Oblivious Routing in Directed Graphs with Random Demands

MohammadTaghi Hajiaghayi[*†] Jeong Han Kim [‡] 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. More precisely, Räcke showed that there is an oblivious routing algorithm with polylogarithmic competitive ratio (w.r.t. edge congestion) for any undirected graph. Comparatively little positive results are known about oblivious routing in general *directed* networks. Using a novel approach, we present the first oblivious routing algorithm which is $O(\log^2 n)$-competitive with high probability in directed graphs given that the demands are chosen randomly from a known demand-distribution. On the other hand, we show that no oblivious routing algorithm can be $o(\frac{\log n}{\log \log n})$ competitive even with constant probability in general directed graphs.

Our routing algorithms are not oblivious in the traditional definition, but we add the concept of demand-dependence, i.e., the path chosen for an s-t pair may depend on the demand between s and t. This concept that still preserves that routing decisions are only based on local information proves very powerful in our randomized demand model.

Finally, we show that our approach for designing competitive oblivious routing algorithms is quite general and has applications in other contexts like stochastic scheduling.

*Department of Mathematics and Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 200 Technology Square, Cambridge, MA 02139, U.S.A., Emails: {hajiagha,ftl}@theory.csail.mit.edu

†Part of this work was done while the author was an intern in the Microsoft Research Theory Group.

‡Microsoft Research, One Microsoft Way, Redmond WA 98052, U.S.A., Emails: jehkim@microsoft.com

§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@andrew.cmu.edu

STOC'05, May 22-24, 2005, Baltimore, Maryland, USA.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*; F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms

Algorithms

Keywords

Oblivious Routing, Demand Distributions, Directed Graphs

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.)

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 [15] 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 [13] obtained oblivious routing schemes with polylogarithmic competitive ratio for general undirected graphs. 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]).

In this paper we analyze a model in which the demands between node-pairs are not worst-case 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. We further augment our routing algorithms by the possibility of *demand-dependence*, i.e., the path (or flow) chosen for a node-pair may not only depend on the pair, but also on the demand value for this pair (note that this is still local knowledge in the sense that the node that sets up a routing path should know about the corresponding demand).

We show that for any directed graph if the demands for different node-pairs are independent there is a demand-dependent oblivious routing algorithm that is within $O(\log^2 n)$ of the optimum congestion, with high probability. On the other hand, we show that the concept of demand-dependence is necessary for obtaining efficient algorithms. We show that there are directed networks in which traditional oblivious algorithms perform very badly.

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 [15] 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 [13] 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 Azar et al. [3] who show 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 by Azar et al. does not give the possibility 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 hierarchy used in [13], which has proven to be useful in many applications (see e.g. [1, 6, 11]). 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.

Recently, Hajiaghayi et al. [9] have considered the problem of oblivious routing for directed graphs with a single-sink. They show that we cannot obtain a competitive ratio better than $\Omega(\sqrt{n})$ and we 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. In undirected graphs, they also show that for the single-sink case, we cannot obtain a competitive ratio better than $\Omega(\log n)$ (the best competitive ratio so far is $O(\log^2 n \log\log n)$, the same as the one for the general undirected case). For the cost-measure of throughput instead of congestion, Räcke and Rosen [14] give a distributed online call control algorithm which is closely related to oblivious throughput maximization in undirected graphs. Awerbuch et al. [2] establishes 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 case of directed graphs.

1.2 Our Results

We analyze oblivious routing algorithms in directed graphs when the demands are randomized. We develop a demand-dependent oblivious routing algorithm that is $O(\log^2 n)$-competitive, with high probability. This result forms a strong difference to the standard worst case competitive model for oblivious routing in which for some newtorks no algorithm with competitive ratio $o(\sqrt{n})$ can be obtained.

On the other hand, we show that for some networks no oblivious routing algorithm can be $o(\frac{\log n}{\log\log n})$ competitive even with constant probability.

In addition, we show that the main ingridients of our model, i.e., independence of demands between different node-pairs, and allowing an oblivious routing algorithm to be demand-dependent are necessary in order to obtain a polylogarithmic competitive ratio (versus a polynomial competitive ratio). We show that for general demand distributions (i.e., demands between different node-pairs may be dependent) there are scenarios for which no (demand-dependent) oblivious routing algorithm obtains a good competitive ratio. Further, we show that demand-independent oblivious algorithms perform very badly even for demand-distributions in which demands for different pairs are independent. We also obtain a constant competitive ratio for a class of "symmetric" demand distributions.

All our results work for oblivious routing in node-capacitated undirected graphs for the cost-measure of node-congestion (in which the same polynomial lower bound on the competitive ratio of worst case oblivious routing holds) and also have applications to stochastic scheduling (see recent papers [7, 8] for similar stochastic problems; see Section 4 for the exact statement of the applications).

Finally, it is worth mentioning that our approach of designing competitive oblivious routing algorithms that is based on sampling from the distribution, solving the problem optimally, and then taking the average of the optimums, is quite general and might be useful to design competitive oblivious algorithms for other flow-based problems.

1.3 Formal Definition of the Problem

Our graph terminology is as follows. We represent the network as a graph $G = (V, E)$ (directed or undirected), where V denotes the set of vertices (or nodes) and E denotes the set of edges. We denote the number of vertices by n. We will assume that a capacity function cap is given, assigning a capacity (or bandwidth) to edges in the graph. This models the physical communication potential of the network resources.

In this work, we consider oblivious routing when the demands between node-pairs are chosen randomly from a known demand distribution. An oblivious routing scheme consists of a unit flow from s to t for every node-pair (s, t). The flow for a pair s, t determines how the demand from s to t is routed.

The goal is to minimize the congestion which 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, in which each simple path from s to t for the commodity pair (s,t) gets flow proportional to its share in the routing corresponding to the unit demand. Let $\mathrm{C}_{\mathrm{opt}}(D)$ be the optimum edge-congestion for the demand-matrix D, which can be obtained by solving a linear program (we drop D when it is clear from the context). 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) such that the competitive ratio is small with high (or at least constant) probability, when chosing the demands from the distribution.

We show that for general distributions this goal cannot be achieved. There are graphs for which under general demand distributions any oblivious routing scheme has a large competitive ratio with high probability. Therefore, we usually (unless otherwise stated) refer to demand-distributions in which the demands for different source-target pairs are independent. However, the demand for an individual source-target pair may be chosen according to an arbitrary distribution.

Furthermore, we differentiate between *demand-dependent* and *demand-independent* oblivious routing algorithms. A demand-independent oblivious routing scheme consists of one flow for every source-target pair as described above. A *demand-dependent* routing scheme however defines several $s \to t$ flows for every pair (s,t). Which of these flows is used for routing between s and t may depend on the demand between s and t. This concept of demand-dependence makes an oblivious algorithm more powerful while still preserving the fact that only local knowledge is used for making routing decisions (clearly, the demand to be routed between s and t is known to the node that sets up the routing path). We show that in our model of randomized demands the possibility of demand-dependence is extremely important in order to obtain good routing algorithms.

The following proposition can be easily seen from our model definition.

PROPOSITION 1. *Assume there is an upper bound r such that for each commodity pair the ratio of the maximum demand value to the minimum demand value is at most r. Then we can construct an oblivious routing which is always r-competitive.*

We introduce the following notation used in the paper. We denote the demand from s to t by $D[s,t]$. In several places in the paper, we refer to the demand-matrix D as a demand-vector D indexed by all commodities. In this case $\mathrm{demand}(j)$, denotes the demand for commodity j, and $\mathrm{mincut}(j)$, denotes the minimum capacity of an edge set whose removal disconnects the sink of the commodity j from its source (in a "directed" sense).

2. OBLIVIOUS ALGORITHM

In this section we present a demand-dependent oblivious routing algorithm for general directed graphs with polylogarithmic competitive ratio in our randomized demand model.

THEOREM 2. *For every α there is a demand-dependent oblivious routing algorithm that obtains a competitive ratio of $\alpha \cdot O(\log^2 n)$ with probability at least $1 - \frac{1}{n^\alpha}$.*

A crucial step for designing a routing algorithm with low competitive ratio is to derive a lower bound on the optimum congestion for a given demand pattern. For this we partition the demands into classes in the following way. We say that the demand for a commodity j is in class $\mathcal{C}_k$, $k \in \mathbb{Z}$ if

$$2^k \cdot \mathrm{mincut}(j) \le \mathrm{demand}(j) < 2^{k+1} \cdot \mathrm{mincut}(j) \ .$$

If the demand for commodity j is in $\mathcal{C}_k$ we call j *active* for class $\mathcal{C}_k$. Further, we call a class $\mathcal{C}_k$ *active* if at least one commodity is active for this class. The following observation gives a first lower bound on the optimum congestion.

OBSERVATION 3. *For a demand-vector D let $k_{\max}$ denote the number of the highest active class. Then $\mathrm{C}_{\mathrm{opt}}(D) \ge 2^{k_{\max}}$.*

However, this observation only gives a very crude way for lower bounding the optimum congestion. For our application we need an additional bound that is based on a more sophisticated classification scheme.

Let $\mathrm{C}^k_{\mathrm{opt}}$ and $\mathrm{C}^k_{\mathrm{obl}}$ denote the congestion of the optimum and the oblivious routing algorithm, respectively, for routing commodities in class $\mathcal{C}_k$. Similarly, we define $\mathrm{C}^{\le k}_{\mathrm{opt}}$ and $\mathrm{C}^{\le k}_{\mathrm{obl}}$ as the optimum and oblivious congestion, respectively, for routing commodities in classes with number at most k. Further, let H denote a random variable that describes the highest active class according to the random demands. Define ℓ as

$$\ell := \max \left\{ k \mid \mathbf{E}[\mathrm{C}^{\le k}_{\mathrm{opt}} \mid H = \mathcal{C}_k] > 8\delta \ln n \cdot 2^{k+1} \right\} \qquad (1)$$

if the maximum exists and $\ell := -\infty$ otherwise. (The parameter $\delta > 0$ will be chosen later.) Intuitively, the class $\mathcal{C}_\ell$ is the highest class for which $\mathbf{E}[\mathrm{C}^{\le \ell}_{\mathrm{opt}}]$ is much larger than 2^ℓ. However, for technical reasons we need to condition on the fact that no class higher than $\mathcal{C}_\ell$ is active.

We merge all demands in classes $\mathcal{C}_k$, $k \le \ell$ into one class $\mathcal{B}$ (i.e., a commodity j is in $\mathcal{B}$ if $\mathrm{demand}(j) \le 2^{\ell+1} \cdot \mathrm{mincut}(j)$). If the maximum in Equation 1 does not exist, $\mathcal{B}$ is the empty set. We call $\mathcal{B}$ the *base class* and classes $\mathcal{C}_k$, $k > \ell$ are called *higher order classes*.

The following simple but crucial observation directly follows from this classification scheme.

OBSERVATION 4. *For every higher order class $\mathcal{C}_k$, $\mathbf{E}[\mathrm{C}^k_{\mathrm{opt}} \mid H = \mathcal{C}_k] \le O(\log n \cdot 2^k)$.*

Based on the class-definitions the demand-dependent oblivious routing algorithm constructs a *demand-independent* routing scheme for each higher order class and a demand-dependent routing scheme for the base class $\mathcal{B}$. A commodity j is routed by first determining the class j belongs to, and then routing according to the $s_j \to t_j$ flow in the routing scheme of this class. In Section 2.1 we show that for each higher order class $\mathcal{C}_k$ there is an oblivious routing scheme such that for any α

$$\mathbf{Pr}\left[\mathrm{C}^k_{\mathrm{obl}} \ge \alpha \cdot O(\log n \cdot \mathbf{E}[\mathrm{C}^k_{\mathrm{opt}} \mid \mathrm{H}{=}\mathcal{C}_k]) \;\middle|\; \mathrm{H}{=}\mathcal{C}_k\right] \le \tfrac{1}{n^\alpha} \ . \qquad (2)$$

Furthermore, we show in Section 2.2 that

$$\mathrm{C}_{\mathrm{opt}} \ge \frac{1}{2}\,\mathbf{E}[\mathrm{C}^{\mathcal{B}}_{\mathrm{opt}} \mid H = \mathcal{B}] \qquad (3)$$

holds with probability at least $1 - \frac{1}{n^\delta}$ where δ is the constant used in the definition of the base class $\mathcal{B}$. Finally we present a demand-dependent routing scheme for the class $\mathcal{B}$ for which

$$\mathrm{C}_{\mathrm{obl}}^{\mathcal{B}} \leq \alpha \cdot O(\log^2 n \cdot \mathbf{E}[\mathrm{C}_{\mathrm{opt}}^{\mathcal{B}} \mid H = \mathcal{B}]) \qquad (4)$$

holds with probability at least $1 - \frac{1}{n^\alpha}$. The following proof combines theses results to yield Theorem 2.

Proof of Theorem 2. We choose δ in the definition of $\mathcal{B}$ and the parameter in Equation 4 such that equations 3 and 4 hold together with probability at least $1 - \frac{1}{2n^\alpha}$. In the following we assume that both equations hold.

Now, we distinguish two cases. First suppose that no higher order classes are active. Then equations 3 and 4 already guarantee a competitive ratio of at most $\alpha \cdot O(\log^2 n)$.

Now, suppose that there is an active higher order class. Let $k_{\max}$ denote the number of the highest active class (all probabilities are conditioned on the event that $\mathcal{C}_{k_{\max}}$ is the highest active class). Observation 3 gives that the optimum congestion is at least $2^{k_{\max}}$. It remains to bound the congestion of the oblivious routing algorithm. For classes $\mathcal{C}$,$\mathcal{C}'$ and any threshold t

$$\mathbf{Pr}[\mathrm{C}_{\mathrm{opt}}^{\mathcal{C}} \leq t \mid H = \mathcal{C}'] \geq \mathbf{Pr}[\mathrm{C}_{\mathrm{opt}}^{\mathcal{C}} \leq t \mid H = \mathcal{C}] \enspace .$$

Therefore we can set $\mathcal{C}'$ to $\mathcal{C}_{k_{\max}}$ and use Equation 2 to derive a high-probability upper bound on $\mathrm{C}_{\mathrm{obl}}^{\mathcal{C}}$ that also holds for our case where probabilities are conditioned on the event that $\mathcal{C}_{k_{\max}}$ is the highest active class (instead on the event that $\mathcal{C}$ is the highest class). Then we want to combine these bounds to get a bound for $\mathrm{C}_{\mathrm{obl}}$. However, since Equation 2 only holds with high probability we can apply this bound only to a polynomial number of classes in order to guarantee that with high probability the bound holds for each such class. In fact, it is sufficient if we apply Equation 2 only for the classes $\mathcal{C}_{k_{\max}-2\log n}, \ldots, \mathcal{C}_{k_{\max}}$. For all other classes $\mathcal{C}$ we apply $\mathrm{C}_{\mathrm{obl}}^{k} \leq n^2 \cdot 2^{k+1}$. Let $k' := k_{\max} - 2\log n$. We get with high probability

$$\begin{aligned}
\mathrm{C}_{\mathrm{obl}} &\leq \mathrm{C}_{\mathrm{obl}}^{\mathcal{B}} + \sum_{k=\ell+1}^{k_{\max}} \mathrm{C}_{\mathrm{obl}}^{k} \\
&\leq \mathrm{C}_{\mathrm{obl}}^{\mathcal{B}} + \sum_{k=\ell+1}^{k'-1} \mathrm{C}_{\mathrm{obl}}^{k} + \sum_{k=k'}^{k_{\max}} \mathrm{C}_{\mathrm{obl}}^{k} \\
&\leq O(\log^2 n \cdot \mathbf{E}[\mathrm{C}_{\mathrm{opt}}^{\mathcal{B}} \mid H = \mathcal{B}]) \\
&\quad + \sum_{k=k'}^{k_{\max}} O(\log n \cdot \mathbf{E}[\mathrm{C}_{\mathrm{opt}}^{k} \mid H = \mathcal{C}_k]) \\
&\quad + \sum_{k=\ell+1}^{k'-1} O(n^2 \cdot \mathbf{E}[\mathrm{C}_{\mathrm{opt}}^{k} \mid H = \mathcal{C}_k]) \\
&\leq O(\log^2 n \cdot \mathrm{C}_{\mathrm{opt}}) + \sum_{k=k'}^{k_{\max}} O(\log^2 n \cdot 2^k) \\
&\quad + \sum_{k=\ell+1}^{k'-1} O(n^2 \cdot 2^k) \\
&\leq O(\log^2 n \cdot \mathrm{C}_{\mathrm{opt}} + \log^2 n \cdot 2^{k_{\max}}) \\
&\leq O(\log^2 n \cdot \mathrm{C}_{\mathrm{opt}}) \enspace .
\end{aligned}$$

■

```
ComputeRoutingScheme (D,C)
  for i = 1 to n^3 do
    draw a demand-vector D_i
    according to distribution D_C
    compute D'_i from D_i by setting all
    demands not active for class C to zero.
    compute an optimal
    multicommodity flow opt-flow(D_i)
  end
  sum-flow := Σ_i opt-flow(D_i)
  normalize sum-flow to get a multicommodity flow in
  which the demand from each source to its sink is 1.
```

Figure 1: The algorithm for computing the oblivious routing scheme for a higher order class $\mathcal{C}$.

2.1 Routing algorithms for demand-classes

In this section we proof the following lemma that is crucial for the performance analysis of our demand-dependent oblivious routing algorithm as described in the previous section.

LEMMA 5. *For a high order class $\mathcal{C}$ there is an oblivious routing scheme such that*

$$\mathbf{Pr}\Big[\mathrm{C}_{\mathrm{obl}}^{\mathcal{C}} \geq \alpha \cdot O(\log n \cdot \mathbf{E}[\mathrm{C}_{\mathrm{opt}}^{\mathcal{C}} \mid H = \mathcal{C}]) \;\Big|\; H = \mathcal{C}\Big] \leq \frac{1}{n^\alpha} \enspace .$$

Proof. We start by presenting the algorithm ComputeRoutingScheme that computes the demand-independent oblivious routing scheme for a class $\mathcal{C}$.

Let $\mathcal{D}_{\mathcal{C}}$ denote the demand distribution conditioned on the event that $\mathcal{C}$ is the highest active class. We can sample a demand-vector from $\mathcal{D}_{\mathcal{C}}$ because we know the distribution $\mathcal{D}$. Let $\mathrm{maxdem}(\mathcal{C})$ denote the maximum (relative) demand of the class $\mathcal{C}$ (i.e. if $\mathcal{C}$ is the base class $\mathrm{maxdem}(\mathcal{C}) = 2^{\ell+1}$ and if $\mathcal{C} = \mathcal{C}_k$ $\mathrm{maxdem}(\mathcal{C}) = 2^{k+1}$).

The algorithm ComputeRoutingScheme works as follows (also see Figure 1). It first fixes a routing scheme for commodities j that have a very low expected demand namely $\mathbf{E}[\mathrm{demand}(j) \mid H = \mathcal{C}] \leq \frac{1}{n^2}\,\mathrm{maxdem}(\mathcal{C}) \cdot \mathrm{mincut}(j)$. These commodities are routed according to a maximum $s_j \rightarrow t_j$ flow (scaled by $\frac{1}{\mathrm{mincut}(j)}$).

For the other commodities the algorithm first samples demand-vectors D_i according to distribution $\mathcal{D}_{\mathcal{C}}$. Then for every D_i it deletes the demand for commodities that are not in class $\mathcal{C}$ and the demands for the low-expectation commodities. For every resulting vector D'_i an optimal multicommodity flow opt-flow(D'_i) is computed that routes the remaining demands optimally. Finally it computes for each commodity j the *average flow* that is used in solutions opt-flow(D'_i). This is done by first adding all optimum flows up. Let sum-flow denote the result of this operation, and let sumdem(j) denote the demand that is routed for commodity j in sum-flow. In the normalization step (for computing the average flow) the flow for commodity j in sum-flow is multiplied by $\frac{1}{\mathrm{sumdem}(j)}$ to obtain a unit flow.

The following lemma shows some important properties of sum-flow that are crucial for proving Lemma 5.

LEMMA 6. *With high probability* sum-flow *fulfills the following properties.*

1. *For any high-expectation commodity j,* $\text{sumdem}(j) = \Omega(\mathbf{E}[\text{demand}(j) \mid H = \mathcal{C}] \cdot n^3)$.
2. *The load on any edge is at most* $O(n^3 \cdot \mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}])$.
3. *The load on any edge for routing a specific demand j is only $O(p_j \cdot n^3 \cdot \mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}])$, where p_j denotes the probability that commodity j is active for class $\mathcal{C}$.*

Proof.

1. The random variable $\text{sumdem}(j)$ is a sum of independent random variables. The expectation is $\mathbf{E}[\text{demand}(j) | H = \mathcal{C}] \cdot n^3 \geq n \cdot \text{maxdem}(\mathcal{C}) \cdot \text{mincut}(j)$ which is much larger than the maximum contribution of each variable (which is $\text{maxdem}(\mathcal{C}) \cdot \text{mincut}(j)$). Applying a Chernoff Bound gives that the probability that $\text{sumdem}(j)$ deviates too much from its expectation is small.
2. Fix an edge e. The expectation of the load that is added on e in each round is at most $\mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}]$. Hence, the expected load for edge e is at most $O(n^3 \cdot \mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}])$. Each trial can create a load of at most $n^2 \cdot \text{maxdem}(\mathcal{C})$ because the demand for all commodities that are active for $\mathcal{C}$ can be routed with this congestion. Again, we can write the load of e as a sum of independent random variables where the expectation is much larger than the maximum contribution of a single variable. Using a Chernoff Bound gives that the probability that the second property is not fulfilled is very low.
3. Let p_j denote the probability that commodity j becomes active for $\mathcal{C}$. Further, let $\text{load}_i(e)$ denote the load on edge e for some fixed commodity j in the *i-th* round of the algorithm.

$$\begin{aligned}\mathbf{E}[\text{load}_i(e)] &= \mathbf{E}[\text{load}_j(e) \mid j \text{ is active}] \cdot p_j \\ &\leq (\mathbf{E}[C_{\text{opt}} \mid H = \mathcal{C}] \\ &\quad + \text{maxdem}(\mathcal{C})) \cdot p_j \\ &\leq 3p_j \cdot \mathbf{E}[C^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}] \ ,\end{aligned}$$

   where the second inequality holds since j can be routed individually with congestion at most $\text{maxdem}(\mathcal{C})$. The last inequality holds since for a higher order class $\mathcal{C}_k$, $\mathbf{E}[\mathrm{C}^k_{\text{opt}} | H = \mathcal{C}] \geq 2^k$, and $\text{maxdem}(\mathcal{C}_k) = 2^{k+1}$. The above equation means that the expected load on e for commodity j is

$$\begin{aligned}&\mathbf{E}\left[\textstyle\sum_i \text{load}_i(e)\right] \\ &\leq 3n^3 p_j \cdot \mathbf{E}\left[C^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}\right] \ .\end{aligned}$$

   All commodities active for class $\mathcal{C}$ can be routed with congestion at most $\text{maxdem} \cdot n^2$. Therefore, each random variable $\text{load}_i(e) \leq \text{maxdem}(\mathcal{C}) \cdot n^2 \leq 2\mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}}] \cdot n^2$. We can apply a Chernoff Bound which gives that with high probability $\mathbf{E}[\sum_i \text{load}_i(e)] = O(p_j \cdot n^3 \cdot \mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}]$.

■

Claim 7. *If* sum-flow *fulfills the properties of Lemma 6 the oblivious routing scheme achieves congestion $O(\log n \cdot \mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} | H = \mathcal{C}])$ with high probability.*

Proof. Fix an edge e. Let $\text{load}(e)$ denote a random variable that describes the load on edge e using the oblivious routing scheme. Further, let $\text{sum-flow}(e, j)$ denote the load on edge e for commodity j in sum-flow, and let $\text{demand}(j)$ denote a random variable that is 0 if the demand for j is not in $\mathcal{C}$ and equals this demand, otherwise. Let X_j denote the contribution from commodity j to $\text{load}(e)$. $X_j = (\frac{\text{sum-flow}(e,j)}{\text{sumdem}(j)}) \cdot \text{demand}(j)$. Hence,

$$\begin{aligned}\mathbf{E}[\text{load}(e)] &= \mathbf{E}\left[\sum\nolimits_j X_j\right] \\ &= \mathbf{E}\left[\sum\nolimits_j \frac{\text{sum-flow}(e,j)}{\text{sumdem}(j)} \text{demand}(j)\right] \\ &= \sum_j \left(\frac{\text{sum-flow}(e,j)}{\text{sumdem}(j)}\right) \cdot \mathbf{E}[\text{demand}(j)] \\ &\leq O\left(\sum\nolimits_j \text{sum-flow}(e,j)/n^3\right) \\ &= O(\mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}]) \ .\end{aligned}$$

Now, we bound the maximum value of a random variable X_j.

$$\begin{aligned}X_j &= \left(\tfrac{\text{sum-flow}(e,j)}{\text{sumdem}(j)}\right) \cdot \text{demand}(j) \\ &\leq O\left(\tfrac{p_j \cdot n^3 \cdot \mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} | H = \mathcal{C}]}{\mathbf{E}[\text{demand}(j) | H = \mathcal{C}] \cdot n^3}\right) \cdot \text{demand}(j) \\ &\leq O\left(\tfrac{p_j \cdot \mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} | H = \mathcal{C}]}{p_j \cdot \text{maxdem}(\mathcal{C}) \cdot \text{mincut}(j)}\right) \cdot \text{demand}(j) \\ &\leq O(\mathbf{E}[\mathrm{C}^{\mathcal{C}}_{\text{opt}} \mid H = \mathcal{C}])\end{aligned}$$

■

Lemma 6 and Claim 7 give the lemma. ■

The following lemma shows the existence of a good demand-dependent routing scheme for the base class $\mathcal{B}$.

Lemma 8. *There is a demand-dependent oblivious routing scheme for $\mathcal{B}$ that obtains congestion $\mathrm{C}^{\mathcal{B}}_{\text{obl}} = O(\log^2 n \cdot \mathbf{E}[\mathrm{C}^{\mathcal{B}}_{\text{opt}} \mid H = \mathcal{B}])$ with high probability.*

Proof. The routing scheme works as follows. All demands for a commodity j that are smaller than $2^\ell/n^2$ are routed according to a maximum $s_j \to t_j$ flow. Since, there are at most n^2 commodities, these demands can cause a congestion of at most $2^\ell \leq O(\log^2 n \cdot \mathbf{E}[\mathrm{C}^{\mathcal{B}}_{\text{opt}} \mid H = \mathcal{B}])$.

We partition the remaining demands into $2 \log n$ classes $\mathcal{B}_i$, $1 \leq i \leq 2 \log n$ such that commodity j is in class $\mathcal{B}_i$ if $2^{\ell - i} \leq \text{demand}(j) \leq 2^{\ell - i + 1}$. For each such class we can compute the oblivious routing scheme as we did for classes $\mathcal{C}_k$ in the proof of Lemma 5 because demands for a commodity in a class only vary by a constant factor.

Then $\mathrm{C}^{\mathcal{B}_i}_{\text{opt}}$ is less than $O(\log n \cdot \mathbf{E}[\mathrm{C}^{\mathcal{B}}_{\text{opt}} \mid H = \mathcal{B}])$ for each class with high probability. Therefore with high probability $\mathrm{C}^{\mathcal{B}}_{\text{opt}} \leq \sum_i \mathrm{C}^{\mathcal{B}_i}_{\text{opt}} \leq O(\log^2 n \cdot \mathbf{E}[\mathrm{C}^{\mathcal{B}}_{\text{opt}} \mid H = \mathcal{B}]$. ■

2.2 The concentration result

In this section, we prove a high-probability lower bound for the optimum congestion in terms of the expected congestion for routing commodities in the base class $\mathcal{B}$.

Lemma 9. *With probability at least $1 - \frac{1}{n^\delta}$,*

$$\mathrm{C}_{\text{opt}} \geq \frac{1}{2}\mathbf{E}[\mathrm{C}^{\mathcal{B}}_{\text{opt}} \mid H = \mathcal{B}] \ .$$

Proof. We show that $\mathbf{Pr}\left[\mathrm{C}^{\mathcal{B}}_{\text{opt}} \geq \frac{1}{2}\mathbf{E}[\mathrm{C}^{\mathcal{B}}_{\text{opt}} \mid H = \mathcal{B}] \;\middle|\; H = \mathcal{B}\right]$ is large. The lemma then follows from the fact that

$$\begin{aligned}\mathbf{Pr}\left[\mathrm{C}^{\mathcal{B}}_{\text{opt}} \geq t \mid H = \mathcal{B}\right] &\leq \mathbf{Pr}[\mathrm{C}_{\text{opt}} \geq t \mid H = \mathcal{B}] \\ &\leq \mathbf{Pr}[\mathrm{C}_{\text{opt}} \geq t] \ .\end{aligned}$$

In the following all events and expectations are conditioned on the event that $\mathcal{B}$ is the highest active class. We can derive the lemma as a corollary from a very powerful tail estimate by Boucheron et al. [5]. We introduce the following definition that is slightly modified from an analogous definition in their paper.

DEFINITION 10. *Let $X_1, \ldots, X_n$ denote independent random variables where each $X_i \in [0, W_i]$. A function $f : [0, W_1] \times \ldots \times [0, W_n] \to \mathbb{R}^+$ is self-bounded if*

$$1 \geq f(X_1, \ldots, X_n) - f(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots, X_n) \geq 0$$

and

$$(n-1) \cdot f(X_1, \ldots, X_n) \leq \sum_i f(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots, X_n)\,.$$

Boucheron et al. show that a Chernoff-like concentration bound holds for self-bounded functions.

THEOREM 11 (BOUCHERON, LUGOSI, MASSART [5]). *Let $X_1, \ldots, X_n$ denote independent random variables with $X_i \in [0, \ldots, W_i]$, let f denote a self-bounded function, and let μ denote the expectation of $f(X_1, \ldots, X_n)$. Then*

$$\mathbf{Pr}[f(X_1, \ldots, X_n) \leq (1-\epsilon)\mu] \leq e^{-\epsilon^2\mu/2}\ .$$

For our application the independent variables X_i are the entries in the demand-vector. Since the demand for any commodity that is active for $\mathcal{B}$ can be routed with congestion $2^{\ell+1}$, the function $\frac{1}{2^{\ell+1}}\,\mathrm{C}^{\mathcal{B}}_{\mathrm{opt}}$ fulfills the first property of a self-bounded function. The second property holds because the congestion function $\mathrm{C}^{\mathcal{B}}_{\mathrm{opt}}$ is sub-additive (i.e., $\mathrm{C}^{\mathcal{B}}_{\mathrm{opt}}(D_1 + D_2) \leq \mathrm{C}^{\mathcal{B}}_{\mathrm{opt}}(D_1) + \mathrm{C}^{\mathcal{B}}_{\mathrm{opt}}(D_2)$).

For the function $\frac{1}{2^{\ell+1}}\,\mathrm{C}^{\mathcal{B}}_{\mathrm{opt}}$ the expectation μ is at least $8\delta \ln n$. Hence, applying the above theorem for $\epsilon = 1/2$ gives $\mathbf{Pr}\left[\mathrm{C}^{\mathcal{B}}_{\mathrm{opt}} \leq 1/2 \cdot \mathbf{E}[\mathrm{C}^{\mathcal{B}}_{\mathrm{opt}}]\right] \leq \frac{1}{n^\delta}$. ■

3. LOWER BOUNDS

In this section, we present several lower bounds for oblivious routing with randomized demands.

THEOREM 12. *There exists a graph for which every demand independent oblivious routing algorithm has competitive ratio $\Omega(\frac{\sqrt{n}}{\log n})$ with high probability.*

Proof. Consider a graph G consisting of k sources $s_1, \ldots, s_k$, k sinks $t_1, \ldots, t_k$, and two additional nodes u and v. Each source s_i is connected to t_i via a directed edge of capacity 1. Furthermore, there exist infinite capacity edges (s_i, u) and (v, t_i) for each i, and u is connected to v via an edge with capacity $\sqrt{k}$. Note that each source-target pair s_i, t_i is connected via two paths; one with capacity 1 which is exclusive to the pair s_i, t_i; and one with capacity $\sqrt{k}$ via edge (u, v) that may be shared by all pairs.

Consider the demand distribution that assigns a demand of $\sqrt{k}$ to commodity i with probability $p := \frac{2(\alpha+1)\ln n}{k}$, and a demand of 1 with probability $1 - p$. A Chernoff Bound shows that with high probability ($\geq 1 - \frac{1}{k^{\alpha+1}}$) the number of high-demand pairs (i.e., pairs with demand $\sqrt{k}$) is only $\alpha \cdot O(\log n)$. In this case, by routing the high-demand pairs over the high capacity edge, and the remaining commodities directly along the corresponding (s_i, t_i) edge, the optimum congestion is $\alpha \cdot O(\log k)$.

However, an oblivious routing algorithm has to fix the routing paths without knowing the high-demand pairs. We show that every oblivious routing algorithm creates congestion $\Omega(\sqrt{k})$, w.h.p.

Fix an oblivious routing algorithm. Let h denote the number of source-target pairs for which the algorithm routes at least $1/2$ of the demand along the high capacity edge (u, v). Clearly, if $h > k/2$ the load on edge (u, v) will be at least $\frac{1}{2}h/\sqrt{k} \geq \sqrt{k}/4$.

On the other hand, if $h < k/2$ there exists at least $k/2$ pairs that route at least $1/2$ of their demand via edge (s_i, t_i). The probability that none of these pairs gets a high demand is only $(1 - \frac{2(\alpha+1)\ln n}{k})^{k/2} \leq \frac{1}{k^{\alpha+1}}$. This means that with high probability one of them will get a high demand which leads to a congestion of $\Omega(\sqrt{k})$.

Combining the result for the optimum and the oblivious algorithm gives that with high probability the competitive ratio is $\Omega(\sqrt{k}/\log k)$. ■

The above theorem shows that demand-dependence is essential for deriving oblivious routing algorithms with low competitive ratio in the randomized demand model. Note that the above proof also rules out an oblivious routing scheme that is good with respect to any demand-distribution.

Further, it is worth mentioning that the concept of demand dependence does not help for the standard worst case scenario of oblivious routing since the counter example [3] that proves a lower bound of $\sqrt{n}$ on the competitive ratio in directed graphs works with uniform demands.

Next, we show that in order to obtain a polylogarithmic competitive ratio the assumption that demands for different commodities are indpendent is necessary.

THEOREM 13. *In the case of general demand-distributions, i.e. when demands for different pairs are not necessarily independent, there exist graphs for which even a demand-dependent oblivious routing scheme has competitive ratio $\Omega(n^{1/3}/\log n)$, with high probability.*

Proof. We choose the graph G from the proof of Theorem 12 and add for each source node s_i a set of $\sqrt{k}$ nodes $s_i^1, \ldots, s_i^{\sqrt{k}}$ each with a direct link to s_i. We call the nodes $s_i^1, \ldots, s_i^{\sqrt{k}}$ the *sources corresponding to s_i*. Note that the new graph contains $n = O(k \cdot \sqrt{k})$ vertices.

We choose the following demand-distribution. First for every nodes s_i we choose a random node s_i^j among its sources and assign a demand of 1 from this source to target t_i. Then we choose $\Theta(\log k)$ random nodes from $\{s_1, \ldots, s_k\}$ and create for all their sources a demand of 1 to the corresponding target node (i.e., sources corresponding to s_j are assigned a demand to target t_j). We call the chosen nodes the *high demand nodes*. Note that the demand assigned to a commodity is either 0 or 1, i.e., there is no difference between demand dependent and demand independent oblivious routing.

Clearly, by sending the traffic produced by sources of high demand nodes over edge (u, v) and the remaining traffic via the corresponding (s_i, t_i)-link an optimal strategy can always route the above demand pattern with congestion $O(\log k)$.

Now, we prove analogously to Theorem 12 that any oblivious algorithm creates a large congestion with high probability. Let h denote the number of nodes s_i for which at least 50% of the corresponding sources $s_i^1, \ldots, s_i^{\sqrt{k}}$ route more than a constant fraction of their demand along the high-capacity link. Then, if $h > k/2$ the total demand along (u, v) due to the oblivious algorithm will be $\Theta(k)$, w.h.p., which results in a congestion of $\Omega(\sqrt{k})$.

Otherwise, there exist more than $k/2$ nodes s_i for which many sources send a large fraction of their traffic along edge (s_i, t_i). If one of these nodes s_i becomes a high demand node the resulting congestion will be $\Omega(\sqrt{k})$. This happens with high probability. ■

Theorem 14. *There is a graph G for which any demand dependent oblivious routing algorithm obtains competitive ratio $\Omega(\log n/\log\log n)$, with high probability.*

Proof. Consider the following graph G with three levels. The first level contains $\binom{k}{2}$ nodes denoted by a_{ij} for $1 \le i < j \le k$. The second level contains k nodes denoted by b_i for $1 \le i \le k$ and the third level contains a super-sink s. Each node a_{ij} is connected via two directed edges to nodes b_i and b_j. Further, each node b_i is connected to the sink. All edges have unit capacity.

Consider the demand distribtuion that for a node a_{ij} creates a demand of 1 with probability $\frac{1}{k}$ and a demand of 0 with probability $1 - 1/k$ (all demands are routed to s). Fix an oblivious routing algorithm A. We first show that with high probability A creates congestion $\Omega(\frac{\log n}{\log\log n})$. We mark for each node a_{ij} the outgoing edge that - according to A - routes most of the $a_{ij} \to s$ flow as *heavy* (if both outgoing edges have half the flow we mark one of them arbitrarily). We call a node b_i on the second level *heavy* if it is incident to at least $k/4$ heavy edges. Since each node b_i can only be incident to at most $k - 1$ heavy edges, there are at least $k/4$ heavy nodes (otherwise there would only be $\frac{3}{4}k \cdot \frac{k}{4} + \frac{k}{4} \cdot (k-1) < \binom{k}{2}$ heavy edges).

Now, each heavy node b_d makes $k/4$ trials (for each incident heavy edge) with probability $1/k$. If a trial is a hit (i.e., the corresponding node a_{ij} has demand 1) a load of at least $1/2$ is created on the edge connecting b_d to s. There are $k/4$ heavy nodes. With high probability one of them will have at least $\Omega(\log k/\log\log k)$ hits (The analysis for this is analogous to the analysis for balls-into-bins scenarios. See [12]). $\Omega(\log k/\log\log k)$ hits result in a congestion of $\Omega(\log k/\log\log k)$.

Next, we show that with high probability the optimum congestion is constant. For an instance of the problem in which a set X of a_{ij} sources, $1 \le i < j \le k$, are active (i.e., their demands are one), from Hall's theorem, we observe that the optimum congestion is equal to maximum ratio $\frac{|X \cap \{a_{ij} | b_i \in L \wedge b_j \in L\}|}{|L|}$ over all $L \subseteq \{b_1, b_2, \ldots, b_k\}$. For a set $L \subset \{b_1, b_2, \ldots, b_k\}$, we call set $\{a_{ij} | b_i \in L \wedge b_j \in L\}$ *incoming sources of* L and we let $\ell = |L|$. We now finish the proof by showing that with high probability, for each set $L \subseteq \{b_1, b_2, \ldots, b_k\}$, the number of active incoming sources is at most $c\ell$ for some constant c. By the Chernoff bound, we know that

$$\mathbf{Pr}[\text{number of active incoming sources of } L > (1+\delta)\mu]$$

$$\le \left[\frac{e^\delta}{(1+\delta)^{1+\delta}}\right]^\mu \le \left[\frac{e}{\delta}\right]^{\delta\mu}$$

where $\mu = \binom{\ell}{2}/k \ge \frac{\ell^2}{4k}$. By setting $\delta = \frac{12ek}{\ell}$ we get that this probability is less than $[\frac{\ell}{ek}]^{8\ell}$. Since the number of subsets of size $\ell > 2$ of $\{b_1, \ldots, b_k\}$ is $\binom{k}{\ell} \le (\frac{ke}{\ell})^\ell$, the probability that a subset L of size $\ell \ge 2$ has load greater than $(\delta+1)\mu = O(\ell)$ is at most $(\frac{ke}{\ell})^\ell \cdot [\frac{\ell}{ek}]^{8\ell} \le [\frac{\ell}{ek}]^{7\ell} \le [\frac{\ell}{2k}]^{7\ell}$.

Now, assume that $\ell \ge \sqrt{k}$. Then the probability is smaller than $[\frac{\ell}{2k}]^{7\ell} \le [\frac{1}{2}]^\ell \le [\frac{1}{2}]^{\sqrt{k}} = o(1/k^3)$. Otherwise, in the case that $\ell \le \sqrt{k}$ we get that the probability is less than $[\frac{\ell}{2k}]^{7\ell} \le [\frac{\ell}{k}]^7 \le o(1/k^3)$. By applying a union bound for all sizes ℓ we get that with probability at least $1 - o(1/k^2) = 1 - o(1/n)$ the optimum congestion is at most $O(1)$.

Hence, the competitive ratio is at least $\Omega(\frac{\log n}{\log\log n})$ with high probability.

■

4. FURTHER DISCUSSION

In this section, we present further results on oblivious routing with known demand distributions and some open problems.

Symmetric distributions. In Section 2 we showed that in our randomized model an oblivious routing algorithm can obtain a polylogarithmic competitive ratio for general demand distributions, as long as demands for different commodities are independent. Furthermore, Theorem 14 shows that this result in general cannot be improved by too much, as there are networks and demand distributions such that any oblivious routing algorithm (with high probability) produces a congestion that is a factor of $\Omega(\log n/\log\log n)$ larger than the optimum congestion. However, in this section we show that for special distributions this bound can be improved.

We call a probability distribution *symmetric around its expectaion* (or just *symmetric*) if $\rho(\mu - a) = \rho(\mu + a)$ holds for any real a, where μ denotes the expectation and $\rho(\cdot)$ denotes the probability density function of the distribution. Many practical distribution like e.g. Uniform or Gaussian distributions are symmetric. The following theorem shows that for symmetric demand-distributions it is possible to obtain a constant comepetitive ratio with constant probability.

Theorem 15. *For any graph G and for symmetric demand distributions there is an oblivious routing algorithm that obtains competitive ratio 2 with probability at least $\frac{1}{2}$.*

Proof. The routing paths for the oblivious routing algorithm are obtained by solving a concurrent multicommodity flow problem in which the demand from node i to j equals the expected demand according to the demand distribution for commodity i-j. Let $\mathrm{C_{opt}}(D_{\exp})$ denote the congestion of this multicommodity flow solution, and let μ_{ij} denote the expected demand for commodity i-j.

Now, let D denote a demand matrix, and let D' denote a matrix that is obtained from D by setting the ij-entry to $D'_{ij} := 2\mu_{ij} - D_{ij}$. The optimum congestion for routing the demand matrix $D'' := D + D'$ is $\mathrm{C_{opt}}(D'') = \mathrm{C_{opt}}(D_{\exp})$. Since $\mathrm{C_{opt}}(D) + \mathrm{C_{opt}}(D') \ge \mathrm{C_{opt}}(D'') = 2\,\mathrm{C_{opt}}(D_{\exp})$, we know that for at least one of D and D', say D, $\mathrm{C_{opt}}(D) \ge \mathrm{C_{opt}}(D_{\exp})$.

On the other hand, we know that the congestion that is produced for D by the oblivious routing algorithm is at most $2\,\mathrm{C_{opt}}(D_{\exp})$. It means for at least one of the demand matrices D and D', the oblivious routing algorithm is 2-competitive.

Furthermore, since the demand distributions are symmetric the probability for matrix D and D' is equal. Since we can partition the sample points of the joint distribution into such pairs and for at least one sample point of each pair, we are 2-competitive, we obtain that, in total, we are 2-competitive with probability at least 1/2. ∎

Introducing other classes of distributions for which we can be constant competitive with constant probability (or with high probability) is an interesting open question.

Source-oblivious routing. Our oblivious routing algorithm in Section 2, each intermediate node should know the source and the sink of each incoming packet in order to forward it through an appropriate edge. It is interesting to know whether we can obtain more compact routing tables by designing a *source-oblivious routing scheme*, i.e., a routing scheme in which each intermediate node decides the destination of each packet only based on its sink (and not its source). Below, we show that essentially we cannot be better than polynomial competitive in this setting

THEOREM 16. *There is a graph G for which the competitive ratio of any source-oblivious routing algorithm is $\Omega(\frac{\sqrt{n}}{\log n})$ with high probability.*

Proof. First, we consider the graph G constructed in the proof of Theorem 14. Now we add to G vertices c_i, $1 \le i \le k$ and a'_{ij}, $1 \le i < j \le k$, and edges (a'_{ij}, a_{ij}), $1 \le i < j \le k$, (c_i, a_{ij}), $1 \le i < j \le k$ and (c_j, a_{ij}), $1 \le i < j \le k$. Note that again $n = \Theta(k^2)$. Edge-capacities of (a'_{ij}, a_{ij}), $1 \le i < j \le k$, are $k-1$ and all other edge capacities are one. Assume that with probability $\frac{4\log k}{k}$, we have a demand $k-1$ and with probability $1 - \frac{4\log k}{k}$, we have a demand zero for commodity pair $c_i \to s$, $1 \le i \le k$. Consider an oblivious routing algorithm O.

First we show that with high probability algorithm O has congestion at least $\Omega(k)$ on G. We define *heavy edges* (a_{ij}, b_d), $d \in \{i, j\}$, and *heavy nodes (vertices)* b_i, $1 \le i \le k$, exactly the same way that we defined in the proof of Theorem 14. Again we can observe that we have $\Theta(k)$ heavy vertices. Now using the Chernoff bound, we observe that with high probability at least one c_i, $1 \le i \le k$, whose corresponding vertex b_i is a heavy vertex gets demand $k-1$. In this case, since our routing is source-oblivious, edge (b_i, s) gets congestion $\Omega(k)$.

Next, we show that with high probability $C_{\rm opt}$ is in $O(\log k)$ and thus we are done. We consider the following algorithm. When a source c_h, $1 \le h \le k$, gets a demand $k-1$ send one unit of flow to each a'_{ij}, where $h \in \{i, j\}$. Each node a'_{ij} forward the whole flow to a_{ij}. Each node a_{ij} sends each unit of flow originated from c_i to b_j and each unit of flow originated from c_j to b_i. Each b_i, $1 \le i \le k$, sends the incoming flow directly to sink s. Since, with high probability, at most $O(\log k)$ sources c_h, $1 \le h \le k$, get non-zero demands, it is easy to observe that using the algorithm described here, the total congestion would be $O(\log k)$ with high probability. ∎

Note that in the above proof, even the knowledge from which incoming edge the flow arrives cannot help to improve the polynomial lower bound on the competitiveness (since for each node a_{ij} all flow comes from node a'_{ij}). It is worth mentioning that designing source-oblivious routing with polylogarithmic competitive ratio in undirected graphs parallel to the work of Räche [13] would be quite interesting, if it is possible.

Applications in scheduling. The results of this paper also have applications to stochastic scheduling (the reader is referred to recent papers [7, 8] to see similar stochastic problems). In this scheduling prolbem you are given a set of jobs $j_1, j_2, \ldots, j_r$, and a set of machines $M = \{m_1, \cdots, m_c\}$. Each job j_i has a processing time t_i and can be processed on a machine from a subset $M_i \subset M$. Further, each job has a probability p_i that describes the probability that this job appears. The goal is to schedule all jobs that appear and to minimize the makespan of the scheduling.

An intersting application of our result is to obtain an *oblivious scheduling*, i.e., fixed decisions that determine for each job on which machine it has to be scheduled if it appears, such that the makespan of the oblivious scheduling is competitive with the makespan of the optimum scheduling with high probability. Of course, we can obtain such an oblivious scheduling by considering an oblivious routing algorithm in a 3-layered graph G which has a vertex for each job j_i, $1 \le i \le r$, a vertex for each machine m_h, $1 \le h \le c$, and a super sink vertex s, an edge (j_i, m_h) of capacity t_i if job j_i can be performed in machine m_h and an edge (m_h, s), $1 \le h \le c$, of capacity one. Now, the makespan of the oblivious scheduling algorithm corresponds to the congestion of the oblivious routing algorithm. Hence we show in this paper that there is an oblivious scheduling that with high probability is $O(\text{polylog}\, n)$ competitive.

Open problems. The main open problem is whether the upper bound of $O(\log^2 n)$ on the competitive ratio in our randomized demand model can be improved. We derive our bound in a two step process. First we show that the oblivious algorithm is w.h.p. only a logarithmic factor larger than the expected optimum congestion. Then we show that the optimum congestion is w.h.p. only a logarithmic factor smaller than its expectation. This results in a competitive ratio of $O(\log^2 n)$. Both steps cannot be improved for themselves. There are instances in which the congestion of any oblivious algorithm may be an $\Omega(\log n / \log\log n)$ factor away from the expected optimum congestion, and there are instances in which the optimum congestion is far away from its expectation with fair probability. Therefore in order to improve our result one might show that there are no instances for which both theses events happen with a reasonable probability.

Another interesting open problem is to trie to reduce the competitive ratio for undirected graphs. In particular it would be interesting to know whether in the random demand model a constant competitive ratio is possible.

5. ACKNOWLEDGEMENTS

We would like to thank Yashar Ganjali who introduced us to the important and practical assumption of knowing demand distributions for oblivious routing in optical networks. Special thanks go to Anupam Gupta who pointed us to the tail estimate in [5] used in the proof of Lemma 9.

6. 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 symposium on Discrete Algorithms, ACM Press, 2004, pp. 942–951.

[2] B. Awerbuch, M. Hajiaghayi, R. Kleinberg, and T. Leighton, *Online client-server load balancing without global information*, in Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms, 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, ACM Press, 2003, pp. 383–388.

[4] M. Bienkowski, M. Korzeniowski, and H. Räcke, *A practical algorithm for constructing oblivious routing schemes*, in Proceedings of the 15th ACM Symposium on Parallel Algorithms and Architectures, ACM Press, 2003, pp. 24–33.

[5] S. Boucheron, G. Lugosi, and P. Massart, *A sharp concentration inequality with applications*, Random Structures and Algorithms, 16 (2000), pp. 277–292.

[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, ACM Press, 2004, pp. 156–165.

[7] B. Dean, M. Goemans, and J. Vondrak, *Approximating the stochastic knapsack problem: The benefit of adaptivity*, in Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, 2004, pp. 208–217.

[8] ———, *Adaptivity and approximation for stochastic packing problems*, in Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA), ACM Press, 2005, pp. 395–404.

[9] M. Hajiaghayi, R. D. Kleinberg, 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), ACM Press, 2005, pp. 782–790.

[10] C. Harrelson, K. Hildrum, and S. Rao, *A polynomial-time tree decomposition to minimize congestion*, in Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures, 2003, pp. 34–43.

[11] B. M. Maggs, G. L. Miller, O. Parekh, R. Ravi, and S. L. M. Woo, *Solving symmetric diagonally-dominant systems by preconditioning.* manuscript, 2003.

[12] R. Motwani and P. Rahgavan, *Randomized Algorithms*, Cambridge University Press, 1995.

[13] H. Räcke, *Minimizing congestion in general networks*, in Proceedings of the 43rd Symposium on Foundations of Computer Science, IEEE Computer Society, 2002, pp. 43–52.

[14] H. Räcke and A. Rosen, *Distributed online call control in general networks*, in Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 791–800.

[15] L. G. Valiant and G. J. Brebner, *Universal schemes for parallel communication*, in Proceedings of the 13th ACM symposium on Theory of Computing, 1981, pp. 263–277.