s,t}] \cdot \mathcal{D}_{s,t}. \quad (2)$$

On the other hand, we now give an upper bound for $\sum_{s,t} \mathbb{E}[C_{s,t} \cdot \mathcal{D}_{s,t}]$.

$$\begin{aligned}
&\sum_{s,t} \mathbb{E}[C_{s,t}\mathcal{D}_{s,t}] \\
&= \sum_{s,t} \mathcal{D}_{s,t} \cdot \mathbb{E}\left[\mathbb{I}[V(P^*_{s,t}) \subseteq V(T)] \cdot \sum_{i=0}^{\ell-1} \text{flow}(T^G_{v_i,v_{i+1}})\right] \\
&= \sum_{s,t} \mathcal{D}_{s,t} \cdot \mathbb{E}\Big[\mathbb{I}[V(P^*_{s,t}) \subseteq V(T)] \cdot \\
&\qquad \sum_{\{u,v\}\in E(G)} \text{flow}(P^*_{s,t}, \{u,v\}) \cdot \text{flow}(T^G_{u,v})\Big] \\
&= \sum_{\{u,v\}\in E(G)} \mathbb{E}\Big[\text{flow}(T^G_{u,v}) \cdot \\
&\qquad \sum_{s,t} \mathcal{D}_{s,t} \cdot \mathbb{I}[V(P^*_{s,t}) \subseteq V(T)] \cdot \text{flow}(P^*_{s,t}, \{u,v\})\Big] \\
&\le \sum_{\{u,v\}\in E(G)} \mathbb{E}[\mathbb{I}[u,v \in V(T)] \cdot \text{flow}(T^G_{u,v})] \cdot \\
&\qquad \sum_{s,t} \mathcal{D}_{s,t} \cdot \mathbb{E}\left[\text{flow}(P^*_{s,t}, \{u,v\})\right] \\
&\le \sum_{\{u,v\}\in E(G)} \mathbb{E}[\mathbb{I}[u,v \in V(T)] \cdot \text{flow}(T^G_{u,v})] \cdot \\
&\qquad \text{opt}^{(h)}(\mathcal{D}) \cdot c_{\{u,v\}} \qquad \text{(Equation (1))} \\
&\le \text{opt}^{(h)}(\mathcal{D}) \cdot \alpha \cdot \vec{c_G} \qquad (\mathcal{D}^{(1)}\text{-router properties})
\end{aligned}$$

Combining the above with Equation (2) we get that

$$\text{cong}_G\Big(\sum_{s,t} \mathbb{E}[\text{flow}(R'_{s,t}) \cdot \mathcal{D}_{s,t}]\Big) \le \alpha \cdot \text{opt}^{(h)}(\mathcal{D}). \qquad \square$$

## 4.2 Correcting Subflow Failures

In this section, we show how to drive down the failure bound $\gamma$ from $1/2$ to $n^{-O(1)}$. We construct a $n^{-O(1)}$-subflow routing $F$ in the following way. Suppose that $R$ is an $1/2$-subflow routing and fix $s,t \in V(G)$. We sample $r := O(\log n)$ paths $\{q_i\}_{i=1}^r$ from $R_{s,t}$ and let $\mathcal{T}$ be a $\mathcal{D}_1$-router with exclusion probability at most $(2\sum_{i=1}^r \text{hop}(q_i))^{-1}$. We sample a single tree embedding $(T, T^G) \sim \mathcal{T}$ conditioned on $V(T)$ containing all nodes of $\{q_i\}_{i=1}^r$, i.e., $\bigcup_{i=1}^r V(q_i) \subseteq V(T)$. The sampled path in our new routing scheme is then $F_{s,t} := T^G_{s,t}$: we claim $F$ is an $n^{-O(1)}$-subflow routing.

Lemma 4.4 (Reducing $\gamma$). *Given an $h$-hop $\gamma$-subflow routing $R$ with hop stretch $\beta$ and congestion approximation $\alpha$ for a (complete) capacitated graph $G$, there exists an $h$-hop $(\gamma^r)$-subflow routing $F$ with hop stretch $O(r\beta\log^3 n)$ and congestion approximation $O(\frac{\alpha}{1-\gamma} \cdot \log n \cdot \log(rh\beta\log n))$, for every integer $r \ge 2$.*

Proof. For simplicity of notation, we extend the definition of flow (which maps paths to their flows in $\mathbb{R}^{E(G)}_{\ge 0}$) to $\text{flow}(\perp) = \vec{0}$. Due to this, we can replace $\mathbb{I}[p \ne \perp] \cdot \text{flow}(p) = \text{flow}(p)$.

Let $\mathcal{T}$ be a $\mathcal{D}^{(1)}$-router with exclusion probability $1/(2rh\beta)$, dilation $O(rh\beta\log^3 n)$, and congestion $O(\log n \cdot \log(rh\beta\log n))$ (via Lemma 3.7).

**Construction of the routing scheme** $F = \{F_{u,v}\}_{u,v\in V(G)}$. Fix $u,v \in V(G)$. We construct $F_{u,v}$ as follows. Independently sample $r$ paths $q_{u,v,1}, q_{u,v,2}, \ldots, q_{u,v,r}$ from $R_{u,v}$. Let $S_{u,v} = \bigcup_{i=1}^r V(q_{u,v,i})$ be the set of nodes on the union of the $r$ sampled paths. Note that We have $|S_{u,v}| \le |\{u,v\}| + \sum_{i=1}^r \big(\text{hop}(q_{u,v,i}) - 1\big) \le 2 + r(h\beta - 1) \le rh\beta$. Now, consider the partial tree distribution $(T, T^G)$ from $\mathcal{T}$ conditioned on $S_{u,v} \subseteq V(T)$. We denote this conditional distribution as "$\mathcal{T} \mid S_{u,v} \subseteq V(T)$". With this notation in place, we set $F_{u,v} = T^G_{u,v}$ where $(T, T^G)$ is (independently) sampled from $\mathcal{T} \mid S_{u,v} \subseteq V(T)$.

**Hop stretch.** Since $\mathcal{T}$ has dilation $h \cdot O(r\beta\log^3 n)$ we conclude that the hop stretch of $F$ is $O(r\beta\log^3 n)$.

**Construction of subdistributions.** Fix a demand $\mathcal{D} : V(G) \times V(G) \to \mathbb{R}_{\ge 0}$. We now construct the collection of subdistributions $F' = \{F'_{u,v}\}_{u,v}$. Fix $u,v \in V(G)$. We can reinterpret the construction of $F_{u,v}$ in the following way. The original process independently samples $r$ paths $q_{u,v,1}, \ldots, q_{u,v,r} \sim R_{u,v}$. We reinterpret this as sampling $r$ paths $q'_{u,v,1}, \ldots, q'_{u,v,r}$ independently from $R'_{u,v}$, which depends on the demand $\mathcal{D}$ (unlike $R_{u,v}$ that does not depend on $\mathcal{D}$). This effectively stores $q'_{u,v,j} \leftarrow q_{u,v,j}$, except that we replace $q'_{u,v,j} \leftarrow \perp$ with probability $\Pr[R'_{u,v} = \perp] \le \gamma$. We define a new random variable $l'_{u,v}$ to be $q'_{u,v,j}$ where $j = \min\{j : q'_{u,v,j} \ne \perp\}$; otherwise we define $l'_{u,v} = \perp$ if all $q_{u,v,j} = \perp$ for all $j$. If $l'_{u,v} = \perp$, then we set $F'_{u,v} \leftarrow \perp$. Otherwise $F'_{u,v} \leftarrow F_{u,v}$.

**Property: $\{F'_{u,v}\}_{u,v}$ are subdistributions of $\{F_{u,v}\}_{u,v}$.** First, it is clear that for $x \ne \perp$ we have $\Pr[F'_{u,v} = x] = \Pr[F_{u,v} = x, l'_{u,v} \ne \perp] \le \Pr[F_{u,v} = x]$. Furthermore, we have that $F'_{u,v} = \perp$ only when all $r$ paths $\{q'_{u,v,j}\}_j$ are sampled as $\perp$, which happens with probability at most $\gamma^r$, therefore $\Pr[F'_{u,v} = \perp] \le \gamma^r$. We conclude that (the distribution of) $F'_{u,v}$ is a $(\gamma^r)$-subdistribution of (the distribution of) $F_{u,v}$.

**The collection of subdistributions $\{l'_{s,t}\}_{s,t\in V(G)}$ has small congestion.** Each $l'_{s,t}$ defines a distribution over paths connecting $s$ and $t$, or $\perp$. We argue that

$$\text{cong}_G\Big(\mathbb{E}\left[\sum_{s,t} \text{flow}(l'_{s,t}) \cdot \mathcal{D}_{s,t}\right]\Big) \le \alpha \cdot \frac{1}{1-\gamma} \cdot \text{opt}^{(h)}(\mathcal{D}). \quad (3)$$

To this end, we consider the event $l'_{s,t} \ne \perp$. An equivalent process of sampling $l'_{s,t}$ is the following: sample a path from $R'_{s,t}$ and repeat until the path is not $\perp$. Such a rejection sampling is clearly equivalent to sampling from the conditional distribution $R'_{s,t} \mid R'_{s,t} \ne \perp$. Therefore,

$$\begin{aligned}
\mathbb{E}[\text{flow}(l'_{s,t})] &\le (\Pr[R'_{s,t} \ne \perp])^{-1} \cdot \mathbb{E}[\text{flow}(R'_{s,t})] \\
&\le \frac{1}{1-\gamma} \cdot \mathbb{E}[\text{flow}(R'_{s,t})].
\end{aligned}$$

Therefore, $\mathbb{E}[\sum_{s,t} \text{flow}(l'_{s,t}) \cdot \mathcal{D}_{s,t}] \leq \frac{1}{1-\gamma} \cdot \mathbb{E}[\sum_{s,t} \text{flow}(R'_{s,t}) \cdot \mathcal{D}_{s,t}]$. Finally, due to the congestion approximation property of $R'$, we have that $\text{cong}_G(\mathbb{E}[\sum_{s,t} \text{flow}(R'_{s,t}) \cdot \mathcal{D}_{s,t}]) \leq \alpha \cdot \frac{1}{1-\gamma} \cdot \text{opt}^{(h)}(\mathcal{D})$.

**Property: congestion of subdistributions.** We now argue about $\mathbb{E}[\sum_{s,t} \text{flow}(F'_{s,t}) \cdot \mathcal{D}_{s,t}]$. First, we introduce some notation: we remind the reader that $F_{s,t}$ is drawn from (an embedding from) $\mathcal{T} \mid S_{u,v} \subseteq V(T)$. On the other hand, we define $(T, T^G)$ to be an independent random variable drawn from $\mathcal{T}$. Finally, we define an event $EV := \{l'_{s,t} \neq \perp, S_{u,v} \subseteq V(T)\}$.

$$\begin{aligned}\mathbb{E}[\text{flow}(F'_{s,t})] &= \mathbb{E}[\mathbb{I}[l'_{s,t} \neq \perp] \cdot \text{flow}(F_{s,t})] \\ &\leq \mathbb{E}\left[\mathbb{I}[EV] \cdot \text{flow}(T^G_{s,t})\right] / \Pr[S_{u,v} \subseteq V(T)] \\ &\leq 2 \cdot \mathbb{E}\left[\mathbb{I}[EV] \cdot \text{flow}(T^G_{s,t})\right]\end{aligned}$$

The last inequality follows from $\Pr[S_{u,v} \subseteq V(T)] \geq 1/2$ which is a result of a simple union bound, $|S_{u,v}| \leq rh\beta$, and the exclusion probability of $\mathcal{T}$ being at most $1/(2rh\beta)$.

For each $s, t \in V(G)$ we introduce a random flow variable $C_{s,t} \in \mathbb{R}^{E(G)}_{\geq 0}$ as follows. In the event $EV$ we consider the nodes on the path $l'_{s,t} = (s = v_0, v_1, \ldots, v_{\ell-1}, v_\ell = t)$ and assign

$$C_{s,t} := \mathbb{I}[EV] \cdot \sum_{i=0}^{\ell-1} \text{flow}(T^G_{v_i,v_{i+1}}) \geq \mathbb{I}[EV] \cdot \text{flow}(T^G_{s,t}). \tag{4}$$

Note that the right-hand side inequality of Equation (4) holds because in any partial tree embedding $\text{flow}(T^G_{a,c}) \leq \text{flow}(T^G_{a,b}) + \text{flow}(T^G_{b,c})$ for all $a, b, c \in V(T)$. Therefore,

$$\begin{aligned}\sum_{s,t} \mathbb{E}[\text{flow}(F'_{s,t}) \cdot \mathcal{D}_{s,t}] &\leq 2 \sum_{s,t} \mathbb{E}\left[\mathbb{I}[EV] \cdot \text{flow}(T^G_{s,t}) \cdot \mathcal{D}_{s,t}\right] \\ &\leq 2 \sum_{s,t} \mathbb{E}[C_{s,t}] \cdot \mathcal{D}_{s,t}. \end{aligned}\tag{5}$$

On the other hand, we now give an upper bound for $\sum_{s,t} \mathbb{E}[C_{s,t} \cdot \mathcal{D}_{s,t}]$.

$$\begin{aligned}&\sum_{s,t} \mathbb{E}[C_{s,t} \mathcal{D}_{s,t}] \\ &= \sum_{s,t} \mathcal{D}_{s,t} \cdot \mathbb{E}\left[\mathbb{I}[EV] \cdot \sum_{i=0}^{\ell-1} \text{flow}(T^G_{v_i,v_{i+1}})\right] \\ &= \sum_{s,t} \mathcal{D}_{s,t} \cdot \mathbb{E}\left[\mathbb{I}[EV] \cdot \sum_{\{u,v\} \in E(G)} \text{flow}(l'_{s,t}, \{u,v\}) \cdot \text{flow}(T^G_{u,v})\right] \\ &= \sum_{\{u,v\} \in E(G)} \mathbb{E}\left[\text{flow}(T^G_{u,v}) \cdot \sum_{s,t} \mathcal{D}_{s,t} \cdot \mathbb{I}[EV] \cdot \text{flow}(l'_{s,t}, \{u,v\})\right] \\ &\leq \sum_{\{u,v\} \in E(G)} \mathbb{E}[\mathbb{I}[u, v \in V(T)] \cdot \text{flow}(T^G_{u,v})] \cdot \\ &\qquad \sum_{s,t} \mathcal{D}_{s,t} \cdot \mathbb{E}\left[\text{flow}(l'_{s,t}, \{u,v\})\right] \\ &\leq \sum_{\{u,v\} \in E(G)} \mathbb{E}[\mathbb{I}[u, v \in V(T)] \cdot \text{flow}(T^G_{u,v})] \cdot \\ &\qquad \frac{\alpha}{1-\gamma} \cdot \text{opt}^{(h)}(\mathcal{D}) \cdot c_{\{u,v\}} \qquad \text{(Equation (3))} \\ &= \frac{\alpha}{1-\gamma} \cdot \text{opt}^{(h)}(\mathcal{D}) \cdot \\ &\qquad \sum_{\{u,v\} \in E(G)} c_{\{u,v\}} \cdot \mathbb{E}[\mathbb{I}[u, v \in V(T)] \cdot \text{flow}(T^G_{u,v})] \\ &= \frac{\alpha}{1-\gamma} \cdot \text{opt}^{(h)}(\mathcal{D}) \cdot O(\log n \cdot \log(rh\beta \log n)) \cdot \vec{c_G}\end{aligned}$$

Combining the above with Equation (5) we get that

$$\begin{aligned}&\text{cong}_G(\sum_{s,t} \mathbb{E}[\text{flow}(F'_{s,t}) \cdot \mathcal{D}_{s,t}]) \\ &\quad \leq O(\frac{\alpha}{1-\gamma} \cdot \log n \cdot \log(rh\beta \log n)) \cdot \text{opt}^{(h)}(\mathcal{D}).\end{aligned}$$ □

We combine the results that we developed so far.

Corollary 4.5. *Every (complete) capacitated graph $G = (V, E, c)$ and every $h \geq 1, r = O(1)$, there exists an $h$-hop $(n^{-r})$-subflow routing for $G$ with congestion approximation $O(\log^2 n \cdot \log^2(h \log n))$ and hop stretch $O(\log^7 n)$.*

Proof. There exists a $\mathcal{D}^{(1)}$-router $\mathcal{T}_1$ on $G$ with dilation $O(h \log^3 n)$, exclusion probability $1/(4h)$, and congestion $O(\log n \log(h \log n))$ (via Lemma 3.7).

Define $R := \{R_{s,t}\}_{s,t \in V(G)}$ with $R_{s,t} := T^G_{s,t}$, where $(T, T^G) \sim \mathcal{T}_1$. Applying Lemma 4.3), we conclude that $R$ is an $h$-hop $\frac{1}{2}$-subflow routing with congestion $O(\log n \log(h \log n))$ and hop stretch $O(\log^3 n)$.

Finally, correcting the failures in the subflow routing using Lemma 4.4 by setting $r_{(\textit{Lemma 4.4})} := r \log_2 n = O(\log n)$, we construct an $h$-hop $(n^{-r})$-subflow routing with congestion approximation $O(\log^2 n \cdot \log^2(h \log n))$, hop stretch $O(\log^7 n)$. □

## 4.3 Putting it Together: Non-Subflow Routing on General Graphs

In this section, we prove our main result by combining all of the above. On a high-level, the main technical contribution of this section is to (1) completely eliminate failures, and to (2) extend the results from complete capacitated graphs to general capacitated graphs. However, both of these issues can be resolved in the following way.

(1) A $n^{-O(1)}$-fraction of failures can readily be ignored since they contribute an insignificant amount to the congestion.
(2) We can "complete" a general capacitated graph into its completed counterpart by converting "non-edges" to edges of sufficiently small capacity $n^{-O(1)}$. We construct the hop-constrained oblivious routing $R$ on the completed graph. Note that $R$ is supported (with small probability) over non-edges of the original graph. However, we can easily condition on these paths not using non-edges, which will only insignificantly increase the congestion.

Theorem 3.1. *For every (general) capacitated graph $G = (V, E, c)$ and every $h \geq 1$, there exists an $h$-hop oblivious routing with hop stretch $O(\log^7 n)$ and congestion approximation $O(\log^2 n \cdot \log^2(h \log n))$.*

Proof. For simplicity of notation, we extend the definition of flow (which maps paths to their flows in $\mathbb{R}^{E(G)}_{\geq 0}$) to $\text{flow}(\perp) = \vec{0}$. Due

to this, we can replace $\mathbb{I}[p \neq \perp] \cdot \text{flow}(p) = \text{flow}(p)$. Furthermore, let $c_{\min} = \min_{e \in E(G)} c_G(e)$, $c_{\max} = \max_{e \in E(G)} e_G(e)$, and let $C = O(1)$ be a sufficiently large constant. We remind the reader that because the capacities are polynomially bounded we have $c_{\min} \geq n^{-O(1)}$ and $c_{\min} \leq n^{O(1)}$.

**Completing the graph.** We first construct a "completed" capacitated graph $H = (V(G), \binom{V(G)}{2}, c_H)$ where $c_H(e) := c_G(e)$ if $e \in E(G)$, or $c_H(e) := c_{\min} \cdot n^{-C}$ if $e \in E(H) \setminus E(G)$. Due to this choice, for any demand $\mathcal{D}$, we have that $\text{opt}_H^{(h)}(\mathcal{D}) \geq \frac{1}{n^2 c_{\max}} \cdot \sum_{s,t} \mathcal{D}_{s,t}$ since we are pushing $\sum_{s,t} \mathcal{D}_{s,t}$ units of flow across at most $n^2$ edges of capacity of at most $c_{\max}$. Therefore, we conclude that $\text{opt}_H^{(h)}(\mathcal{D}) \geq n^{-O(1)} \cdot \sum_{s,t} \mathcal{D}_{s,t}$.

For the rest of the proof let $R = \{R_{s,t}\}_{s,t \in V(G)}$ be an $h$-hop $(n^{-2C})$-subflow routing on $H$ with congestion approximation $\alpha := O(\log^2 n \cdot \log^2(h \log n))$ and hop stretch $O(\log^7 n)$. Furthermore, we can assume that the support of $R_{s,t}$ is over simple paths since we can always simplify each path without increasing the congestion.

**$R$ has good congestion approximation on $H$ without subflows.** Fix a demand $\mathcal{D}$ and let $R' = \{R'_{s,t}\}_{s,t \in V(G)}$ be the collection of $(n^{-2C})$-subdistributions of $R = \{R_{s,t}\}_{s,t}$ with respect to $\mathcal{D}$. We also denote by $R_{s,t}$ and $R'_{s,t}$ the random variables (drawn from the distribution of the same name), coupled so that $R'_{s,t} \in \{R_{s,t}, \perp\}$. With this notation, we now show that the routing scheme $R$ achieves a good congestion approximation on (all) $\mathcal{D}$.

$$
\begin{aligned}
&\text{cong}_H\Big(\sum_{s,t} \mathbb{E}[\text{flow}_H(R_{s,t}) \cdot \mathcal{D}_{s,t}]\Big) \\
&\leq \text{cong}_H\left(\sum_{s,t} \mathbb{E}[\text{flow}_H(R'_{s,t}) \cdot \mathcal{D}_{s,t}]\right) && (6)\\
&\quad + \text{cong}_H\left(\sum_{s,t} \mathbb{E}[\mathbb{I}[R'_{s,t} = \perp] \, \text{flow}_H(R_{s,t}) \cdot \mathcal{D}_{s,t}]\right) \\
&\leq \alpha \cdot \text{opt}_H^{(h)}(\mathcal{D}) + \text{cong}_H\left(\sum_{s,t} \mathbb{E}[\mathbb{I}[R'_{s,t} = \perp] \, \text{flow}_H(R_{s,t}) \cdot \mathcal{D}_{s,t}]\right) \\
&\leq \alpha \cdot \text{opt}_H^{(h)}(\mathcal{D}) + \Pr[R'_{s,t} = \perp] \cdot (n^C / c_{\min}) \cdot \sum_{s,t} \mathcal{D}_{s,t} && (7)\\
&\leq \alpha \cdot \text{opt}_H^{(h)}(\mathcal{D}) + n^{-2C} \cdot \frac{n^C}{n^{O(1)}} \cdot n^{O(1)} \cdot \sum_{s,t} \mathcal{D}_{s,t} \\
&\leq \alpha \cdot \text{opt}_H^{(h)}(\mathcal{D}) + \text{opt}_H^{(h)}(\mathcal{D}) && (8)\\
&\leq 2\alpha \cdot \text{opt}_H^{(h)}(\mathcal{D})
\end{aligned}
$$

Equation (7) holds because each "bad" path $R_{s,t}$ (i.e., $R_{s,t}$ when $R'_{s,t} = \perp$) incurs at most $(\min_{e \in E(H)} c_H(e))^{-1} \leq (c_{\min} \cdot n^{-C})^{-1}$ congestion per each unit of demand (remember that we can assume paths are simple). Equation (8) holds for sufficiently large $C$ because $n^{-O(1)} \cdot \sum_{s,t} \mathcal{D}_{s,t} \leq \text{opt}_H^{(h)}(\mathcal{D})$ (as argued before). With this calculation, we conclude that $R$ is an $h$-hop oblivious routing for $H$ with hop stretch $O(\log^7 n)$ and congestion approximation $O(\alpha) = O(\log^2 n \cdot \log^2(h \log n))$.

**Routing scheme for the general graph.** We now adapt the routing scheme $R = \{R_{s,t}\}_{s,t \in V(G)}$, which is defined on the completed graph $H$, to a new routing scheme $F = \{F_{s,t}\}_{s,t \in V(G)}$ which is valid on the original (general) graph $G$. We simply define $F_{s,t}$ be $R_{s,t}$ conditioned on the sampled path traversing only edges in $E(G)$. We denote this conditional distribution with $F_{s,t} = R_{s,t} \mid R_{s,t} \subseteq G$.

Fix $s, t \in V(G)$ such that there exists an $h$-hop path in $G$ between $s$ and $t$. Our aim is to bound the probability that $R_{s,t}$ uses edges not in $E(G)$. Let $\mathcal{D}^{(s,t)}$ be the demand that has a single request between $s$ and $t$; i.e., $\mathcal{D}^{(s,t)}_{u,v} := \mathbb{I}[(u,v) = (s,t)]$. Since there is an $h$-hop path between $s$ and $t$ with edges of capacities at least $c_{\min}$ we have $\text{opt}_G^{(h)}(\mathcal{D}^{(s,t)}) \leq \frac{1}{c_{\min}} \leq n^{O(1)}$. Fix an edge $e \in E(H) \setminus E(G)$, i.e., that does not exist in $G$. By the assumption that $R$ is an $h$-hop oblivious routing for $H$ with congestion stretch $O(\alpha) \leq n^{O(1)}$, its routing of $\mathcal{D}^{(s,t)}$ on $H$ is $n^{O(1)}$-competitive, therefore:

$$
\begin{aligned}
\Pr[e \in R_{s,t}] &\leq \mathbb{E}[\text{flow}(R_{s,t}, e)] \\
&\leq n^{O(1)} \cdot \text{opt}_H^{(h)}(\mathcal{D}) \cdot c_H(e) \leq n^{O(1)-C}.
\end{aligned}
$$

Union-bounding over all $E(H) \setminus E(G)$ we have that

$$
\begin{aligned}
\Pr[R_{s,t} \subseteq G] &\leq |E(H) \setminus E(G)| \Pr[e \in R_{s,t}] \\
&\leq n^2 \cdot n^{O(1)-C} \leq 1/2 \qquad \text{(for a sufficiently large } C\text{)}.
\end{aligned}
$$

Fix an arbitrary demand $\mathcal{D}$. If there exists $s, t \in V(G)$ such that $\mathcal{D}_{s,t} > 0$, but there is no $h$-hop path between them, then $\text{opt}_G^{(h)}(\mathcal{D}) = \infty$ and the claim is trivial. If this is not the case, we bound the congestion of the routing scheme $F_{s,t}$ (which is conditioned on going only over edges in $G$):

$$
\begin{aligned}
&\text{cong}_G\left(\sum_{s,t} \mathbb{E}[\text{flow}_G(F_{s,t}) \cdot \mathcal{D}_{s,t}]\right) \\
&\leq \text{cong}_G\left(\sum_{s,t} \mathbb{E}[\mathbb{I}[R_{s,t} \subseteq G] \cdot \text{flow}_G(R_{s,t}) \cdot \mathcal{D}_{s,t}]\right) / \Pr[R_{s,t} \subseteq G] \\
&\leq 2 \cdot \text{cong}_H\left(\sum_{s,t} \mathbb{E}[\text{flow}_H(R_{s,t}) \cdot \mathcal{D}_{s,t}]\right) && (9)\\
&\leq O(\alpha) \cdot \text{opt}_H^{(h)}(\mathcal{D}) \\
&\leq O(\alpha) \cdot \text{opt}_G^{(h)}(\mathcal{D}). && (10)
\end{aligned}
$$

Equation (9) follows because all paths in the preceding equation go only over edges in $G$; both $G$ and $H$ agree on the capacities of such edges. Equation (10) follows from $c_H(e) \geq c_G(e)$, hence $\text{opt}_H^{(h)}(\mathcal{D}) \leq \text{opt}_G^{(h)}(\mathcal{D})$. We conclude that $F$ is an $h$-hop oblivious routing on $G$ with hop stretch $O(\log^7 n)$ and congestion approximation $O(\log^2 n \cdot \log^2(h \log n))$ for all demands $\mathcal{D}$. □

## ACKNOWLEDGEMENTS

This research was supported in part by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 853109 and grant agreement No. 949272), by NSF grants CCF-1527110, CCF-1618280, CCF-1814603, CCF-1910588, NSF CAREER award CCF-1750808, a Sloan Research Fellowship, and the Swiss National Foundation (project grant 200021-184735).

# A CONSTRUCTION OF $\mathcal{D}^{(1)}$-ROUTERS

Lemma 3.7. *For every (complete) capacitated graph $G$ and $0 < \varepsilon < 1/3$ there exists a $\mathcal{D}^{(1)}$-router with dilation $O(\frac{\log^3 n}{\varepsilon})$, exclusion probability $\varepsilon$, and congestion $O(\log n \cdot \log \frac{\log n}{\varepsilon})$.*

Proof. We write a linear program over $\mathcal{D}^{(1)}$-routers with the goal of minimizing the congestion $\alpha$ while satisfying the dilation and exclusion probability properties. Let $(T_1, T_1^G), (T_2, T_2^G), \ldots, (T_Q, T_Q^G)$ the (finite) set of possible partial tree embeddings of $G$ satisfying the dilation property, i.e., where $\mathrm{hop}(T_{u,v}^G) \leq O(\frac{\log^3 n}{\varepsilon})$ for all $u, v \in V(T)$. Given a vector $\lambda \in \{x \in \mathbb{R}_{\geq 0}^Q \mid \sum_j x_j = 1\}$ we denote with $\mathcal{T}(\lambda)$ the distribution over $\{(T_i, T_i^G)\}_i$ where $\Pr[\mathcal{T}(\lambda) = (T_i, T_i^G)] = \lambda_i$. Furthermore, let $\Lambda$ the set of vectors $\lambda = (\lambda_i)_i$ satisfying the exclusion probability property, i.e., $\Pr_{(T,\cdot)\sim\mathcal{T}(\lambda)}[v \in V(T)] \geq 1 - \varepsilon$ for all $v \in V(G)$.

We now present the linear program. Note that $\mathrm{cong}_G(x) \leq \alpha$ can be written as $x_e \leq \alpha \cdot c_e$ where $c_e$ is the capacity of an edge $e$ in $G$.

$$\begin{aligned}
&\min_{\alpha,\lambda} . \quad \alpha \\
&\text{such that} \quad \lambda \in \Lambda, \text{ and} \\
&\forall e \in E(G) \quad \sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot \mathbb{E}_{(T,T^G)\sim\mathcal{T}(\lambda)}\Big[ \\
&\qquad \mathbb{I}[u,v \in V(T)] \cdot \mathrm{flow}(T_{u,v}^G, e)\Big] \leq \alpha \cdot c_e
\end{aligned}$$

We dualize the linear program. Note that the primal can be written as $\min\{\alpha \mid M\lambda \leq \alpha, \lambda \in \Lambda\}$ for an appropriately chosen matrix $M$ since $\mathrm{cong}_G(x)$ is simply $\max\{x_e/c_e\}_e$. Therefore, we use the dualization formula

$$\min\{\alpha \mid M\lambda \leq \alpha, \lambda \in \Lambda\} = \max_{\ell \geq 0, 1^T\ell = 1} \min_{\lambda \in \Lambda} \ell^T M\lambda.$$

Using these values, we rewrite the right-hand side of the equation.

$$\begin{aligned}
&\ell^T M\lambda \\
&= \sum_{e\in E(G)} \ell_e \cdot \mathbb{E}_{(T,T^G)\sim\mathcal{T}(\lambda)}\Big[ \\
&\qquad \sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot \mathbb{I}[u,v \in V(T)] \cdot \mathrm{flow}(T_{u,v}^G, e)\Big]/c_e \\
&= \sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot \mathbb{E}_{(T,T^G)\sim\mathcal{T}(\lambda)}\Big[ \\
&\qquad \mathbb{I}[u,v \in V(T)] \sum_{e\in E(G)} \ell_e/c_e \cdot \mathrm{flow}(T_{u,v}^G, e)\Big] \\
&= \sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot \mathbb{E}_{(T,T^G)\sim\mathcal{T}(\lambda)}\Big[\mathbb{I}[u,v \in V(T)] \cdot w_{G'(\ell)}(T_{u,v}^G)\Big]
\end{aligned}$$

In the last line, we introduced a new weighted graph $G'(\ell)$ that is defined as having the same node set and edge set as $G$, while its weights $w_{G(\ell)}$ are set to $w_{G(\ell)}(e) = \ell_e/c_e \geq 0$. With this, we present the dual:

$$\begin{aligned}
&\max_{\beta,\ell} . \quad \beta \\
&\text{such that} \\
&\forall \lambda \in \Lambda \quad \sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot \mathbb{E}_{(T,T^G)\sim\mathcal{T}(\lambda)}\Big[ \\
&\qquad \mathbb{I}[u,v \in V(T)] \cdot w_{G'(\ell)}(T_{u,v}^G)\Big] \geq \beta \\
&\ell \geq 0, \sum_{e\in E(G)} \ell_e = 1
\end{aligned}$$

By inspecting to dual, we see that in order to show that the optimal value of the linear program is at most $\beta$, it is sufficient to show that for every distribution $(\ell_e)_{e\in E(G)}$ there exists a $\lambda \in \Lambda$ where $\ell^T M\lambda \leq \beta$. To this end, fix any $(\ell_e)_e$ and consider $G'(\ell)$ as defined above. Via Theorem 2.4, there exists a 1-hop partial embedding distribution $\mathcal{T}'$ with exclusion probability $\varepsilon$, hop stretch $O(\frac{\log^3 n}{\varepsilon})$, and expected distance stretch $O(\log n \cdot \log \frac{\log n}{\varepsilon})$, i.e.,

$$\begin{aligned}
&\mathbb{E}_{(T,T^G)\sim\mathcal{T}'}\Big[\mathbb{I}[u,v \in V(T)] \cdot w_{G'(\ell)}(T_{u,v}^G)\Big] \leq \\
&\qquad w_{G'(\ell)}(\{u,v\}) \cdot O(\log n \cdot \log \frac{\log n}{\varepsilon}).
\end{aligned}$$

Note that $\mathcal{T}'$ can be represented as $\mathcal{T}(\lambda')$ for some $\lambda' \in \Lambda$ since the distribution satisfies the exclusion property and each embedding in the support satisfies the dilation properties (due to the hop stretch). Therefore, for $\lambda = \lambda'$ we have:

$$\begin{aligned}
\ell^T M\lambda &= \sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot \\
&\qquad \cdot \mathbb{E}_{(T,T^G)\sim\mathcal{T}(\lambda')}\Big[\mathbb{I}[u,v \in V(T)] \cdot w_{G'(\ell)}(T_{u,v}^G)\Big] \\
&\leq \sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot w_{G'(\ell)}(\{u,v\}) \cdot O(\log n \cdot \log \frac{\log n}{\varepsilon}) \\
&= O(\log n \cdot \log \frac{\log n}{\varepsilon}) \sum_{\{u,v\}\in E(G)} c_{\{u,v\}} \cdot \frac{\ell_{\{u,v\}}}{c_{\{u,v\}}} \\
&= O(\log n \cdot \log \frac{\log n}{\varepsilon})
\end{aligned}$$

In other words, we conclude that the optimal value of the linear program is at most $O(\log n \cdot \log \frac{\log n}{\varepsilon})$, showing that there exists a distribution over partial tree embeddings of $G$ that are a $\mathcal{D}^{(1)}$-router with dilation $O(\frac{\log^3 n}{\varepsilon})$ (implied by $\lambda \in \Lambda$), exclusion probability $\varepsilon$ (implied by $\lambda \in \Lambda$), and congestion $O(\log n \cdot \log \frac{\log n}{\varepsilon})$ (optimal linear program value).

**Remark.** We note that the proof above leaves out the following detail. Theorem 2.4 assumes that the weighted graph $G'(\ell)$ has polynomially-bounded weights. However, the straightforward dualization process offers no such guarantees. However, this is easily remedied by standard arguments. For instance, the framework of multiplicative weights [1] shows that one can 2-approximate the value of a linear program by only considering duals with polynomially-bounded coefficients. □

## B COMPUTATIONAL ASPECTS

This paper primarily focuses on the existence of hop-constrained oblivious routings without talking about how to efficiently construct them. However, it is relatively straightforward to give a randomized construction of such routings in polynomial time. Examining Algorithm 1 that constructs these routings, we observe that all of the steps involved are straightforward to implement in polynomial time except constructing $\mathcal{D}^{(1)}$-routers. Moreover, the existence of $\mathcal{D}_1$-routers is proven via strong duality, making it less clear how to make the result algorithmic. However, the standard technique of multiplicative weights [1] allows us to do just that and recover a polynomial-time sampling of $\mathcal{D}_1$-routers, and therefore also of hop-constrained oblivious routings. We note that (in the framework of multiplicative weights), the width of the oracle corresponds to the (maximum) expectation that a distribution of partial tree embeddings incurs over some edge $e$ when routing the $\mathcal{D}^{(1)}$ demand. Since the capacities are polynomially-bounded, we conclude that the width of the oracle is polynomial.

We leave it as an open problem whether hop-constrained oblivious routings can be constructed in almost-linear, $O(m^{1+o(1)})$, time.

## REFERENCES


[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. 2012. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing* 8, 1 (2012), 121–164.

[2] James Aspnes, Costas Busch, Shlomi Dolev, Panagiota Fatourou, Chryssis Georgiou, Alexander A Shvartsman, Paul G Spirakis, and Roger Wattenhofer. 2006. Eight Open Problems in Distributed Computing. *Bulletin of the EATCS* 90 (2006), 109–126.

[3] Yossi Azar, Edith Cohen, Amos Fiat, Haim Kaplan, and Harald Räcke. 2004. Optimal oblivious routing in polynomial time. *J. Comput. System Sci.* 69, 3 (2004), 383–394.

[4] Yair Bartal. 1996. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of 37th Conference on Foundations of Computer Science*. IEEE, 184–193.

[5] Yair Bartal. 1998. On approximating arbitrary metrices by tree metrics. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 161–168.

[6] Marcin Bienkowski, Miroslaw Korzeniowski, and Harald Räcke. 2003. A practical algorithm for constructing oblivious routing schemes. In *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*. 24–33.

[7] Costas Busch and Malik Magdon-Ismail. 2010. Optimal oblivious routing in hole-free networks. In *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Springer, 421–437.

[8] Costas Busch, Malik Magdon-Ismail, and Jing Xi. 2005. Oblivious routing on geometric networks. In *Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures*. 316–324.

[9] Costas Busch, Malik Magdon-Ismail, and Jing Xi. 2008. Optimal oblivious path selection on the mesh. *IEEE Trans. Comput.* 57, 5 (2008), 660–671.

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

[11] Matthias Englert and Harald Räcke. 2009. Oblivious routing for the Lp-norm. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 32–40.

[12] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. 2003. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. 448–455.

[13] Anupam Gupta and Jochen Könemann. 2011. Approximation algorithms for network design: A survey. *Surveys in Operations Research and Management Science* 16, 1 (2011), 3–20.

[14] Bernhard Haeupler, D Ellis Hershkowitz, and Goran Zuzic. 2020. Tree Embeddings for Hop-Constrained Network Design. *arXiv preprint* (2020).

[15] Chris Harrelson, Kirsten Hildrum, and Satish Rao. 2003. A polynomial-time tree decomposition to minimize congestion. In *Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*. 34–43.

[16] Frank Thomson Leighton, Bruce M Maggs, and Satish B Rao. 1994. Packet routing and job-shop scheduling in $O$(congestion+ dilation) steps. *Combinatorica* 14, 2 (1994), 167–186.

[17] Michael O Rabin. 1989. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM (JACM)* 36, 2 (1989), 335–348.

[18] Harald Racke. 2002. Minimizing congestion in general networks. In *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings*. IEEE, 43–52.

[19] Harald Räcke. 2008. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 255–264.

[20] Harald Räcke. 2009. Survey on oblivious routing strategies. In *Conference on Computability in Europe*. Springer, 419–429.

[21] Christian Scheideler. 2006. *Universal routing strategies for interconnection networks*. Vol. 1390. Springer.

[22] Aravind Srinivasan and Chung-Piaw Teo. 2001. A constant-factor approximation algorithm for packet routing and balancing local vs. global criteria. *SIAM J. Comput.* 30, 6 (2001), 2051–2068.

[23] Eli Upfal. 1984. Efficient schemes for parallel communication. *Journal of the ACM (JACM)* 31, 3 (1984), 507–517.

[24] Leslie G Valiant and Gordon J Brebner. 1981. Universal schemes for parallel communication. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*. 263–277.

[25] David P Williamson and David B Shmoys. 2011. *The design of approximation algorithms*. Cambridge university press.