Oblivious Routing on Geometric Networks

Costas Busch
Computer Science Dept.
Rensselaer Polytechnic Inst.
Troy, NY 12180, USA
buschc@cs.rpi.edu

Malik Magdon-Ismail
Computer Science Dept.
Rensselaer Polytechnic Inst.
Troy, NY 12180, USA
magdon@cs.rpi.edu

Jing Xi
Computer Science Dept.
Rensselaer Polytechnic Inst.
Troy, NY 12180, USA
xij2@cs.rpi.edu

ABSTRACT

We study oblivious routing in which the packet paths are constructed independently of each other. We give a simple oblivious routing algorithm for geometric networks in which the nodes are embedded in the Euclidean plane. In our algorithm, a packet path is constructed by first choosing a random intermediate node in the space between the source and destination, and then the packet is sent to its destination through the intermediate node. We analyze the performance of the algorithm in terms of the stretch and congestion of the resulting paths. We show that the stretch is constant, and the congestion is near optimal when the network paths can be chosen to be close to the geodesic lines that connect the end points of the paths. We give applications of our general result to the mesh topology and uniformly distributed disc graphs. Previous oblivious routing algorithms with near optimal congestion use many intermediate nodes and do not control the stretch.

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; Performance; Theory

Keywords

Oblivious Routing; Congestion; Stretch; Geometric Networks; Disc Graphs

1. INTRODUCTION

An routing algorithm specifies the paths to be followed by packets in a network. The routing algorithm is *oblivious* if the path of every packet is specified independently of the paths of the other packets. Oblivious algorithms are by their nature distributed and capable of solving online routing problems, where packets continuously arrive in the network. We give an oblivious routing algorithm for geometric networks. In these networks, the nodes are placed in the 2-dimensional Euclidian space (Figure 1), and the edges of the network are un-weighted. We assume that all the nodes are contained in some geographic area $\mathcal{A}$.

Suppose that a packet wants to go from a node s to a node t in the network. Our algorithm is to choose a random intermediate node w in the space between the s and t, then sends the packet to w, and then sends the packet from w to its destination (see Figure 1). In order to implement this idea, we assume that between every pair of nodes there is a dedicated path which we call the *default path.* For example, the default path between two nodes u and v could be a shortest path that connects them. We denote the set of all default paths by Q. The choice of the default paths affects the performance of our algorithm, and the closer the default paths are to the geodesics, the lines that connect the respective end points of the paths, the better the performance of the algorithm.

We analyze the algorithm in terms of stretch and congestion. Consider some set of paths P produced by our algorithm. Denote by $\mathsf{stretch}(P)$ the maximum ratio of a path length to the length of the respective shortest path (the length is measured in number of node hops). The *node congestion* C_{node} is the maximum number of paths that use any node in the network. The *edge congestion* C_{edge} is the maximum number of paths that use any edge in the network. Let C^*_{node} and C^*_{edge} denote the optimal node and edge congestions, which could be obtained by a brute force search through all possible paths from the sources to the destinations in P.

The stretch and congestion of the paths P produced by our algorithm depend on the quality of the default paths Q. In particular, provided that the geometric embedding is "faithful" to the topology of the network (i.e. nodes far apart are connected with more hops than nodes closer to each other) we obtain:

$$\begin{aligned}\mathsf{stretch}(P) &= O\big(\mathsf{stretch}(Q)\big),\\ C_{node} &= O\big(C^*_{node}\cdot(1+\mathsf{deviation}^3(Q))\\ &\quad \cdot\log(n+\mathsf{deviation}(Q))\big),\end{aligned}$$

where n is the number of nodes, and $\mathsf{deviation}(Q)$ measures the extent of deviation of the default paths from geodesics (see Figure 1). We also obtain a corresponding result for the edge congestion. The congestion results hold with high probability, while the stretch result is deterministic.

SPAA'05, July 18–20, 2005, Las Vegas, Nevada, USA.

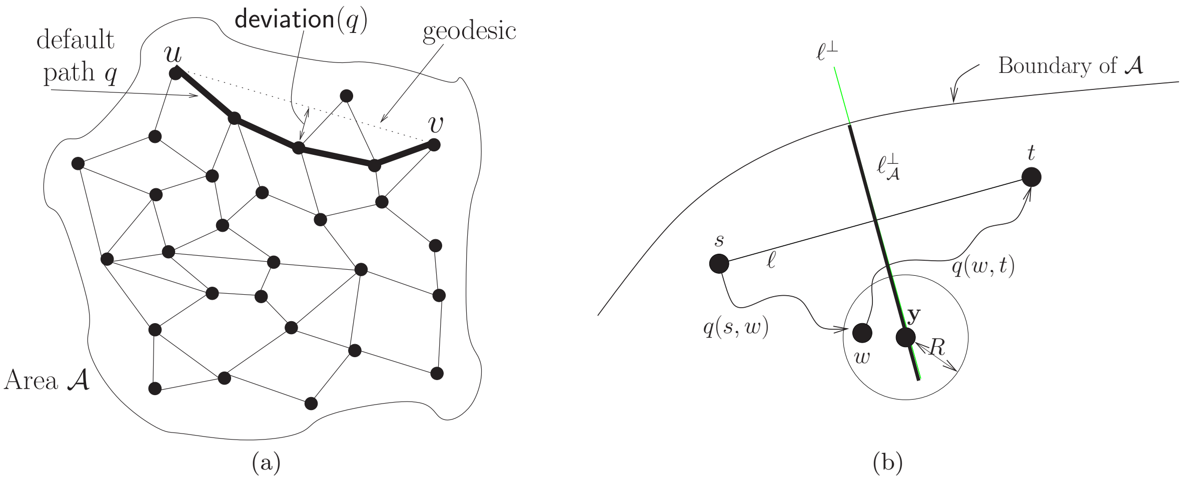


Figure 1: (a) A geometric network. (b) The algorithm.

We apply our general result to two particular geometric networks, the mesh network and uniformly distributed disc networks, which have geometric embeddings that are faithful to the network topologies. The mesh network is a 2-dimensional grid of nodes. In disc networks, each node is connected to any node within a specific disc radius. In the uniformly distributed disc graphs, each unit square area contains a constant number of nodes. In these networks, we can choose default paths with constant stretch and deviation. Therefore, our algorithm gives paths with *constant* stretch. We obtain node and edge congestions which are within logarithmic factors of optimal, $C_{node} = O(C^*_{node} \log n)$ and $C_{edge} = O(C^*_{edge} \log n)$, with high probability. Maggs *et al.* [12] give a worst case edge congestion lower bound of $\Omega(C^*_{edge} \log n)$ for any oblivious routing algorithm in the 2-dimensional mesh. Therefore, in addition to constant stretch, the congestion we obtain is optimal, within constant factors, for oblivious algorithms.

The disc networks, as well as the mesh network, are popular models for wireless networks. In many applications, wireless nodes have limited power, for example in wireless sensor networks each sensor is operated with a battery with limited energy. To maximize the lifetime of the nodes (the time until nodes run out of power), it is important to minimize the utilization of individual nodes, i.e. minimize the node congestion. Our algorithm achieves optimal node congestion (up to a $\log n$ factor), thus, it balances the packet traffic load among the nodes and extends the lifetime of the nodes. Our algorithm is easy to implement in wireless networks, on account of its simplicity and its oblivious nature, which make it a suitable choice for these networks.

1.1 Related Work

Valiant [19] is the first to propose the technique of using a single random intermediate node in order to minimize congestion in networks. He gives a general approach appropriate for oblivious routing based on solutions to flow problems in the network. Applications are permutation routing problems on the hypercube and butterfly networks.

The motivation for minimizing congestion and stretch simultaneously is because there exist packet scheduling algorithms [10, 11, 13, 15] which deliver the packets along the given paths in time very close to the optimal $O(C_{edge} + D)$, where D is the maximum path length.

Maggs *et al.* [12] give an oblivious algorithm for the d-dimensional mesh with congestion $O(dC^*_{edge} \log n)$. Following this work, there have been extensions to general networks [3, 4, 8, 16], where progressively better oblivious algorithms with near optimal (up to logarithmic factors) congestion. However, in all these algorithms the stretch is unbounded. Further, all these algorithms are based on a hierarchical decomposition of the network into clusters, which requires a logarithmic number of intermediate nodes. Our algorithm on the other hand, uses only *a single* intermediate node and doesn't depend on any hierarchical clustering. Maggs et al. [12] also give a lower bound of $\Omega(\frac{C^*_{edge}}{d} \log n)$ on the congestion of any oblivious algorithm on the mesh (thus, for $d = 2$ our algorithm has optimal congestion). A variety of other such lower bounds also exist [5, 9, 20].

In [6] we give an oblivious algorithm which simultaneously minimizes the congestion and has constant stretch. That algorithm is for the d-dimensional mesh, and is also based on a hierarchical decomposition of the network. In the same paper we show that for achieving near optimal congestion on the mesh, any oblivious routing algorithm requires an amount of random bits which is proportional to the logarithm of the distance between the source and destination. Thus, randomization is unavoidable for oblivious routing.

Non-oblivious routing algorithms with near optimal congestion and stretch are discussed in [1, 2, 17, 18]. These approaches require *a priori* knowledge of the traffic distribution. Trade offs between stretch and congestion have been studied in wireless networks [7]. Our algorithm shows that both can be controlled in special cases of wireless networks.

Paper Outline. We begin with some necessary definitions and preliminary results in Section 2. We give the description of our algorithm in Section 3. We then continue with the analysis of the algorithm in Section 4. The applications of our algorithm to the mesh and disc graphs appear in Section 5. We finish with a discussion in Section 6.

2. PRELIMINARIES

2.1 Geometric Networks

Consider a geometric network G with n nodes which is embedded in the Euclidean plane, $\mathbb{R}^2$. We assume that G is un-weighted, undirected, connected and stationary. Further, its edges are un-weighted, i.e. the communication cost of every link is 1 regardless of the link's Euclidian distance. Every node v_i has a position $\mathbf{x}_i \in \mathbb{R}^2$. We will also use the notation $\mathbf{x}(v)$ to denote the position of the node v. The network is defined over some *area* $\mathcal{A}$. We will also refer to the network itself as $\mathcal{A}$ when the context is clear. Thus, $\mathbf{x}_i \in \mathcal{A}$ for all i. For the area $\mathcal{A}$, we define a *coverage radius* $R(\mathcal{A})$ of the area as follows (we drop the $\mathcal{A}$ dependence when the context is clear). If, for every point $\mathbf{x} \in \mathcal{A}$, there is at least one node v that is located at most a (Euclidean) distance R from $\mathbf{x}$, then R is a coverage radius, i.e., from any point in $\mathcal{A}$, one needs to go a distance at most R to reach some node in the network.

We define the *pseudo-convexity* $\gamma(\mathcal{A})$ of area $\mathcal{A}$ as follows. Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{A}$, and consider the line ℓ joining $\mathbf{x}_1$ to $\mathbf{x}_2$. Let $\ell^\perp$ be a line of equal length to ℓ such that ℓ and $\ell^\perp$ are mutual perpendicular bisectors. Let $\ell^\perp_{\mathcal{A}} \subseteq \ell^\perp$ be the intersection of $\ell^\perp$ with $\mathcal{A}$. Denote by $|\ell^\perp_{\mathcal{A}}|$ the measure, or "length" of $\ell^\perp_{\mathcal{A}}$. We define the local pseudo-convexity at $\mathbf{x}_1, \mathbf{x}_2$ as $\gamma(\mathbf{x}_1, \mathbf{x}_2) = |\ell^\perp_{\mathcal{A}}|/|\ell^\perp|$. The pseudo-convexity γ of $\mathcal{A}$ is the infimum over all pairs $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{A}$ of $\gamma(\mathbf{x}_1, \mathbf{x}_1)$,

$$\gamma = \inf_{\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{A}} \gamma(\mathbf{x}_1, \mathbf{x}_2).$$

In words, γ is a lower bound on the fraction of the perpendicular bisector $\ell^\perp$ that is guaranteed to be in $\mathcal{A}$. Note that $\mathcal{A}$ is convex if $\gamma \geq \frac{1}{2}$ but that the converse is not true (consider a very thin rectangle). For any regular convex polygon, or a circle, $\gamma \geq \frac{1}{2}$. For a network embedded in a fixed area $\mathcal{A}$, γ is independent of n, which will have important consequences on the optimality of our path selection algorithm (provided that $\gamma > 0$).

Since the network is embedded in $\mathbb{R}^2$, there are two notions of distance between two nodes u, v that are useful. The first is the *Euclidean distance*, $\mathsf{dist}_E(u, v)$ which is the length of the straight line (or *geodesic*) joining the positions $\mathbf{x}(u)$ and $\mathbf{x}(v)$. For two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$, $\| \mathbf{x} - \mathbf{y} \|$ is the Euclidean distance between them. Thus, $\mathsf{dist}_E(u, v) = \| \mathbf{x}(u) - \mathbf{x}(v) \|$. The second useful distance measure is the graph-theoretic or *network distance* $\mathsf{dist}_G(u, v)$ which is the length of the shortest path in G from u to v. For any path p in G, we use $|p|$ to denote the length of the path (number of edges in the path), and we define the *Euclidean path length* $|p|_E$ to be the weighted path length, where the weights on the edges are set to the Euclidean distance between the nodes they connect.

For two nodes u, v, we use the measure $\mathsf{dist}_G(u, v)/\mathsf{dist}_E(u, v)$ to represent how well the Euclidean distances in the network embedding represent the network distances. We introduce two parameters α, β to denote lower and upper bounds for this measure. Thus, for every pair of nodes u, v,

$$\alpha \leq \frac{\mathsf{dist}_G(u, v)}{\mathsf{dist}_E(u, v)} \leq \beta$$

Thus, two nodes u, v that are connected by an edge ($\mathsf{dist}_G(u, v) = 1$) cannot be separated by more than a distance of $\frac{1}{\alpha}$. Note also that $\mathsf{dist}_G(u, v) \geq 1$, so $\mathsf{dist}_E(u, v) \geq \frac{1}{\beta}$. We thus have the following useful lemma,

LEMMA 2.1. *For any two nodes, u, v, $\mathsf{dist}_E(u, v) \geq \frac{1}{\beta}$. If u and v are adjacent then $\mathsf{dist}_E(u, v) \leq \frac{1}{\alpha}$.*

Lemma 2.1 allows us to derive an upper bound on the number of nodes that can be in a disc.

LEMMA 2.2. *Consider a disc of radius $r \geq \frac{1}{\beta}$ containing M nodes. Then $M \leq c(\beta r)^2$, where c is a constant, $c \leq 1 + \pi/(\frac{2\pi}{3} - \frac{\sqrt{3}}{2})$.*

PROOF. The intuition is that every node accounts for an area of at least π/β^2. Since the total area is πr^2, there can be at most $\pi r^2/(\pi/\beta^2) = (\beta r)^2$ nodes. The only complication is that nodes near the boundary do not take up the entire area π/β^2, as part of this area could be outside the disc. Taking this boundary phenomenon into account gives us the constant c.

To prove the lemma, consider the circle of radius $r - \frac{1}{\beta}$ with M_1 nodes, and the remaining ring from $r - \frac{1}{\beta}$ to r with M_2 nodes. Since every one of the M_1 nodes defines an area of radius $\frac{1}{\beta}$ that is completely enclosed in the disc, we have the $M_1 \leq (\beta r)^2$. Now consider the ring. The smallest area blocked off by a node occurs when the node is on the boundary, in which case the area is smallest when $r = \frac{1}{\beta}$. Some geometric considerations show that this area blocked of is at least $\frac{1}{\beta^2}(\frac{2\pi}{3} - \frac{\sqrt{3}}{2})$, and since the area of the ring is at most πr^2, $M_2 \leq (\beta r)^2 \pi/(\frac{2\pi}{3} - \frac{\sqrt{3}}{2})$. To conclude, note that $M \leq M_1 + M_2$. $\square$

2.2 Default Paths

For every pair of nodes u, v, we assume that a *default path* $q(u, v)$ in G is provided. For example, the default paths could be the shortest paths connecting the pairs of nodes. The default paths should actually have certain good properties and may not be shortest paths. Denote the set of all $n(n-1)$ default paths by the set Q. For a given default path $q(u, v)$, we define the stretch of the path, $\mathsf{stretch}(q)$, to be $|q(u, v)|/\mathsf{dist}_G(u, v)$ which is the factor by which q is longer than the shortest path between u and v.

Consider the infinite line ℓ drawn through the points $\mathbf{x}(u)$ and $\mathbf{x}(v)$. Let z be any intermediate node in the path $q(u, v)$. The *displacement* of z from ℓ is the perpendicular (Euclidean) distance from $\mathbf{x}(z)$ to ℓ. The deviation of $q(u, v)$ from ℓ, denoted $\mathsf{deviation}(q)$, is the maximum displacement of any intermediate node z of q from ℓ. In other words, $\mathsf{deviation}(q)$ measures how closely the path $q(u, v)$ stays to the straight line (geodesic) from $\mathbf{x}(u)$ to $\mathbf{x}(v)$.

The stretch factor for the entire set of paths Q is the maximum stretch of any path in Q, and similarly with the deviation of Q. We use Σ to denote the stretch and Δ to denote the deviation,

$$\Sigma_Q = \mathsf{stretch}(Q) = \max_{q \in Q} \mathsf{stretch}(q),$$
$$\Delta_Q = \mathsf{deviation}(Q) = \max_{q \in Q} \mathsf{deviation}(q).$$

As we will see later in the analysis of our path selection algorithm, if the default paths have small stretch and small deviation, then the path selection performance is closer to optimal. Thus it is beneficial to select default paths that make these parameters as small as possible. We will see later that for a variety of networks they can be made constants.

3. ALGORITHM

Here we describe our oblivious routing algorithm. The task of the algorithm is to provide a path for each packet in the network. It is assumed that each node knows the default paths that connect it to other nodes in the network. The algorithm is randomized and we assume that each node has access to a sequence of random numbers ([6] gives a lower bound on the number of random bits necessary if near optimal congestion is to be obtained).

The path selection algorithm is executed for each packet independently of every other packet, so the algorithm is oblivious, and thus distributed and online. Algorithm 1 is the detailed algorithm for a particular packet. The algorithm is similar for any other packet. Figure 1(b) illustrates graphically the algorithm.

Algorithm 1 Routing Algorithm

Input: A graph G embedded in an area $\mathcal{A}$ with default paths Q; and a packet π with source s and destination t;

Output: A path $p(s,t)$ from s to t;

1: Let ℓ be the geodesic line segment that connects $\mathbf{x}(s)$ and $\mathbf{x}(t)$. Let $\ell^\perp$ be the perpendicular bisector of ℓ which has the same length as ℓ and is also bisected by ℓ. Let $\ell^\perp_\mathcal{A}$ be the part of $\ell^\perp$ inside $\mathcal{A}$;

2: Choose a point $\mathbf{y}$ randomly and uniformly on $\ell^\perp_\mathcal{A}$;

3: Find a node w close to $\mathbf{y}$ within coverage radius R;

4: The path $p(s,t)$ from s to t is formed by concatenating the default paths $q(s,w)$ and $q(w,t)$:

$$p(s,t) = q(s,w)q(w,t);$$

4. ANALYSIS

Here, we give the analysis of the algorithm (described in Section 3) for a graph G embedded in an area $\mathcal{A}$.

In the analysis, we will consider a set of N packets Π which we will refer to with their sources and destinations, $\Pi = \{s_i, t_i\}_{i=1}^N$. The result of applying the algorithm to each packet is a set of paths $P = \{p_i\}_{i=1}^N$, where each path $p_i \in P$ is from the source node s_i to the destination node t_i. We define the stretch for a path $p \in P$ as well as the stretch factor for the entire set P as we did in Section 2 with the default paths Q. We define D^* as the maximum shortest path length between any pair of sources and destinations in Π, namely, $D^* = \max_i \mathsf{dist}_G(s_i, t_i)$.

We first analyze the stretch of paths P and then we continue with the node-congestion and edge-congestion.

4.1 Stretch

We now give a bound on $\mathsf{stretch}(P)$, the stretch factor of the paths selected.

THEOREM 4.1. $\mathsf{stretch}(P) \le \frac{\sqrt{2}\beta}{\alpha} \cdot \Sigma_Q \cdot (1 + \sqrt{2}R\alpha)$.

PROOF. We will refer to Figure 1 in our proof. By construction, $\sqrt{2}\|\mathbf{x}(s) - \mathbf{y}\| \le \|\mathbf{x}(s) - \mathbf{x}(t)\|$, and $\|\mathbf{x}(s) - \mathbf{y}\| = \|\mathbf{x}(t) - \mathbf{y}\|$. Since $\|\mathbf{x}(w) - \mathbf{y}\| \le R$, by the triangle inequality, we have that

$$\begin{aligned}\|\mathbf{x}(s) - \mathbf{x}(w)\| &\le \|\mathbf{x}(s) - \mathbf{y}\| + \|\mathbf{x}(w) - \mathbf{y}\| \\ &\le \tfrac{1}{\sqrt{2}}\|\mathbf{x}(s) - \mathbf{x}(t)\| + R.\end{aligned}$$

Similarly,

$$\|\mathbf{x}(t) - \mathbf{x}(w)\| \le \tfrac{1}{\sqrt{2}}\|\mathbf{x}(s) - \mathbf{x}(t)\| + R.$$

From the definition of Σ_Q, the stretch factor of the default paths, we have that

$$\begin{aligned}|q(s,w)| &\le \Sigma_Q \cdot \mathsf{dist}_G(s,w) \\ &\le \beta \cdot \Sigma_Q \cdot \mathsf{dist}_E(s,w),\end{aligned}$$

and similarily

$$|q(w,t)| \le \beta \cdot \Sigma_Q \cdot \mathsf{dist}_E(w,t).$$

We thus conclude that

$$\begin{aligned}|p(s,t)| &= |q(s,w)| + |q(w,t)|, \\ &\le \beta \cdot \Sigma_Q \cdot (\|\mathbf{x}(s) - \mathbf{x}(w)\| + \|\mathbf{x}(t) - \mathbf{x}(w)\|), \\ &\le \beta \cdot \Sigma_Q \cdot (\sqrt{2}\|\mathbf{x}(s) - \mathbf{x}(t)\| + 2R).\end{aligned}$$

Since $\|\mathbf{x}(s) - \mathbf{x}(t)\| \le \frac{1}{\alpha}\mathsf{dist}_G(s,t)$, and $\mathsf{dist}_G(s,t) \ge 1$, we obtain the theorem. □

Typically R, α, β are constants, in which case $\mathsf{stretch}(P) = O(\mathsf{stretch}(Q))$, i.e., the stretch factor of the algorithm is determined by the quality of the default paths.

4.2 Node Congestion

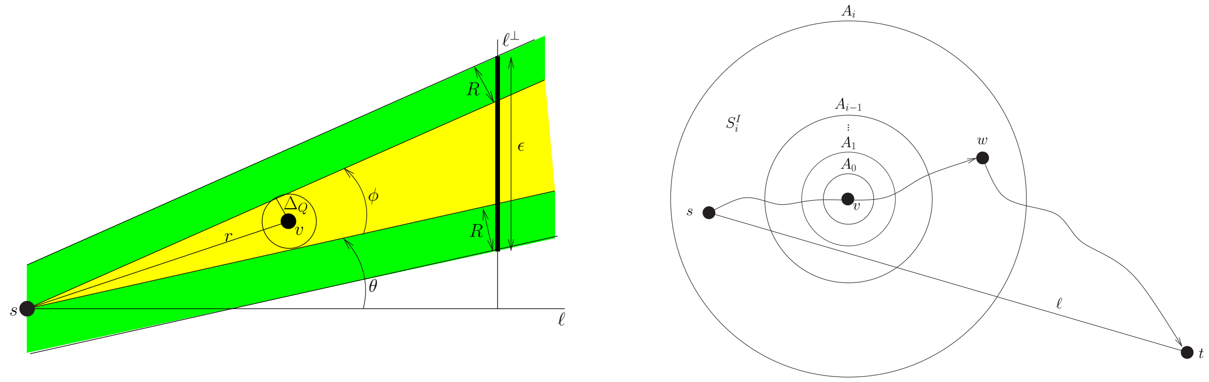
We now turn to the node congestion. We will get a bound on the expected congestion for any particular node with respect to the optimal congestion. We will then use a Chernoff bounding argument to obtain a high probability result.

To bound the expected node congestion for a particular node v, we need to understand the probability that a particular packet might use the node. Thus consider a particular packet π, with source s and destination t, which uses intermediate node w. Phase I of the path $p(s,t)$ corresponds to the first part $q(s,w)$, while phase II to the second part $q(w,t)$. Suppose that the packet uses v in phase I of its path (we will bound the probability that π uses v in phase I of its path, a similar argument applies to phase II of the path). Let r denote $\|\mathbf{x}(v) - \mathbf{x}(s)\|$. The situation is illustrated in Figure 2(a). A circle of radius Δ_Q is drawn around v. We give an upper bound on the probability that π uses node v in the following lemma,

LEMMA 4.2. *Suppose that packet π has source s and destination t. Let P_I be the probability that π uses node v in phase I of its path and P_{II} be the probability that π uses node v in phase II of its path. Then,*

$$\begin{aligned}P_I &\le \frac{5}{\gamma}\left(\frac{R}{\|\mathbf{x}(s) - \mathbf{x}(t)\|} + \frac{\Delta_Q}{\|\mathbf{x}(s) - \mathbf{x}(v)\|}\right), \\ P_{II} &\le \frac{5}{\gamma}\left(\frac{R}{\|\mathbf{x}(s) - \mathbf{x}(t)\|} + \frac{\Delta_Q}{\|\mathbf{x}(t) - \mathbf{x}(v)\|}\right).\end{aligned}$$

PROOF. Consider the shaded cone subtended by the source s, tangent to the circle of radius Δ_Q centered on v. Since the deviation of the default paths is Δ_Q, The intermediate node must lie within the shaded cone if the path $q(s,w)$ is to pass through v. If the intermediate node is in the cone, the random intermediate point $\mathbf{y}$ must lie either in the cone or in one of the two shaded strips of thickness R around the cone. Since $\mathbf{y}$ must also be on $\ell^\perp$, $\mathbf{y}$ must lie on the line segment illustrated by the thick line of length ϵ illustrated in Figure 2(a). The probability to use v is then bounded by $\epsilon/|\ell^\perp_\mathcal{A}|$. We use the definitions of θ, ϕ as shown



(a) (b)

Figure 2: (a) Probability of using a node v. (b) Expected congestion at a node v.

in Figure 2(a). Using some elementary geometry, we find that

$$\epsilon = R \cdot \left(\frac{1}{\cos(\theta)} + \frac{1}{\cos(\theta + \phi)} \right) + \frac{1}{2}|\ell| \cdot (\tan(\theta + \phi) - tan(\theta)).$$

We observe that ϵ is largest when $\theta \le \frac{\pi}{4}$ and $\theta + \phi \le \frac{\pi}{4}$, so using some trignometric identities, we have that

$$\begin{aligned} \epsilon &\le 2\sqrt{2}R + \frac{|\ell|}{2} \cdot \frac{\tan\phi(1+\tan^2\theta)}{1 - \tan\theta\tan\phi}, \\ &\le 2\sqrt{2}R + |\ell| \cdot \frac{\tan\phi}{1-\tan\phi}, \end{aligned}$$

where the last line follows because $\tan\theta < 1$. Since $|\ell^{\perp}_{\mathcal{A}}| \ge \gamma|\ell^{\perp}| = \gamma|\ell|$, we have that the probability to use v is at most

$$\begin{aligned} \text{Prob} &\le 2\sqrt{2}\frac{R}{\gamma|\ell|} + \frac{1}{\gamma}\frac{\tan\phi}{1-\tan\phi}, \\ &\overset{(a)}{\le} 2\sqrt{2}\frac{R}{\gamma|\ell|} + \frac{2\tan\phi}{\gamma}, \\ &\overset{(b)}{=} 2\sqrt{2}\frac{R}{\gamma|\ell|} + \frac{4}{\gamma}\frac{\tan\frac{\phi}{2}}{1-\tan^2\frac{\phi}{2}}, \\ &\overset{(c)}{\le} 2\sqrt{2}\frac{R}{\gamma|\ell|} + \frac{64\tan\frac{\phi}{2}}{15\gamma}, \\ &\overset{(d)}{=} 2\sqrt{2}\frac{R}{\gamma|\ell|} + \frac{64}{15\gamma}\frac{\Delta_Q/r}{\sqrt{1-\Delta_Q^2/r^2}}, \\ &\overset{(e)}{\le} \frac{5}{\gamma}\left(\frac{R}{|\ell|} + \frac{\Delta_Q}{r}\right). \end{aligned}$$

Inequality (a) follows because when $\tan\phi \le \frac{1}{2}$, $\tan\phi/(1-\tan\phi) \le 2\tan\phi$, and when $\tan\phi > \frac{1}{2}$, $2\tan\phi > 1$, in which case it is a trivially valid upper bound for the probability. (b) follows by using a double angle identity. (c) follows by a similar argument that lead to (a) by considering separately $\tan\frac{\phi}{2} \le \frac{1}{4}$ and $\tan\frac{\phi}{2} > \frac{1}{4}$. (d) follows because from Figure 2(a), we see that $\tan\frac{\phi}{2} = \Delta_Q/\sqrt{r^2 - \Delta_Q^2}$. Finally, (e) follows using $2\sqrt{2} < 5$ and by considering separately the cases $\Delta_Q/r \le \frac{1}{5}$ and $\Delta_Q/r > \frac{1}{5}$ (similar with (a) and (c)).

To conclude, note that by symmetry, the situation is exactly reversed if the packet uses v in phase II of its path, except that now r will be the distance from v to the destination t. □

In order to bound the congestion on node v, we need to bound the number of packets that can cross v, and then using Lemma 4.2 we will be able to bound the expected congestion on v. We first compute how far the packets that cross v have their sources or destinations from v, this will help to bound the number of those packets. Let X^I denote the packets that could possibly use v during phase I of their path, and similarly X^{II}. Consider only the packets in X^I. Let $S^I = \{s_k\}$ denote the sources of all the packets in X^I. Let r_{max} be the maximum (Euclidean) distance from the positions of these sources to $\mathbf{x}(v)$, thus, $r_{max} = \max_{s \in S^I} \| \mathbf{x}(s) - \mathbf{x}(v) \|$. We have the following result (a similar result holds for the destinations).

Lemma 4.3. $r_{max} \le \frac{D^*}{\sqrt{2}\alpha} + R + \Delta_Q$.

Proof. Let s be a source that could possibly use v in phase I and let t be the corresponding destination. Let w be a possible intermediate node. Then $\| \mathbf{x}(s) - \mathbf{x}(w) \| \le \frac{1}{\sqrt{2}} \| \mathbf{x}(s) - \mathbf{x}(t) \| + R$. Since the path cannot deviate by more than Δ_Q from the line joining $\mathbf{x}(s)$ to $\mathbf{x}(w)$, and the path passes through v, it follows that

$$\begin{aligned} \| \mathbf{x}(v) - \mathbf{x}(s) \| &\le \| \mathbf{x}(s) - \mathbf{x}(w) \| + \Delta_Q \\ &\le \frac{1}{\sqrt{2}} \| \mathbf{x}(s) - \mathbf{x}(t) \| + R + \Delta_Q. \end{aligned}$$

To conclude, note that $\| \mathbf{x}(s) - \mathbf{x}(t) \| \le \frac{1}{\alpha}\mathsf{dist}_G(s,t)$ and $\mathsf{dist}_G(s,t) \le D^*$. □

In order to bound the congestion on v, we will divide the area around v into concentric rings with maximum radius r_{max}. We will then bound the number of packets that originate in each ring and use v. The number of packets from each ring will be used to bound the expected congestion caused by each ring. The sum of the expected congestions from the rings will determine the total congestion on node v.

Consider concentric rings $A_0, A_1, A_2, \ldots$ of exponentially increasing radius, centered at $\mathbf{x}(v)$. Ring A_i has radius $r_i = 2^i/\beta$, for $i \ge 0$. Let $i_{max} = \lceil \log(r_{max}\beta) \rceil$ (logarithms are base 2). Note that all the sources in S^I are contained in $A_{i_{max}}$. For $i > 0$, we collect in set S_i^I all the sources which are in ring A_i, but not in A_{i-1} (that is, they are in area between A_{i-1} and A_i). Figure 2(b) illustrates the situation. Consider a particular i and the packets X_i^I with sources in S_i^I. Let $N_i = |X_i^I|$ be the number of packets with sources in S_i^I. In order to obtain an upper bound on the expected congestion at v, we will need to bound N_i in terms of the optimal node congestion C^*_{node}.

LEMMA 4.4. *For any* $i \geq 0$:

$$C^*_{node} \geq \frac{\alpha h_i N_i}{4c(\beta r_i)^2},$$

where,

$$h_i = \max\left\{\frac{1}{\beta}, \sqrt{2}(r_{i-1} - R - \Delta_Q)\right\}.$$

PROOF. As in the proof of Lemma 4.3, $\|\mathbf{x}(s) - \mathbf{x}(v)\| \leq \frac{|\ell|}{\sqrt{2}} + R + \Delta_Q$, and since $\|\mathbf{x}(s) - \mathbf{x}(v)\| \geq r_{i-1}$, we get

$$|\ell| \geq \sqrt{2}(r_{i-1} - R - \Delta_Q).$$

From Lemma 2.1, $|\ell| \geq \frac{1}{\beta}$, therefore $|\ell| \geq h_i$. Furthermore, from the definition of α and Lemma 2.1, we have that the minimum number of hops from s to t is at least $\alpha|\ell| \geq \alpha h_i$, and each of these hops moves a distance at most $\frac{1}{\alpha}$. So, for the N_i such paths, any path selection algorithm will have to use at least αh_i hops per path, within a disc of radius

$$r = r_i + h_i \leq r_i + 2r_{i-1} \leq 2r_i.$$

By Lemma 2.2, there are at most $4c(\beta r_i)^2$ nodes within this disc of radius r. The minimum total number of times these nodes are used by *any* path selection algorithm is $\alpha h_i N_i$. Thus, the average number of times T_{avg} a node is used in radius r is at least

$$T_{avg} \geq \frac{\alpha h_i N_i}{4c(\beta r_i)^2},$$

where c is the constant defined in Lemma 2.2. Since one of these nodes has to be used at least T_{avg} times, we obtain a lower bound on the congestion for any path selection algorithm, and hence for the optimal congestion $C^*_{node} \geq T_{avg}$. $\square$

Note that inverting the bound in Lemma 4.4, we get an upper bound for N_i, when $i \geq 1$:

$$N_i \leq \frac{4c(\beta r_i)^2 C^*_{node}}{\alpha h_i} \qquad (1)$$

Note that for $i = 0$ it holds trivially that $N_0 \leq 0$, since no node except for v can be in ring A_0 (a consequence of Lemma 2.1). The upper bound for N_i together with the upper bound for the probability that any of these packets uses node v (Lemma 4.2) allows us to bound the expected congestion,

THEOREM 4.5. *The expected congestion on node v is*

$$E[C(v)] \leq f(\gamma, \alpha, \beta, R, \Delta_Q, D^*) \cdot C^*_{node},$$

where,

$$\begin{aligned} &f(\gamma, \alpha, \beta, R, \Delta_Q, D^*) = \\ &\frac{40c\beta^2(R + 2\Delta_Q)}{\gamma\alpha} \cdot ((3 + 4\beta(R + \Delta_Q))^2 \\ &+4\log\left(\frac{\beta D^*}{\sqrt{2}\alpha} + \beta(R + \Delta_Q)\right) + 1). \end{aligned}$$

PROOF. Let $Prob_v(\pi)$ be the probability that packet $\pi \in X_i^I$ uses node v. Then packet π's contribution to the expected node congestion at v is $Prob_v(\pi)$. Using Lemma 4.2, we can bound $Prob_v(\pi)$ by P_I. Then, $N_i P_I$ is an upper bound for the contribution to the expected node congestion at v due to the packets in X_i^I. Since every source in S_i^I is distance at least r_{i-1} from node v, from Lemma 4.2 and using (1), and the fact that $r_i \geq h_i$, we obtain for $i \geq 1$:

$$\begin{aligned} \sum_{\pi \in X_i^I} Prob_v(\pi) &\leq N_i P_I \\ &\leq \frac{20c(\beta r_i)^2 C^*_{node}}{\gamma\alpha h_i}\left(\frac{R}{h_i} + \frac{\Delta_Q}{r_{i-1}}\right) \\ &= \frac{20c\beta^2 C^*_{node}}{\gamma\alpha}\left(R\frac{r_i^2}{h_i^2} + 2\Delta_Q\frac{r_i}{h_i}\right) \\ &\leq \frac{20c\beta^2(R + 2\Delta_Q)C^*_{node}}{\gamma\alpha} \cdot \frac{r_i^2}{h_i^2}. \end{aligned}$$

The expected node congestion at v, is obtained by summing the contributions due to each set X_i^I for $i = 1, \ldots, i_{max}$. Thus,

$$E[C(v)] \leq \frac{20c\beta^2(R + 2\Delta_Q)C^*_{node}}{\gamma\alpha}\sum_{i=1}^{i_{max}}\frac{r_i^2}{h_i^2}.$$

Consider now the ratio h_i/r_i. We have

$$\begin{aligned} \frac{h_i}{r_i} &= \frac{\max\left\{\frac{1}{\beta}, \sqrt{2}(r_{i-1} - R - \Delta_Q)\right\}}{r_i} \\ &= \max\left\{\frac{1}{2^i}, \sqrt{2}\left(\frac{1}{2} - \frac{\beta(R + \Delta_Q)}{2^i}\right)\right\}. \end{aligned}$$

Let

$$i^* = \left\lceil \log(\sqrt{2} + 4\beta(R + \Delta_Q)) \right\rceil.$$

Then for $i \geq i^*$, it holds $\frac{h_i}{r_i} \geq \frac{\sqrt{2}}{4}$, or equivalently $\frac{r_i}{h_i} \leq 2^{3/4} < 2$. For $1 \leq i < i^*$, we have that $\frac{h_i}{r_i} \geq \frac{1}{2^i}$, or in other words, $\frac{r_i}{h_i} \leq 2^i$. Since $i_{max} = \lceil \log(r_{max}\beta) \rceil$, using the bound in Lemma 4.3, we get:

$$\begin{aligned} \sum_{i=1}^{i_{max}}\frac{r_i^2}{h_i^2} &= \sum_{i=1}^{i^*-1}\frac{r_i^2}{h_i^2} + \sum_{i=i^*}^{i_{max}}\frac{r_i^2}{h_i^2} \\ &\leq \sum_{i=1}^{i^*-1}4^i + \sum_{i=i^*}^{i_{max}}4 \\ &\leq (2^{i^*})^2 + 4i_{max}, \\ &\leq (3 + 4\beta(R + \Delta_Q))^2 \\ &\quad +4\log\left(\frac{\beta D^*}{\sqrt{2}\alpha} + \beta(R + \Delta_Q)\right) + 1. \end{aligned}$$

A symmetrical argument applies to the second phase of the paths, which contributes an additional factor of 2, concluding the proof. $\square$

Note that without increasing the expected congestion, we can always remove any cycles in a path, so without loss of generality, we will assume that the paths are acyclic. We now obtain a concentration result on the congestion using a straightforward Chernoff bounding argument and the fact that every packet selects its path independently of every other packet. To simplify the presentation, we give the result for constant γ, α, β, R in which case Theorem 4.5 gives

$$E[C(v)] = O\left(C^*_{node} \cdot (\Delta_Q^3 + (1+\Delta_Q)\log(D^* + \Delta_Q))\right).$$

The general case can be handled similarly. We have the following theorem.

THEOREM 4.6. *When γ, α, β, R are constants, the node congestion is*

$$C_{node} = O\left(C^*_{node} \cdot (1+\Delta_Q^3) \cdot \log(n + \Delta_Q)\right),$$

with high probability.

PROOF. Let $X_i = 1$ if path $p(s_i, t_i)$ uses node v, and $X_i = 0$ otherwise. Then, by Theorem 4.5, there is a constant A such that

$$\begin{aligned} E[C(v)] &= E[\sum_i X_i] \\ &\leq A \cdot C^*_{node} \\ &\quad \cdot \left(\Delta_Q^3 + (1+\Delta_Q)\log(D^* + \Delta_Q)\right) \\ &\leq A \cdot C^*_{node} \\ &\quad \cdot \left(\Delta_Q^3 \log n + (1+\Delta_Q)\log(n(D^* + \Delta_Q))\right) \\ &:= B. \end{aligned}$$

Let $\kappa > 2e$. Since $\sum_i X_i$ is a sum of independent Bernoulli trials, by applying a Chernoff bound [14] we obtain

$$P[C(v) > \kappa B] < 2^{-\kappa B} \leq 1/n^{\kappa A},$$

where we used the facts that $C^*_{node}, D^* \geq 1$ and $\Delta_Q \geq 0$. Taking a union bound over the n nodes multiplies by an additional n, reducing the exponent on the right to $\kappa A - 1$. Choosing κ large enough, and noting that $D^* = O(n)$, we obtain the theorem. □

4.3 Edge Congestion

For the edge congestion, the proof is similar as in the node congestion. In order to carry through the same analysis, we need an upper bound on the number of edges in the area, so we can get a lower bound on the average edge congestion. If the maximum degree (maximum number of edges adjacent per node) in the network is δ, then the maximum number of edges is at most a factor δ times the maximum number of nodes. Therefore, the result is that the optimal edge congestion is at most a factor δ smaller than the optimal node congestion, giving the following theorem for the expected edge congestion,

THEOREM 4.7. *Let δ be the maximum node degree. The expected congestion on an edge e is*

$$E[C(e)] \leq \delta \cdot f(\gamma, \alpha, \beta, R, \Delta_Q, D^*) \cdot C^*_{edge}.$$

A concentration result can be also obtained for the edge congestion.

THEOREM 4.8. *When γ, α, β, R are constants, the edge congestion is*

$$C_{edge} = O\left(\delta \cdot C^*_{edge} \cdot (1+\Delta_Q^3) \cdot \log(n + \Delta_Q)\right),$$

with high probability.

5. APPLICATIONS

5.1 Mesh

The 2-dimensional mesh is an $\sqrt{n} \times \sqrt{n}$ grid of nodes, where each node is connected with at most 4 adjacent neighbors (see Figure 3). The nodes are placed at a unit distance from each other, and thus $R = 1/\sqrt{2}$. The area $\mathcal{A}$ is a square defined by the border nodes of the mesh so the pseudo-convexity $\gamma = 1/2$. For the default path between a pair of nodes, we choose the shortest path that connects the nodes which is closest to the geodesic and therefore $\mathsf{deviation}(Q) \leq 1/\sqrt{2}$. Since the default paths are shortest paths, $\mathsf{stretch}(Q) = 1$. Since adjacent nodes cannot be further than a unit distance, we have that $\alpha = 1$. Moreover, the number of nodes used per unit distance in the shortest path is maximized when the geodesic between the nodes is 45 degrees, which gives $\beta = \sqrt{2}$. Since the maximum node degree is 4, using Theorems 4.1, 4.6 and 4.8, we obtain:

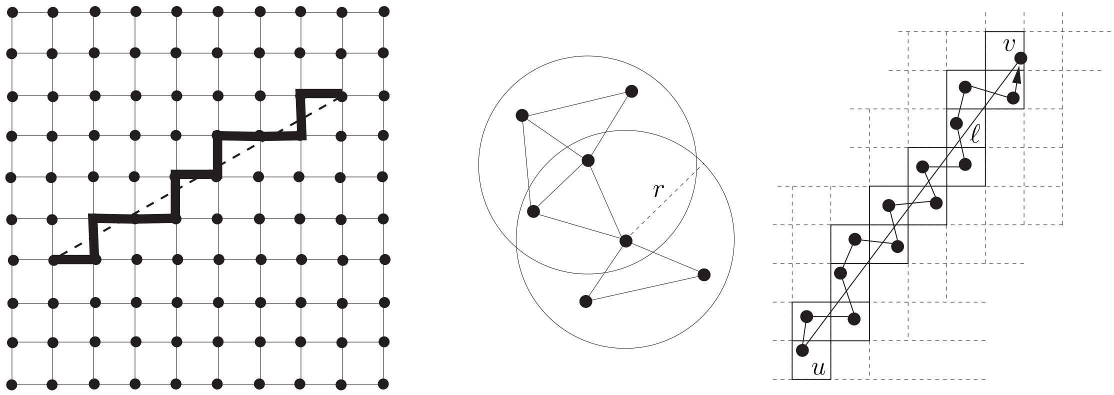
THEOREM 5.1. *The oblivious algorithm on the mesh has $\mathsf{stretch}(P) < 2\sqrt{2}$, and node-congestion $O(C^*_{node} \cdot \log n)$ and edge-congestion $O(C^*_{edge} \cdot \log n)$ with high probability.*

5.2 Uniform Disc Graphs

We consider *uniform disc graphs* with n nodes distributed in an $s_1 \times s_2$ rectangle area $\mathcal{A}$, with constant pseudo-convexity $\gamma = \min\{s_1, s_2\}/2\max\{s_1, s_2\}$ (i.e., the sides are proportional to each other). In a disc graph, each node has a constant radius r and is connected to any node within this radius (see figure 3). We set the radius $r = 2\sqrt{2}$, and assume that no two nodes are placed within a constant distance l of each other. We consider a uniform distribution for the nodes in the area, i.e., the area is divided into non-overlapping unit squares, and every unit square area contains a number of nodes between 1 and $k = O(\frac{1}{l^2})$ nodes, where k is a constant. By the choice of r, two nodes within the same square or in adjacent squares will be connected. Thus, $R \leq \sqrt{2}$, and since there are at most 32 squares containing nodes which could possibly be adjacent to a particular node, the maximum node degree is bounded by $\delta \leq 32k$.

We now explain how to construct the default paths (see Figure 3). Consider two nodes u and v in area $\mathcal{A}$ and construct the line ℓ that connects $\mathbf{x}(u)$ to $\mathbf{x}(v)$. This line passes through a collection of unit squares, forming a path with adjacent unit squares. We pick one node from each square and construct the default path by connecting these nodes. Since for every node in the path, the line passes through the corresponding unit square containing the node, $\mathsf{deviation}(Q) \leq \sqrt{2}$. The number of unit squares in the formation of the default path is no more than $2|\ell|$, so the longest default path consists of at most $2|\ell|$ nodes. The shortest path has to use at least $|\ell|/r$ nodes; therefore, $\mathsf{stretch}(Q) \leq 2r$. Since $\mathsf{dist}_G(u,v) \geq \mathsf{dist}_E(u,v)/r$, $\alpha \geq 1/r$. If $\mathsf{dist}_G(u,v) = 1$, $\mathsf{dist}_E(u,v) \geq l$, so $\mathsf{dist}_G(u,v)/\mathsf{dist}_E(u,v) \leq \frac{1}{l}$. More generally, we know that $\mathsf{dist}_G(u,v) \leq 2|\ell|$ since the default path has $2|\ell|$ hops, so the shortest path cannot have more. Thus, $\mathsf{dist}_G(u,v)/\mathsf{dist}_E(u,v) \leq \max\{2, 1/l\}$, so $\beta \leq \max\{2, 1/l\}$, Applying Theorems 4.1, 4.6 and 4.8, we obtain:

THEOREM 5.2. *On uniform disc graphs, the oblivious routing algorithm has $\mathsf{stretch}(P) = O(1)$, and node-congestion $O(C^*_{node} \cdot \log n)$ and edge-congestion $O(C^*_{edge} \cdot \log n)$ with high probability.*



(a) (b)

Figure 3: (a) The Mesh. (b) Connectivity of a disc graph, and default path construction.

6. DISCUSSION

We have given an oblivious routing algorithm which controls both stretch and congestion for arbitrary networks using geometric embeddings: the stretch and congestion are bounded in terms of the embedding parameters which reflect how faithfully the embedding represents the network topology and how good the default paths are. The stretch of the resulting paths depend on the stretch of the default paths, along which packets are sent in the network. When the default paths are close to the geodesics, then the algorithm achieves small congestion. We gave applications of our general result to the mesh network and uniform disc graphs, where we obtained constant stretch, and congestion within a logarithmic factor from optimal.

In general, the congestion and stretch cannot be minimized simultaneously, as we show below with an example. Thus, stretch and congestion can be simultaneously minimized only on particular network topologies and we give a general parameterization of network topologies in terms of embedding properties for which near optimal results can be obtained. To see that in general stretch and congestion cannot be simultaneously near optimal, consider a network in which there are two main nodes and $\sqrt{n}$ disjoint paths that connect the two main nodes, where the first path has length 1 (an edge connects directly the main nodes), while the remaining paths have length $\Theta(\sqrt{n})$. Consider now $N = \sqrt{n}$ packets which all wish to go from one main node to the other. If all the packets use the first path then the stretch is optimal, and the congestion is $\Omega(N)$ from optimal. Any other path assignment leads to stretch N, which is a factor N from optimal.

Using the same example we can also show that $C + D$ cannot be minimized in oblivious algorithms. For example, consider the case where $N = 1$. The single packet has to follow the first path with $C + D = 2$, since any other choice would give suboptimal $C + D > 2$. Since the algorithm is oblivious, every packet has to follow the first path, even when $N > 1$. For $N = k\sqrt{n}$, this results to paths with $C + D = \Theta(k\sqrt{n})$, while the optimal choice is to send k packets along each available path which results to $C + D = \Theta(k + \sqrt{n})$. Thus, for $k > \sqrt{n}$ the oblivious solution differs from the optimal solution by a factor of $\sqrt{n}$.

An interesting open issue is to study networks in which γ is not a constant. Another problem is to study Euclidian embeddings of other known network architectures and compute the resulting congestion and stretch. It is also interesting to explore optimal algorithms using other parameterizations of the network topology.

Acknowledgements. We are grateful to the reviewers of this conference for their valuable comments and suggestions.

7. REFERENCES

[1] J. Aspens, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. Online load balancing with applications to machine scheduling and virtual circuit routing. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 623–631, 1993.

[2] B. Awerbuch and Y. Azar. Local optimization of global objectives: competitive distributed deadlock resolution and resource allocation. In *Proceedings of 35th Annual Symposium on Foundations of Computer Science*, pages 240–249, Santa Fe, New Mexico, 1994.

[3] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Racke. Optimal oblivious routing in polynomial time. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 383–388, San Diego, CA, June 2003. ACM Press.

[4] Marcin Bienkowski, Miroslaw Korzeniowski, and Harald Räcke. A practical algrorithm for constructing oblivious routing schemes. In *Proceedings of the 15th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 24–33, Jun. 2003.

[5] A. Borodin and J. E. Hopcroft. Routing, merging, and sorting on parallel models of computation. *Journal of Computer and System Science*, 30:130–145, 1985.

[6] Costas Busch, Malik Magdon-Ismail, and Jing Xi. Optimal oblivious path selection on the mesh. In *Proceedings of the 19th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2005)*, Denver, Colorado, April 2005.

[7] Jie Gao and Li Zhang. Tradeoffs between stretch factor and load balancing ratio in routing on growth restricted graphs. In *PODC '04: Proceedings of the*

twenty-third annual ACM symposium on Principles of distributed computing, pages 189–196, New York, NY, USA, 2004.

[8] Chris Harrelson, Kristen Hildrum, and Satish Rao. A polynomial-time tree decomposition to minize congestion. In *Proceedings of the 15th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 34–43, Jun. 2003.

[9] Christos Kaklamanis, Danny Krizanc, and Thanasis Tsantilas. Tight bounds for oblivious routing in the hypercube. In *Proceedings of 2nd IEEE Symposium on Parallel and Distributed Processing (2nd SPAA 90)*, pages 31–36, Crete, Greece, July 1990.

[10] F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet routing and job-scheduling in $O(congestion + dilation)$ steps. *Combinatorica*, 14:167–186, 1994.

[11] Tom Leighton, Bruce Maggs, and Andrea W. Richa. Fast algorithms for finding O(congestion + dilation) packet routing schedules. *Combinatorica*, 19:375–401, 1999.

[12] B. M. Maggs, F. Meyer auf der Heide, B. Vöcking, and M. Westerman. Exploiting locality in data management in systems of limited bandwidth. In *Proceedings of the 38th Annual Symposium on the Foundations of Computer Science*, pages 284–293, 1997.

[13] Friedhelm Meyer auf der Heide and Berthold Vöcking. Shortest-path routing in arbitrary networks. *Journal of Algorithms*, 31(1):105–131, April 1999.

[14] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, UK, 2000.

[15] Rafail Ostrovsky and Yuval Rabani. Universal $O(\text{congestion}+\text{dilation}+\log^{1+\varepsilon} N)$ local control packet switching algorithms. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, pages 644–653, New York, May 1997.

[16] Harald Räcke. Minimizing congestion in general networks. In *Proceedings of the 43rd Annual Symposium on the Foundations of Computer Science*, pages 43–52, Nov. 2002.

[17] P. Raghavan and C. D. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7:365–374, 1987.

[18] A. Srinivasan and C-P. Teo. A constant factor approximation algorithm for packet routing, and balancing local vs. global criteria. In *Proceedings of the ACM Symposium on the Theory of Computing (STOC)*, pages 636–643, 1997.

[19] L. G. Valiant. A scheme for fast parallel communication. *SIAM Journal on Computing*, 11:350–361, 1982.

[20] L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, pages 263–277, May 1981.