New Lower Bounds for Oblivious Routing in Undirected Graphs

Mohammad T. Hajiaghayi[*]   Robert D. Kleinberg[*‡]   Tom Leighton[*§]   Harald Räcke[¶]

Abstract

Oblivious routing algorithms for general undirected networks were introduced by Räcke, and this work has led to many subsequent improvements and applications. Räcke showed that there is an oblivious routing algorithm with polylogarithmic competitive ratio (with respect to edge congestion) for any undirected graph. However, there are directed networks for which the competitive ratio is in $\Omega(\sqrt{n})$.

To cope with this inherent hardness in general directed networks, the concept of oblivious routing with demands chosen randomly from a known demand distribution was introduced recently. Under this new model, $O(\log^2 n)$-competitiveness with high probability is possible in general directed graphs.

However, it remained an open problem whether or not the competitive ratio, under this new model, could also be significantly improved in undirected graphs. In this paper, we rule out this possibility by providing a lower bound of $\Omega(\frac{\log n}{\log \log n})$ for the multicommodity case and $\Omega(\sqrt{\log n})$ for the single-sink case for oblivious routing in a random demand model.

We also introduce a natural candidate model for evaluating the throughput of an oblivious routing scheme which subsumes all suggested models for the throughput of oblivious routing considered so far. In this general model, we first prove a lower bound $\Omega(\frac{\log n}{\log \log n})$ for the competitive ratio of any oblivious routing scheme. Interestingly, the graphs that we consider for the lower bound in this case are expanders, for which we also obtain a lower bound $\Omega(\frac{\log n}{\log \log n})$ on the competitive ratio of congestion based oblivious routing with adversarial demands.

1 Introduction

The concept of oblivious routing aims at developing routing algorithms that base their routing decisions only on local knowledge and that therefore can be implemented very efficiently in a distributed environment. In this paper we study oblivious routing algorithms that aim to minimize the congestion, which is defined as the maximum relative load of a network edge. (The relative load of an edge is the number of routing paths traversing the edge divided by the capacity of the edge.) We also consider maximizing throughput, which is the total flow that we can deliver from sources to sinks without violating any edge capacity constraints. In this paper, we develop a natural model for proving lower bounds on the competitive ratio of oblivious routing with respect to throughput.

Traditionally, for an oblivious routing algorithm the routing path chosen between a source s and a target t may only depend on s and t. Valiant and Brebner [16] show e.g. how to obtain efficient routing algorithms for the hypercube in this scenario. Their algorithm obtains a *competitive ratio* of $O(\log n)$, i.e., the congestion of their algorithm is always within a logarithmic factor of the best possible congestion. Later Räcke [14] obtained oblivious routing schemes with polylogarithmic competitive ratio for general undirected graphs; this competitive ratio was subsequently improved by Harrelson, Hildrum, and Rao [10]. However, a serious drawback in this line of research is that already for very simple directed graphs it is not possible to obtain a polylogarithmic competitive ratio (see [3]).

Hajiaghayi et al. [8] introduced a new model in which the demands between node-pairs are not assumed to be a worst-case input for the given oblivious routing algorithm (as in the standard competitive analysis) but are drawn from a demand distribution that is known in advance. In many practical applications this assumption is justified. They show that for any directed graph, if the demands for different node-pairs are independent, there is an oblivious routing algorithm whose congestion is within $O(\log^2 n)$ of the optimum congestion, with high probability. This gives an improvement over the worst case analysis where in general an oblivious routing algorithm cannot obtain competitive ratio better than $\Omega(\sqrt{n})$.

A problem left open by this work is whether, under this new model, one can also improve the competitive ratio in

---

[*]Department of Mathematics and Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA 02139, U.S.A., Emails: {hajiagha,rdk,ftl}@theory.csail.mit.edu.

[†]Department of Computer Science, Cornell University, Ithaca, NY 14853.

[‡]Supported by a Fannie and John Hertz Foundation Fellowship.

[§]Akamai Technologies, Eight Cambridge Center, Cambridge, MA 02139, U.S.A.

[¶]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A., Email: harry@cs.cmu.edu.

undirected graphs (to possibly $O(1)$). We rule out such a possibility by constructing a counterexample. In doing so, we demonstrate that for undirected graphs there is no strong qualitative difference between average-case and worst-case performance of oblivious routing, in contrast to the strong qualitative difference ($O(\log^2 n)$ versus $\Omega(\sqrt{n})$) for directed graphs as observed in [8].

1.1 Related Work. The idea of selecting routing paths oblivious to the traffic in the network has been intensively studied for special network topologies, since such algorithms allow for very efficient implementations due to their simple structure. Valiant and Brebner [16] initiate the worst case theoretical analysis for oblivious routing on the hypercube. They design a randomized packet routing algorithm that routes any permutation in $O(\log n)$ steps. This result gives a virtual circuit routing algorithm that obtains a competitive ratio of $O(\log n)$ with respect to edge-congestion.

In [14] it was shown that there is an oblivious routing algorithm with polylogarithmic competitive ratio (w.r.t. edge-congestion) for any undirected graph. However, this result was non-constructive in the sense that only an exponential-time algorithm was given for constructing the routing scheme. This issue was subsequently addressed by Bienkowski et al. [4], Harrelson et al. [10], and Azar et al. [3]; in fact the latter paper shows that the optimum oblivious routing scheme, i.e., the scheme that guarantees the best possible competitive ratio, can be constructed in polynomial time by using a linear program. This result holds for edge-congestion, node congestion and in general directed and undirected graphs. Furthermore, they show that there are directed graphs such that every oblivious routing algorithm has a competitive ratio of $\Omega(\sqrt{n})$.

The method used by Azar et al. does not enable one to derive general bounds on the competitive ratio for certain types of graphs. Another disadvantage of [3] was that it did not give a polynomial-time construction of the *hierarchical tree decomposition* used in [14], which has proven to be useful in many applications (see e.g. [1, 6, 13]). A polynomial time algorithm for this problem was independently given by [4] and [10]. Whereas the first result shows a slightly weaker competitive ratio for the constructed hierarchy than the non-constructive result in the original paper, the second paper by Harrelson, Hildrum and Rao has even improved the competitive ratio to $O(\log^2 n \log \log n)$. This is currently the best known bound for oblivious routing in general undirected graphs.

Hajiaghayi, Kleinberg, Leighton and Räcke [9] considered the problem of oblivious routing for directed graphs with a single sink. They show that one cannot obtain a competitive ratio better than $\Omega(\sqrt{n})$ and that one can obtain competitive ratio $\Omega(\sqrt{n} \log n)$. They also demonstrate the first non-trivial upper bounds for competitive ratio of oblivious routing in undirected networks with node capacities and general directed networks. However, these bounds are still far away from the polylogarithmic ratio that can be obtained on undirected graphs.

To cope with the inherent hardness of oblivious routing for general directed networks, Hajiaghayi, Kim, Leighton and Räcke [8] recently introduced the concept of oblivious routing with demands chosen randomly from a known demand distribution (the demands for different source-target pairs are independent in this setting). Under this new model, they show that it is possible to be $O(\log^2 n)$-competitive with high probability in general directed graphs.

For the cost-measure of throughput instead of congestion, Räcke and Rosen [15] give a distributed online call control algorithm which is closely related to oblivious throughput maximization in undirected graphs. Awerbuch et al. [2] establish nearly tight upper and lower bounds on the performance of oblivious routing schemes in directed bipartite graphs, in terms of throughput. They show that the performance gap between the optimal and the oblivious solution is inherently polynomial even in this restricted graph class.

1.2 Our Results. We show that there are undirected graphs for which any oblivious routing algorithm has a competitive ratio of $\Omega(\frac{\log n}{\log \log n})$, with high probability. For the case in which all source-target pairs share a common destination (single-sink case) we show that there are graphs (and corresponding demand distributions) such that the congestion produced by an oblivious algorithm is with high probability a factor of $\Omega(\sqrt{\log n})$ away from the optimum congestion. Thus, we rule out the possibility of improving the competitive ratio to a constant factor in undirected graphs in the random demand model. This answers a main open problem in [8].

One main difficulty in our proofs is that the demand distribution is assumed to assign demands independently to different source-target pairs. For the case where demands between pairs are not independent it is much easier to develop lower bounds, and e.g. the lower bound of $\Omega(\log n)$ developed by Maggs et al. [12] on the 2-dimensional grid applies to this scenario.

The other main difficulty stems from the fact that we consider undirected graphs and thus we cannot force the flow into a particular direction.

As a second objective we analyze the throughput that a routing protocol that is based on oblivious path selection can obtain. Here the goal is to satisfy a maximum number of demand-pairs (i.e., route their demand) without violating the edge-capacities in the graph (in contrast to the congestion-model where we route all demands and minimize the factor by which edge-capacities are violated). We introduce a natural candidate model for evaluating the throughput of an oblivious routing scheme, which subsumes all models considered

in prior work (including [2, 15]). In this general model, we prove a lower bound $\Omega(\frac{\log n}{\log \log n})$ for the competitive ratio of any oblivious routing scheme with respect to throughput. The exact definition of the model and the proof of this result is presented in Section 4.

It is worth mentioning that this lower bound also gives a new class of graphs for which there is an $\Omega(\frac{\log n}{\log \log n})$ lower bound on the competitive ratio of congestion based oblivious routing with adversarial demands.

1.3 Basic Definitions. Our graph terminology is as follows. We represent the network as a graph $G = (V, E)$, where V denotes the set of vertices (or nodes) and E denotes the set of edges. In general an edge may carry a weight that indicates the physical communication capacity of the corresponding network link. However, for the lower bounds that we obtain in this paper it is sufficient to consider only graphs in which all edges have uniform capacity. Therefore, we will assume that all edge capacities are uniform, in the following.

In this work we consider oblivious routing in a random demand model that was introduced in [8]. A (demand-dependent) oblivious routing algorithm specifies for every positive real value $r \in \mathbb{R}^+$ a flow f_{st}^r of value r from s to t in the network.[1] If the demand between s and t is r, it is routed according to f_{st}^r.

We assume that a demand matrix D specifying a demand for every source-target pair is chosen according to a probability distribution $\mathcal{D}$. We assume that $\mathcal{D}$ assigns demands independently to the demand pairs, i.e., the entry D_{st} is independent of the entry $D_{s't'}$.

We consider two different scenarios in this paper. In the first scenario the goal is to minimize the congestion of an oblivious routing algorithm, whereas in the second scenario we consider the throughput that can be obtained in an oblivious routing model. The precise definition of the throughput model can be found in Section 4. The following definitions apply to the congestion based model.

The congestion is defined, as follows. For a given demand-matrix D and a given routing algorithm, we define the *absolute load* of an edge as the amount of flow routed along this edge. The relative load is the absolute load of an edge divided by its capacity. The *edge-congestion* (or just *congestion*) is defined to be the maximum relative load of an edge.

We define $\mathrm{C}_{\mathrm{obl}}(D)$ to be the edge-congestion of the routing guided by the flow paths of oblivious routing for the demand matrix D. Let $\mathrm{C}_{\mathrm{opt}}(D)$ be the optimum edge-congestion for the demand-matrix D. We call the ratio

$\mathrm{C}_{\mathrm{obl}}(D)/\mathrm{C}_{\mathrm{opt}}(D)$ the *competitive ratio* for a demand matrix D.

The goal is to create an oblivious routing scheme (based on knowledge of the demand distribution $\mathcal{D}$) such that the competitive ratio is $O(1)$ with high (or at least constant) probability in undirected graphs, when the demands are chosen from the distribution. We show that this goal cannot be achieved: There are graphs and corresponding demand distributions such that any oblivious routing scheme has a large competitive ratio with high probability.

2 The general multicommodity case

In this section we construct an undirected graph G and a demand distribution $\mathcal{D}$ such that any oblivious routing algorithm is nearly always (i.e., with high probability according to the random choice of the demand) far away from an optimal solution in terms of congestion. We show that with high probability the ratio between the congestion created by an oblivious algorithm and the optimum congestion is $\Omega(\log n / \log \log n)$.

The construction of the network is as follows. Fix an odd integer ℓ, and integers k and n such that $n = \ell^k$ and $\ell = \Theta(\log^5 n)$ (i.e., $k = \Omega(\log n / \log \log n)$). The graph G for our lower bound is obtained by the following construction.

Start with an n-by-n grid in which all edges have uniform capacities. We add nodes and edges to this grid in the following recursive scheme. First, we add a super-source and a super-sink to the graph, and attach the source to all nodes in column $\lceil n/2 \rceil$ (center column) and the sink to all nodes in column n (rightmost column) via edges of uniform capacities. Now, consider two stripes of width n/ℓ around the center column. The left stripe S_L contains nodes $\{(x, y) \mid x \in \{\lceil \frac{n}{2} \rceil - \frac{n}{\ell}, \ldots, \lceil \frac{n}{2} \rceil - 1\}, y \in \{1, \ldots, n\}\}$ and the right stripe S_R contains nodes $\{(x, y) \mid x \in \{\lceil \frac{n}{2} \rceil + 1, \ldots, \lceil \frac{n}{2} \rceil + \frac{n}{\ell}\}, y \in \{1, \ldots, n\}\}$. We view both stripes as composed out of ℓ square grids of side-length ℓ^{k-1}. To each of these square grids we add sources and sinks recursively. We stop at sub-grids of side-length one.

In the following we refer to a grid that appears in the above recursive construction, and that has side length $n_i := \ell^{k-i+1}$ as a *(sub-)grid on level i*. The demand distribution $\mathcal{D}$ for our lower bound is defined as follows. A source in a grid on level $i > 1$ becomes *active* with probability $p = \frac{1}{\log n}$, and in this case it sends a demand of n_i to the corresponding target. An inactive source does not send anything. The source on level 1 is always active with demand n.

We prove that for this demand distribution any oblivious routing algorithm on G creates congestion $\Omega(k) = \Omega(\log n / \log \log n)$ with high probability, while an optimum algorithm only creates constant congestion, w.h.p. This gives the following theorem.

---
[1]Note that this definition would not be appropriate for developing upper bounds because it does not restrict the storage size required for encoding an oblivious routing algorithm. However, since we are interested in lower bounds we can use this definition.

THEOREM 2.1. *The competitive ratio of any oblivious routing algorithm on G with demand distribution $\mathcal{D}$ is $\Omega(\log n / \log \log n)$, with high probability.*

We start with the lower bound for the oblivious routing algorithm.

LEMMA 2.1. *For demand distribution $\mathcal{D}$, any oblivious routing algorithm on G produces a congestion of $\Omega(k) = \Omega(\log n / \log \log n)$ with high probability.*

Proof. Fix an oblivious routing scheme. If a node (x, y) lies in a sub-grid on level $k + 1$ (sub-grid of side-length 1) in the construction for G, then all nodes in the y-column lie in some level $k + 1$ sub-grid. This means that the union of all level $k + 1$ sub-grids forms a set of columns in the grid. In the following we call these columns the *essential* columns of the grid and nodes on essential columns are called *essential nodes* (essential nodes are in a sub-grid for every level of the recursion).

We prove that either the oblivious algorithm creates edge-congestion $\Omega(\log n)$ or with high probability there is an essential node v such that the oblivious routing algorithm sends demand $\Omega(k) = \Omega(\log n / \log \log n)$ over this node. Since the degree of v is constant we get that the edge-congestion on one of its adjacent edges must be $\Omega(\log n / \log \log n)$.

We use an inductive argument. The induction step is as follows.

CONDITION 2.1. *Given an* active *sub-grid M_i on level i such that the average load on essential nodes within M_i due to sources on levels $j < i$ is at least $\alpha_i := \frac{i-1}{4} - \frac{i-1}{\log n}$. Then, with high probability, there*

- *either exists a sub-grid $M_{i+1} \subset M_i$ with average load $\alpha_{i+1} = \frac{i}{4} - \frac{i}{\log n}$ on essential nodes (due to sources on levels $j < i + 1$), or*

- *there is a node inside M_i with load $\Omega(\log n)$.*

Proof. The sub-grid M_i is active, which means that the source of M_i sends a demand of n_i to its target. We first claim that this demand creates a high load on essential columns within M_i. Let C_i denote the number of essential columns that intersect M_i. Half of these columns lie in the stripe left to the source column and half of them lie right to the source column. Therefore, in order to reach the target, all the demand generated by the source of M_i has to cross at least $C_i/2$ essential columns. Moreover, most of the demand (say at least 50%) has to cross $C_i/2$ essential columns *inside* M_i, because all essential columns lie in a small stripe of width $2\frac{n_i}{\ell} \leq n_i / \log n$ (therefore if 50% of the demand (which is n_i) leaves M_i *before* crossing $C_i/2$ essential columns the load on vertical edges leaving M_i would be $\Omega(\log n)$).

Hence, we can assume that the average load on the $n_i \cdot C_i$ essential nodes inside M_i due to the source for M_i is at least $(\frac{C_i}{2} \cdot \frac{n_i}{2})/(C_i \cdot n_i) \geq 1/4$. This means that the average load on essential nodes inside M_i due to sources on levels $j < i + 1$ is at least $\alpha_i + \frac{1}{4}$.

It remains to show that there is an *active* sub-grid that has high average load. We show by a simple averaging argument that for many sub-grids of M_i the average load on essential nodes is at least $\alpha_{i+1} = \alpha_i + \frac{1}{4} - \frac{1}{\log n}$. Let $|V_{\text{ess}}|$ denote the number of essential nodes within a sub-grid. Assume for contradiction that no sub-grid has average load larger than $\log n$ (otherwise we are done) and that there are less than $a = \Theta(\log^2 n)$ sub-grids that have average load at least α_{i+1}. Then, the total load is less than $2\ell \cdot |V_{\text{ess}}| \cdot (\alpha_i + \frac{1}{4} - \frac{1}{\log n}) + a|V_{\text{ess}}| \cdot \log n < 2\ell \cdot |V_{\text{ess}}| \cdot (\alpha_i + \frac{1}{4})$, where $2\ell \gg a \log^2 n$ is the number of sub-grids. This is a contradiction since we assume that the average load is at least $\alpha_i + \frac{1}{4}$.

Hence, there exist at least $a = \Omega(\log^2 n)$ sub-grids with average load at least $\alpha_{i+1} = \alpha_i + \frac{1}{4} - \frac{1}{\log n}$ due to sources on levels $j < i + 1$. With high probability one of these sub-grids becomes active since the probability that a sub-grid becomes active is $\frac{1}{\log n}$. This proves the claim. $\square$

Note that the source on level 1 is always active. Therefore the whole grid meets the requirements for Claim 2.1. Applying the above lemma for k levels we obtain a sub-grid with side-length one and average load $\Omega(k)$ with high probability. Hence the congestion of the oblivious routing algorithm is $\Omega(\log n / \log \log n)$. $\square$

LEMMA 2.2. *With high probability an optimum algorithm can route the demands generated by $\mathcal{D}$ with constant congestion.*

Proof. We use a recursive routing scheme. On the first level we have to decide where to route the demand created by the source on level 1. Since the right stripe S_R separates the source from its target we have to route the demand through the sub-grids in S_R. The strategy for doing this is as follows. First, distribute the demand evenly among the left border node of *inactive* sub-squares. Then solve the following routing problems recursively for the sub-squares. Each active sub-square M_i only has to route its internal demand (demand for commodities for which source and target are in M_i). Inactive sub-squares, however, have to route their internal demand, and, in addition, they have to push the incoming flow from the left to the right border. We show by induction that both these routing problems can be solved. The induction step is as follows.

CONDITION 2.2. *Let $\lambda_i = (\frac{1}{1-2p})^{i-1}$. Suppose that for a sub-grid M_{i+1} on level $i + 1$ with $1 < i + 1 \leq k + 1$ we can solve the following routing problems:*

1. *active grid M_{i+1}:*
   An active sub-grid M_{i+1} can route all its internal demand with congestion at most $3\lambda_{k+1}$.

2. *inactive grid M_{i+1}:*
   An inactive sub-grid M_{i+1} can route all its internal demand and can route a flow of $\lambda_{i+1} \cdot n_{i+1}$ (evenly distributed among the n_{i+1} left border nodes) to an even distribution among the right border nodes with congestion $3\lambda_{k+1}$.

Then we can solve the corresponding routing problems for M_i with congestion $3\lambda_{k+1}$.

REMARK 2.1. Note that the above claim trivially holds for $i+1 = k+1$, since sub-grids on level $k+1$ form individual vertices. Now the general statement follows via induction from $i+1$ to i. This shows that the proof of Claim 2.2 directly implies the lemma, since the congestion $3\lambda_{k+1}$ is constant.

Proof of Claim 2.2. We start with the proof for an active sub-grid M_i. The probability p that a sub-grid becomes active is only $\frac{1}{\log n}$. Therefore with high probability only $2p\ell$ of the ℓ sub-grids in the stripe S_R are active. The demand of the source is distributed evenly among the left border nodes of all all inactive sub-grids. There are $(1-2p)n_i$ such border nodes. Hence, each node gets a load of $1/(1-2p) < \lambda_{i+1}$. Note that the source can distribute this load while only creating a congestion of $1/(1-2p)$ among its outgoing edges. Now, we use the induction hypothesis as described in Claim 2.2, and route the flow recursively through the sub-grids of the stripe. With high probability the maximum congestion created in one of the sub-grids is at most $3\lambda_{k+1}$.

It remains to describe the routing algorithm for an inactive sub-grid M_i that receives flow $\lambda_i \cdot n_i$ from the left side and has to push this flow to the right side. The flow is first distributed evenly among the left border nodes of the non-active sub-grids in the left stripe. Then the flow is routed recursively through the sub-grids in S_L. After that the flow is distributed among the left border nodes of inactive grids in S_R, using the edges between the source and nodes in the source column. Then it is again routed recursively, and finally it is distributed evenly among the right border nodes of M_i.

We argue that each of these phases can be performed with congestion only $3\lambda_{k+1}$. The first part works since the sub-grids of M_i formed by the first $n_i/2 - n_i/\log n$ columns can be viewed as a crossbar. We can route any multicommodity flow problem between vertices on the left and on the right with congestion $3 \cdot d_{\max}$ where $d_{\max}$ denotes the maximum flow send or received by a vertex (see Claim A.1 in the appendix for a proof). For our problem each node on the left sends λ_i and each node on the right receives at most $\lambda_{i+1} = \lambda_i \cdot 1/(1-2p)$, since with high probability only a fraction of $2p$ of the sub-grids are active. Hence, $d_{\max} \leq \lambda_{i+1}$.

Therefore, the first and the last step of our routing can be performed with congestion $3\lambda_{i+1} \leq 3\lambda_{k+1}$. The intermediate step of distributing flow between the edges coming out of the left stripe and edges going into the right stripe can be done over the source with maximum congestion λ_{i+1}. For the recursive calls we need congestion $3\lambda_{k+1}$. This proves the claim. $\square$

Claim 2.2 directly implies the lemma as argued in Remark 2.1. $\square$

3 The single sink case

In the following we construct a lower bound of $\Omega(\sqrt{\log n})$ on the competitive ratio of oblivious routing algorithms in the random demand model where all routing requests have a common destination. We use a similar recursive construction as in Section 2. However, the exact parameters of the recursion substantially differ in the following ways.

First of all there is only one target node attached to all nodes in the nth column of the n-by-n grid. All sources that we add to the grid will send their demand to this target node. If we would construct the sources and set up the demand distribution in the same way as in Section 2 we would create a lot of load in the network because all sources are far away from the target node. This load could not be routed with constant congestion.

Therefore, we choose a different construction in which the number of sub-grids generated in a recursive step at level i and the probability p_i that a level i sub-grid becomes active depends on i (in Section 2 each recursive step generated 2ℓ sub-grids and the probability $p = \frac{1}{\log n}$ for a sub-grid to become active was independent of i). We use n_i to denote the side-length of sub-grids on level i, and $\ell_i = \frac{n_i}{n_{i+1}}$ to denote the number of sub-grids generated within a single stripe for a grid on level i.

We choose integers n, k, κ, and ℓ_i for $1 \leq i \leq k$ such that $\kappa \geq 2\log^4 n(\log^n + 1) = \Theta(\log^5 n)$, $\ell_i = 2^i \cdot \kappa$ and $n = \prod_{i=1}^{k}\ell_i$. Then $n = \kappa^k\sqrt{2^{k(k-1)}}$ and hence $k = \Theta(\sqrt{\log n})$.

The sources of our graph G are constructed by the following recursive process. We start with the n-by-n grid. We add a source and attach it to all nodes in column $\lceil \frac{n}{2} \rceil$ with edges of unit capacity. Around this source column we consider two stripes (a left stripe S_L and a right stripe S_R) of width $\frac{n}{\ell_1}$ and partition each stripe into ℓ_1 sub-squares of side-length $\frac{n}{\ell_1}$. Then we add sources to the sub-squares recursively.

Note that by the above construction the side-length of a level i sub-grid is $n_i = \prod_{j=i}^{k}\ell_j$. We use the following demand distribution $\mathcal{D}$. The source on level 1 is always active with demand n. A source on a level $i > 1$ is active with probability $p_i = \log^4 n/\ell_{i-1}$ and in this case it has demand n_i (note that our choice of κ ensures that

$2p_j \leq \log n + 1$ for $1 \leq i \leq k$). We prove the following theorem.

THEOREM 3.1. *The competitive ratio of any oblivious routing algorithm on G with demand distribution $\mathcal{D}$ is $\Omega(\sqrt{\log n})$, with high probability.*

We first show that an oblivious routing algorithm on G with demand distribution $\mathcal{D}$ requires congestion $\Omega(\sqrt{\log n})$ with high probability. Then in Lemma 3.2 we show that an optimum algorithm with high probability only creates constant congestion for demands from distribution $\mathcal{D}$. This gives the theorem.

LEMMA 3.1. *For demand distribution $\mathcal{D}$, any oblivious routing algorithm for graph G creates congestion $\Omega(\sqrt{\log n})$ with high probability.*

Proof. The proof is analogous to the proof of Lemma 2.1. We make an inductive argument in which in each induction step we argue that there exists a sub-grid that has high average load (due to sources on higher levels) and that is active (i.e., it achieves load for the next level, as well). The induction step is as follows.

CONDITION 3.1. *Given an* active *sub-grid M_i on level i such that the average load on essential nodes within M_i due to sources on levels $j < i$ is at least $\alpha_i := \frac{i-1}{4} - \frac{i-1}{\log n}$. Then, with high probability, there*

- *either exists a sub-grid $M_{i+1} \subset M_i$ with average load $\alpha_{i+1} = \frac{i}{4} - \frac{i}{\log n}$ on essential nodes (due to sources on levels $j < i + 1$), or*
- *there is a node inside M_i with load $\Omega(\log n)$.*

Proof. Recall that essential nodes are nodes that are in a sub-grid for every level of the recursion. Let C_i denote the number of essential columns that intersect M_i. These columns all lie in stripe S_L or S_R and both stripes have $C_i/2$ essential columns intersecting them. Hence, a routing path that does not intersect $C_i/2$ essential columns *inside* M_i has to leave M_i before traveling a horizontal distance of n_i/ℓ_i. Since there are only $O(n_i/\ell_i)$ vertical edges for leaving M_i in this region the congestion on vertical edges is at least $\Omega(D \cdot \ell_i/n_i)$ where D denotes the demand that is sent along such routing paths. If $D \geq n_i/2$ this congestion is $\Omega(\ell_i/2) = \Omega(\log n)$. Hence, we assume that the source of M_i (which has demand n_i) routes at least 50% of this demand through a stripe (S_L or S_R) before leaving M_i. This results in an average load of $(\frac{C_i}{2} \frac{n_i}{2})/(C_i n_i) \geq 1/4$ on essential nodes within M_i due to the source of M_i.

Adding the load for sources on levels $j < i$ we get an average load of $\alpha_i + \frac{1}{4}$ for sources on levels $j < i + 1$. We now show that with high probability a sub-grid with average load $\alpha_{i+1} = \alpha_i + \frac{1}{4} - \frac{1}{\log n}$ is active. Let $|V_{\text{ess}}|$ denote

the number of essential nodes within a sub-grid and assume for contradiction that there are less than $a = \ell_i/\log^3 n$ sub-grids that have average load less than α_{i+1}. Further, we can assume that no sub-grid has load $\log n$ (otherwise we are done). Hence, the total load is less than $2\ell_i|V_{\text{ess}}|(\alpha_i + \frac{1}{4} - \frac{1}{\log n}) + a \cdot |V_{\text{ess}}| \cdot \log n < 2\ell_i|V_{\text{ess}}|(\alpha_i + 1/4)$ since $2\ell_i/\log n \gg a \cdot \log n$. This is a contradiction to the assumption that the average load is $\alpha_i + 1/4$.

Hence, there are at least $a = \ell_i/\log^3 n$ sub-grids with average load α_{i+1}. The expected number of these sub-grids that become active on the next level is $ap_{i-1} = \Omega(\log n)$. Therefore, with high probability one of these will be active. This sub-grid can be used as M_{i+1}. $\qquad\square$

The source of M_1 is always active and, hence, M_1 fulfills the requirements of Claim 2.1. Therefore, we can use induction to create a sub-grid M_{k+1} with load $\Omega(k)$, with high probability. $\qquad\square$

LEMMA 3.2. *With high probability an optimum algorithm can route the demands generated by $\mathcal{D}$ with constant congestion.*

Proof. The proof that the optimum algorithm can route the demands generated by distribution $\mathcal{D}$ with constant congestion is more involved than the corresponding proof in Section 2, because we have to forward all demand that is created in the network to the single target node. For the analysis it is therefore important to analyze how much demand is created within a sub-grid M_i.

CONDITION 3.2. *All sources within a sub-grid M_i on level i create a demand of $O(\frac{1}{\sqrt{\log n}} \cdot n_i)$ with high probability.*

Proof. The number of level j sub-grids in a grid of level i is $2^{j-i} \cdot \prod_{s=i}^{j-1} \ell_s$. Such a grid is active with probability $p_j = \frac{\log^4 n}{\ell_{j-1}} = \frac{\log^4 n}{2^{j-1}\kappa} \leq \frac{1}{2^j \log n}$, and in this case creates a demand of $n_j = \prod_{s=j}^{k} \ell_j$. Hence the average load create in M_i is

$$\sum_{j=i}^{k} 2^{j-i} \cdot \prod_{s=i}^{j-1} \ell_s \cdot p_j \cdot n_j \leq \frac{k-i}{\log n} \cdot n_i \leq \frac{1}{\sqrt{\log n}} \cdot n_i \ .$$

Let $\mu = (1/\sqrt{\log n}) \cdot n_i$. If the source of M_i is inactive, the demand created in M_i can be written as a sum of independent random variables such that the contribution of each variable is less than $\mu/\log n$. Therefore, the above bound on the expected load holds with high probability (i.e., the probability that the load exceeds $\frac{\beta}{\sqrt{\log n}} n_i$ is less than $n^{-\Omega(\beta)}$). $\qquad\square$

In the following we assume that every inactive sub-grid creates a demand of at most $c \cdot n_i$ for an appropriate value

$c = O(1/\sqrt{\log n})$, and every active sub-grid only creates demand $(c+1) \cdot n_i$.

We present a routing algorithm that with high probability obtains constant congestion for demand distribution $\mathcal{D}$. We use the same algorithm as in the proof of Lemma 2.2, i.e., an active source routes its demand through sub-grids that are inactive on the next level. Similarly, an inactive sub-grid that has to forward flow from its left to its right border sends this flow through inactive sub-grids.

In contrast to the proof in Section 2 we also have to describe how to deal with flow that is created within sub-grids and leaves these sub-grids (in Section 2 demand generated within a sub-grid was directed to a target in the grid). We use the following induction step.

CONDITION 3.3. *Let* $\lambda_i = (1 + \frac{1}{\log n})^{i-1} + c \cdot \sum_{j=1}^{k-1}(1 + \frac{1}{\log n})^j$. *Suppose that for a sub-grid* M_{i+1} *on level* $i+1$ *with* $1 < i+1 \le k+1$ *we can solve the following routing problems:*

1. *active grid* M_{i+1}:
   An active sub-grid M_{i+1} *can distribute all its internally created flow evenly on its right border nodes with congestion only* $4\lambda_{k+1}$.

2. *inactive grid* M_{i+1}:
   An inactive sub-grid M_{i+1} *can distribute an incoming flow of* λ_{i+1} *and its internally created flow evenly on its right border with congestion only* $4\lambda_{k+1}$.

Then we can solve the corresponding routing problems for M_i *with congestion* $4\lambda_{k+1}$.

Proof. We first prove the result for inactive sub-grids M_i. An inactive sub-grid first distributes the incoming flow evenly among left border nodes of inactive sub-clusters in the left stripe. There are ℓ_i sub-grids within this stripe. The probability for each of them to become active is $p_{i+1} = \frac{\log^4 n}{\ell_i}$. This means that the expected number of active sub-grids is larger than $\log n$, and therefore with high probability at most say $2p_{i+1}\ell_i$ are active (a constant factor more than the expected value). Therefore each inactive sub-cluster in S_L only receives flow $\frac{1}{1-2p_{i+1}}\lambda_i \cdot n_{i+1} < \frac{1}{1-1/(\log n+1)}\lambda_i \cdot n_{i+1} < \lambda_{i+1}n_{i+1}$, with high probability. In order to route the incoming flow from the left border nodes of M_i to the left border nodes of inactive sub-grids in S_L we can use the crossbar property of a rectangular grid as shown in Claim A.1. This step can be done with congestion $3 \cdot d_{\max} \le 3 \cdot \lambda_{i+1} < 3\lambda_{k+1}$.

We can route the flow recursively through the sub-grids in S_L with congestion $4\lambda_{k+1}$ because of the induction hypothesis.

The maximum flow that now resides at a right border node of S_L is not more than $(1 + 1/\log n)\lambda_i + c$. For an inactive sub-grid this holds because no more than $(1 +$

$1/\log n)\lambda_i \cdot n_{i+1}$ flow entered the sub-grid, at most $c \cdot n_{i+1}$ flow was generated inside the grid, and furthermore all flow was distributed evenly among the n_{i+1} border nodes of the sub-grid. For active sub-grids this holds since only $(1 + c)n_{i+1}$ flow is generated inside.

Furthermore, we know that the total flow that resides at the border nodes of S_R is at most $(\lambda_i + c)n_i$. We now distribute this flow evenly among the left border nodes of inactive sub-squares of S_R. Since at most a $1/(\log n + 1)$ fraction of the squares are active (the same argument that we used for S_L above), no border node receives more than $(1 + 1/\log n)(\lambda_i + c)n_i \le \lambda_{i+1}n_i$ of this flow. We can do this distribution by using the edges connecting nodes of the source column to the source of M_i. This step induces congestion at most $(1 + 1/\log n)(\lambda_i + c)$.

Now, we can route recursively through the sub-grids with congestion only $4\lambda_{k+1}$.

The total flow at the right border nodes of S_R is $(\lambda_i + c)n_i$, and no node carries more than $(1+1/\log n)(\lambda_i + c) + c$ of this flow. Finally, we have to distribute this flow evenly among the right border nodes of M_i. By using the crossbar property we can do this with congestion at most $4\lambda_{k+1}$.

The routing algorithm for active sub-grids M_i works similar to the algorithm for inactive sub-grids. In fact it is simpler since it needs only the second part of routing through the sub-grids in S_R. This also can be done with congestion $4\lambda_{k+1}$. □

The sub-grids on level $k + 1$ fulfill the requirements of Claim 3.3. Therefore we can make an induction from $k + 1$ to 1 which gives Lemma 3.2. □

4 Lower bounds for oblivious routing in expander graphs

In previous sections we have proved lower bounds for the congestion of oblivious routing schemes in graphs built upon the grid. In this section we consider oblivious routing in expander graphs and prove lower bounds for both the congestion and throughput. A general definition of throughput for oblivious routing schemes has not appeared in prior work, though throughput-maximization problems with an implicit connection to oblivious routing were considered in [2, 15]. We begin by presenting a natural definition of throughput for an oblivious routing scheme.

DEFINITION 4.1. (FEASIBLE DEMAND MATRIX) *Suppose given a directed or undirected graph* $G = (V, E)$ *with non-negative edge capacities* $\{c(e) : e \in E\}$. *A matrix* $D = (D_{ij})_{i,j \in V}$ *is called a* feasible demand matrix *if there exists a multicommodity flow in* G *which delivers demand* D_{ij} *from* i *to* j, *for each* $i, j \in V$, *and which sends at most* $c(e)$ *units of flow on edge* e, *for each* $e \in E$. *The set of feasible demand matrices for* G *will be denoted by* $\mathfrak{D}(G)$.

If $D, D' \in \mathfrak{D}(G)$, we say that D dominates D', denoted by $D' \prec D$, if $D'_{ij} \le D_{ij}$ for each $i, j \in V$.

DEFINITION 4.2. (THROUGHPUT RATIO) *Suppose given an edge-capacitated graph $G = (V, E)$ as in Definition 4.1, and suppose given a flow of unit value f_{ij} from i to j, for each pair of distinct vertices $i, j \in V$. We refer to this collection of flows $\mathbf{f} = (f_{ij})$ as an oblivious routing scheme, and the throughput ratio of $\mathbf{f}$ is defined to be:*

$$\tau(\mathbf{f}) = \max_{D \in \mathfrak{D}(G)} \min_{D' \in \mathfrak{D}(G)}$$

$$\left\{ \frac{\sum_{i,j \in V} D_{ij}}{\sum_{i,j \in V} D'_{ij}} \;\middle|\; D' \prec D \text{ and } \forall e \in E \; \sum_{i,j \in V} D'_{ij} f_{ij}(e) \le c(e) \right\}.$$

Less formally, $\mathbf{f}$ has throughput ratio τ if for every feasible demand matrix D, it is possible to send at least $1/\tau$ fraction of the throughput of D without exceeding the edge capacities, by throwing out a non-negative amount of demand for each commodity and routing the remaining demand using $\mathbf{f}$. Note that the definition does not specify how the admission control decision (i.e. the choice of $D' \prec D$) is to be done in a distributed manner; in fact it allows for centralized computation of D'. In this sense, it is a strong definition for proving lower bounds and a weak one for proving upper bounds.

THEOREM 4.1. *If $\mathbf{f}$ is an oblivious routing scheme in a graph G whose congestion ratio is R, then $\tau(\mathbf{f}) \le R$.*

Proof. Given a feasible demand matrix D, we have $\sum_{i,j \in V(G)} D_{ij} f_{ij}(e) \le Rc(e)$ because the congestion ratio of $\mathbf{f}$ is R. Hence the demand matrix $D'_{ij} = D_{ij}/R$ satisfies $D' \prec D$, $\sum_{i,j \in V(G)} D'_{ij} f_{ij}(e) \le c(e)$, and $\frac{\sum_{i,j} D_{ij}}{\sum_{i,j} D'_{ij}} \le R$. $\square$

COROLLARY 4.1. *For every undirected graph G with n vertices, there is an oblivious routing scheme whose throughput ratio is $O(\log^2 n \log \log n)$.*

Proof. It is known [10] that for every undirected graph G there exists an oblivious routing scheme $\mathbf{f}$ whose congestion ratio is $O(\log^2 n \log \log n)$. $\square$

THEOREM 4.2. *There exist arbitrarily large graphs G such that the throughput of any oblivious routing scheme for G is $\Omega\left(\frac{\log n}{\log \log n}\right)$, where $n = |V(G)|$.*

Proof. Let G be a graph with the following properties:

1. $|V(G)| = n$;

2. G is $(p+1)$-regular, for some $p = O(1)$;

3. G has girth $g = \Omega(\log n)$;

4. G has edge-expansion $\alpha = \Omega(1)$; i.e. for every set $U \subset V(G)$ with $|U| \le n/2$, the number of edges joining U to $V(G) \setminus U$ is at least $\alpha |U|$; and

5. the automorphism group of G acts transitively on $V(G)$.

An example of a graph family satisfying these properties (with n tending to infinity) is the family of expander graphs constructed by Lubotzky, Phillips, and Sarnak [11]. We will assume that all edges of G have capacity 1.

Let $\ell = \lfloor \log_p \log_p n \rfloor$. For an edge $e = (u, v)$, the subgraph $B_\ell(e) \subset G$ (consisting of all vertices and edges reachable from e by a path of length at most ℓ) is a tree consisting of two complete p-ary trees of depth ℓ, rooted at u and v, with edge e joining the roots of the two p-ary trees. Let $S(u, e)$ denote the set of leaves of the subtree rooted at u, and let $S(v, e)$ denote the set of leaves of the subtree rooted at v; more formally, $S(u, e) = \{w : d(w, u) = d(w, v) - 1 = \ell\}$ and $S(v, e) = \{w : d(w, v) = d(w, u) - 1 = \ell\}$. We have $|S(u, e)| = |S(v, e)| = p^\ell = \Omega(\log n)$. Let T denote the set of ordered pairs $(i, j) \in V(G)^2$ such that there exists $e = (u, v) \in E(G)$ with $i \in S(u, e), j \in S(v, e)$. Note that $|T| = p^{2\ell}(p+1)n$, and that $\forall (i, j) \in T \; d(i, j) = 2\ell + 1$.

Consider the demand matrix

$$(4.1) \qquad D_{ij} = \begin{cases} \frac{n}{2(2\ell+1)|T|} & \text{if } (i, j) \in T \\ 0 & otherwise. \end{cases}$$

We claim that $D_{ij} \in \mathfrak{D}(G)$. To see this, consider the routing scheme $\hat{\mathbf{f}}$ which routes all flow from i to j along the unique path of length $2\ell + 1$, for all $(i, j) \in T$. The demand matrix D and the routing scheme $\hat{\mathbf{f}}$ are automorphism-invariant, i.e. if α is any automorphism of G then $D_{ij} = D_{\alpha(i),\alpha(j)}$ and $\hat{f}_{ij}(e) = \hat{f}_{\alpha(i),\alpha(j)}(\alpha(e))$ for all $i, j \in V(G), e \in E(G)$. It follows that the edge congestions induced by $\hat{f}$ are automorphism-invariant, i.e. for all $e \in E(G), \alpha \in \mathrm{Aut}(G), \sum_{i,j} D_{ij} \hat{f}_{ij}(e) = \sum_{ij} D_{ij} \hat{f}(ij)(\alpha(e))$. The total congestion of all edges in G is

$$\sum_e \sum_{i,j} D_{ij} \hat{f}_{ij}(e) = \sum_{i,j} \sum_e D_{ij} \hat{f}_{ij}(e) = \sum_{i,j} (2\ell + 1) D_{ij}$$

$$= (2\ell + 1)|T| \cdot \frac{n}{2(2\ell + 1)|T|} = \frac{n}{2} \;.$$

The action of $\mathrm{Aut}(G)$ on $E(G)$ partitions the edge set into orbits, each containing at least $n/2$ edges. (Each orbit must contain at least one incident edge of each vertex.) Since the edges in each orbit are equally congested, the congestion on each edge is at most 1. This verifies that $D \in \mathfrak{D}(G)$.

Suppose now that we are given an oblivious routing scheme $\mathbf{f}$. For a pair $(i, j) \in T$ let η_{ij} denote the unique edge $e = (u, v)$ such that $i \in S(u, e), j \in S(v, e)$. Let $T_0 \subseteq T$ denote the set of all pairs $(i, j) \in T$ such that $f_{ij}(\eta_{ij}) < 1/2$, and let $T_1 = T \setminus T_0$. Note that for $(i, j) \in T_0$,

at least half the flow from i to j traverses paths of length at least $g - \ell$, where g is the girth of G. By our assumption on G, $g - \ell \geq C \log(n)$ for some constant C. Hence for all $(i,j) \in T_0$, $\sum_{e \in E(G)} f_{ij}(e) \geq (C/2) \log(n)$.

We consider the cases $|T_0| \geq |T|/2$, $|T_1| \geq |T|/2$ separately. If $|T_0| \geq |T|/2$, then let D be the demand matrix defined in (4.1) and let $\widetilde{D}_{ij} = D_{ij}$ if $(i,j) \in T_0$, 0 otherwise. We have $\widetilde{D} \prec D$ hence $\widetilde{D} \in \mathfrak{D}(G)$. If $D' \prec \widetilde{D}$ and $\sum_{i,j} D'_{ij} f_{ij}(e) \leq 1$ for all $e \in E(G)$, then

$$\frac{1}{2}(p+1)n \geq \sum_{e \in E(G)} \sum_{i,j} D'_{ij} f_{ij}(e) = \sum_{i,j} D'_{ij} \sum_{e \in E(G)} f_{ij}(e)$$
$$\geq \sum_{i,j} D'_{ij} \cdot (C/2) \log(n),$$

which implies

$$\sum_{i,j} D'_{ij} \leq \frac{(p+1)n}{C \log(n)} = O\left(\frac{n}{\log n}\right),$$

while

$$\sum_{i,j} \widetilde{D}_{ij} = |T_0| \frac{n}{2(2\ell+1)|T|} \geq \frac{n}{4(2\ell+1)} = \Omega\left(\frac{n}{\log \log n}\right)$$

Thus $\tau(\mathbf{f}) = \Omega\left(\frac{\log n}{\log \log n}\right)$.

Assume now that $|T_1| \geq |T|/2$. The set T may be partitioned into $2|E(G)| = (p+1)n$ disjoint sets $T(u,v) = S(u,e) \times S(v,e)$, one for each edge $e \in E(G)$ and each ordering of the endpoints of e. The elements of $T(u,v)$ are in one-to-one correspondence with edges of the complete bipartite graph on vertex sets $S(u,e), S(v,e)$, and we may partition $T(u,v)$ into p^ℓ disjoint sets of size p^ℓ, corresponding to a partition of the edge set of that complete bipartite graph into perfect matchings. We have thus partitioned T into $p^\ell(p+1)n$ disjoint sets of size p^ℓ. Since $|T_1| \geq |T|/2$, at least one of the pieces of the partition intersects T_1 in a set T_2 whose cardinality is at least $p^\ell/2$. Let $\widetilde{D}_{ij} = 1$ if $(i,j) \in T_2$, 0 otherwise. It is known, e.g. from the work of Broder, Frieze, and Upfal [5, 7], that there exist edge-disjoint paths joining i to j for each $(i,j) \in T_2$; hence $\widetilde{D} \in \mathfrak{D}(G)$. Let e be the edge which is equal to η_{ij} for all $(i,j) \in T_2$. If $D' \prec \widetilde{D}$ and $\sum_{i,j} D'_{ij} f_{ij}(e) \leq 1$, then

$$\sum_{i,j} D'_{ij} = \sum_{(i,j) \in T_2} D'_{ij} \leq 2 \sum_{(i,j) \in T_2} D'_{ij} f_{ij}(e) \leq 2,$$

while

$$\sum_{i,j} D_{ij} = |T_2| \geq \frac{1}{2} p^\ell = \Omega(\log n).$$

Hence $\tau(\mathbf{f}) = \Omega(\log n)$. $\qquad \square$

Combining this result with Theorem 4.1 we see that the congestion ratio of oblivious routing schemes in the Lubotzky-Phillips-Sarnak expanders is also $\Omega(\log n / \log \log n)$.

5 Open problems

The main open problem is whether we can improve the upper bounds of $O(\log^2 n)$ for oblivious routing with random demands in directed graphs [8] and $O(\log^2 n \log \log n)$ for the adversarial case in undirected graphs [10] to $\tilde{O}(\log n)$.

Another interesting question is whether the graph constructed in Section 2 is indeed has $O(1)$ maximum concurrent-flow minimum sparsest-cut ratio (i.e. a constant max-flow mincut gap). Showing this would be instructive since it shows that the oblivious algorithm of [8] for adversarial demands is still tight for this graph up to an $O((\log \log n)^2)$ factor.

Finally, we consider the throughput lower bound proved in Section 4. While the notion of throughput ratio defined in that section constitutes quite a strong definition from the standpoint of lower bounds, it is possible to formulate a still stronger notion of throughput for oblivious routing schemes. Namely, given an oblivious routing scheme $\mathbf{f}$ we can define a partial ordering $\prec_{\mathbf{f}}$ on demand matrices by specifying that $D' \prec_{\mathbf{f}} D$ if and only if there exists a multicommodity flow f' in G which routes D'_{ij} units of demand from i to j, for each i,j, and which satisfies $f'_{ij}(e) \leq D_{ij} f_{ij}(e)$ and $\sum_{i,j \in V} f'_{ij}(e) \leq c(e)$ and all e. (Note that $D' \prec_{\mathbf{f}} D$ implies that $D' \prec D$.) Now put

$$\hat{\tau}(\mathbf{f}) = \max_{D \in \mathfrak{D}(G)} \min_{D' \prec_{\mathbf{f}} D} \left\{ \frac{\sum_{i,j \in V} D_{ij}}{\sum_{i,j \in V} D'_{ij}} \right\}.$$

Less formally, $\mathbf{f}$ satisfies $\hat{\tau} \leq x$ if for every feasible demand matrix D, it is possible to send at least $1/x$ fraction of the throughput of D without exceeding the edge capacities, by source-routing packets from i to j according to the flow distribution f_{ij} and dropping packets selectively along the way, according to the optimal (centralized) admission control scheme for that flow distribution. It is desirable to understand whether comparably strong lower bounds on throughput ratio can be proven under this stricter definition.

References

[1] N. ALON, B. AWERBUCH, Y. AZAR, N. BUCHBINDER, AND J. S. NAOR, *A general approach to online network optimization problems*, in Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2004, pp. 942–951.

[2] B. AWERBUCH, M. T. HAJIAGHAYI, R. D. KLEINBERG, AND F. T. LEIGHTON, *Online client-server load balancing without global information*, in Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2005, pp. 197–206.

[3] Y. AZAR, E. COHEN, A. FIAT, H. KAPLAN, AND H. RÄCKE, *Optimal oblivious routing in polynomial time*, in Proceedings of the 35th ACM Symposium on Theory of Computing (STOC), 2003, pp. 383–388.

[4] M. BIENKOWSKI, M. KORZENIOWSKI, AND H. RÄCKE, *A practical algorithm for constructing oblivious routing schemes*, in Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2003, pp. 24–33.

[5] A. Z. BRODER, A. M. FRIEZE, AND E. UPFAL, *Existence and construction of edge-disjoint paths on expander graphs*, SIAM Journal on Computing, 23 (1994), pp. 976–989.

[6] C. CHEKURI, S. KHANNA, AND B. SHEPHERD, *The all-or-nothing multicommodity flow problem*, in Proceedings of the 36th ACM Symposium on Theory of Computing (STOC), 2004, pp. 156–165.

[7] A. M. FRIEZE, *Disjoint paths in expander graphs via random walks: A short survey*, in Proceedings of the 2nd International Workshop on Randomization and Approximization Techniques in Computer Science (RANDOM), 1998, pp. 1–14.

[8] M. T. HAJIAGHAYI, J. H. KIM, F. T. LEIGHTON, AND H. RÄCKE, *Oblivious routing in directed graphs with random demands*, in Proceedings of the 37th ACM Symposium on Theory of Computing (STOC), 2005, pp. 193–201.

[9] M. T. HAJIAGHAYI, R. D. KLEINBERG, F. T. LEIGHTON, AND H. RÄCKE, *Oblivious routing on node-capacitated and directed graphs*, in Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2005, pp. 782–790.

[10] C. HARRELSON, K. HILDRUM, AND S. B. RAO, *A polynomial-time tree decomposition to minimize congestion*, in Proceedings of the 15th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), 2003, pp. 34–43.

[11] A. LUBOTZKY, R. PHILLIPS, AND P. SARNAK, *Ramanujan graphs*, Combinatorica, 8 (1988), pp. 261–277.

[12] B. M. MAGGS, F. MEYER AUF DER HEIDE, B. VÖCKING, AND M. WESTERMANN, *Exploiting locality for networks of limited bandwidth*, in Proceedings of the 38th IEEE Symposium on Foundations of Computer Science (FOCS), 1997, pp. 284–293.

[13] B. M. MAGGS, G. L. MILLER, O. PAREKH, R. RAVI, AND S. L. M. WOO, *Finding effective support-tree preconditioners*, in Proceedings of the 17th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), 2005, pp. 176–185.

[14] H. RÄCKE, *Minimizing congestion in general networks*, in Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS), 2002, pp. 43–52.

[15] H. RÄCKE AND A. ROSÉN, *Distributed online call control on general networks*, in Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2005, pp. 791–800.

[16] L. G. VALIANT AND G. J. BREBNER, *Universal schemes for parallel communication*, in Proceedings of the 13th ACM Symposium on Theory of Computing (STOC), 1981, pp. 263–277.

A   Appendix

CONDITION A.1. (CROSSBAR PROPERTY) *Let R denote a rectangular n-by-$\lceil \frac{n}{c} \rceil$ grid and suppose that we are given a multicommodity flow problem between sources in the left column and targets in the right column of the grid. Further, let $d_{\max}$ denote the maximum demand that is sent by any one source and received by any one target. Then the flow problem can be solved with congestion $c \cdot d_{\max}$.*

Proof. Partition the targets into $\lceil \frac{n}{c} \rceil$ classes such that no class contains more than c targets. Assign a column to each class. We route a demand between source $(x_s, 1)$ and target $(x_t, \lceil \frac{n}{c} \rceil)$ by routing it along row x_s, to the column for target $(x_t, \lceil \frac{n}{c} \rceil)$, then we route it along this column to row x_t, and finally we route it along row x_t to the target. In this way a row is only shared between one source and one target, and columns are shared among at most c targets. Therefore the congestion is at most $c \cdot d_{\max}$. $\square$