Trade-offs between Stretch Factor and Load Balancing Ratio in Routing on Growth Restricted Graphs

Jie Gao, *Member, IEEE,* and Li Zhang, *Member, IEEE,*

***Abstract*— An unweighted graph has density ρ and growth rate k if the number of nodes in every ball with radius r is bounded by ρr^k. The communication graphs of wireless networks and peer-to-peer networks often have constant bounded density and small growth rate. In this paper we study the trade-off between two quality measures for routing in growth restricted graphs. The two measures we consider are the stretch factor, which measures the lengths of the routing paths, and the load balancing ratio, which measures the evenness of the traffic distribution. We show that if the routing algorithm is required to use paths with stretch factor c, then its load balancing ratio is bounded by $O(\rho^{1/k}(n/c)^{1-1/k})$, and the bound is tight in the worst case. We show the application and extension of the trade-off to the wireless network routing and VLSI layout design. We also present a load balanced routing algorithm with the stretch factor constraint in an online setting, in which the routing requests come one by one.**

***Index Terms*— Routing, load balancing, wireless networks, growth restricted graphs.**

I. INTRODUCTION

The study on routing in communication networks has a long history. Among all the routing algorithms, two most notable families are probably shortest path routing algorithms [11], [10] and the load balanced routing algorithms [13], [6], [9]. These two families can be regarded as to minimize two different quality measures: the *stretch factor*, defined to be the worst case ratio between the length of the path used by the algorithm and the length of the shortest path, and the *load balancing ratio*, defined to be the worst case ratio between the maximum load incurred by the algorithm and that of the optimal load balancing routing algorithm. Both a small stretch factor and a small load balancing ratio are desirable properties for routing. However, these two properties have been treated separately in the past. This probably should not be so surprising as they are conflicting goals to some extent: for a general graph or even a graph with constant degree, one can easily construct examples such that a shortest path routing algorithm necessarily creates heavily loaded nodes, and a load balancing routing algorithm necessarily uses very long paths.

In this paper, we study the trade-offs between those two measures for the family of *growth restricted* graphs. In our definition, an unweighted graph has density ρ and growth rate k if for every $r \geq 1$ and every node v in the graph, there are at most ρr^k nodes within distance (hop counts) r to v. Sometimes, we also say a graph has growth rate k if the density of the graph is bounded by a constant. Graphs with restricted growth rate arise in many practical networks, either due to physical constraints such as in wireless networks and VLSI layout networks, or due to geographical constraints such as in peer-to-peer overlay networks [35], [32], [33]. Routing in 'low dimensional' networks has been studied in the recent years [2], [18], [1], [22], [23], [24]. In this paper we show that for growth restricted graphs, there exists an interesting trade-off between the stretch factor and load balancing ratio with a dependency on the density and growth rate. We show that for a graph of n vertices with density ρ and growth rate k, the load balancing ratio is $O((n/c)^{1-1/k}\rho^{1/k})$ if the stretch factor of the routing paths is at most c. This bound is tight in the worst case.

One application of our result is in routing in wireless networks. A wireless network consists of nodes in the plane that communicate with each other by radio signal. Two nodes can have direct communication only if their distance is under certain threshold, say 1. In a typical wireless network such as a sensor network or an ad-hoc network, the nodes are energy constrained, and the major source of energy drain is from relaying packets. Therefore, it is crucial to balance the number of packets passed by each node. Indeed, there has been extensive work on energy-aware routing in wireless networks [19], [16], in addition to the usual study on the shortest path routing. The ideal algorithm would achieve good performance in terms of both the stretch factor and the load balancing ratio. In some special cases, for example, when the nodes are aligned on a line or in a narrow band [15], it is possible to design a routing algorithm achieving both a constant stretch factor and a constant load balancing ratio. This is unfortunately impossible in general — it is not difficult to construct a set of nodes and routing requests such that any routing algorithm limited to using paths with stretch factor c (named c-short paths) necessarily causes some node to relay $\Omega(n/c)$ packets while the optimal load-balancing algorithm only loads $O(1)$ packets on each node (an example is given in Section III). Such an example, however, uses highly crowded regions in the plane. In practical wireless networks, the nodes are distributed so that each node only has a small number of neighbors to reduce interference of wireless transmission and maximize coverage. When the density of the nodes, defined as the maximum number of nodes covered by any unit disk, is bounded by a constant, the communication graph of the wireless network has restricted growth rate $k = 2$. Our result implies that the load balancing ratio is $\Theta(\sqrt{n/c})$ in the worst case if we restrict the stretch factor to be at most c. For wireless networks, we extend the result to also consider the nodes with low *average density*, defined as the average number of nodes covered by the unit disks centered at the wireless nodes. The notion of average density is more appropriate for a set of

J. Gao is with Department of Computer Science, Stony Brook University, Stony Brook, NY 11794. E-mail: jgao@cs.sunysb.edu. Work was done when the author was at the HP Systems Research Center. L. Zhang is with Information Dynamics Laboratory, HP Labs, Palo Alto, CA 94304. E-mail: l.zhang@hp.com.

wireless nodes with uneven distribution. We can obtain weaker but similar bounds in term of average density.

Another application of our results is in global routing in VLSI design [35], [34]. In VLSI routing, given a graph (typically a mesh) which represents the physical wiring paths of a chip and a set of node pairs, one needs to connect every pair by a path in the graph. The goal is to minimize the line width on each node. The seminal work of Raghavan and Thompson [35] shows that by using the randomized rounding technique, one can approximate the optimal solution within a factor of $O(\log n/\log\log n)$. However, in [3], it is shown that if we only use a restricted set of paths, i.e. the paths that only make one bend, then the line width can be of an $\Omega(\sqrt{n})$ factor more than the optimal solution. Since mesh graphs have growth rate $k = 2$, our $O(\sqrt{n})$ upper bound implies that the construction in [3] is actually the worst possible, even compared to the routing schemes when non-shortest wires are allowed. Further, the bound holds as long as the underlying graph has growth rate 2 and constant density, which is usually the case in VLSI layout.

As shown in [32], the Internet network distance defined by round-trip propagation and transmission delay forms a metric with restricted growth rate. Therefore, our bound may also find application in routing in peer-to-peer overlay networks, such as file sharing networks and content addressable overlay networks.

In addition to the combinatorial bounds, we present an algorithm for achieving $O(\log n)$ competitive (and therefore approximation) ratio in terms of load balancing compared to the optimal c-short-path routing algorithm. Our algorithm is based on the on-line algorithm for virtual-circuit routing developed in [4], [5]. Another common approach of reducing routing complexity in wireless networks is to extract a sparse spanner graph [12] from the unit-disk graph and only route packets on the spanner [14], [28]. We also consider the load-balancing ratio of routing on spanner graphs compared to the optimal algorithm on the unit-disk graph and show a tight $\Theta(\rho c^2)$ competitive ratio when the stretch factor of the spanner graph is c.

There have been several definitions for capturing the graph or metric with smooth growth, such as the growth rate in [29], the expansion rate in [33], [20], and the doubling dimension in [17]. Our definition is almost identical to the one given in [29] except that we also take the density into consideration. The previous work has shown that the graph with smooth growth admits more efficient search algorithms, better embeddings, and more efficient routing algorithms compared to general graphs. Our work examines the load balancing on such graphs and adds another set of interesting properties to such graphs.

The paper is organized as follows. In Section II, we give some definitions and notations. We prove the main result on the trade-off for growth restricted graphs in Section III and then show the extension and application of our result in Section IV. We then present the algorithm for approximating the load balancing ratio with the stretch factor restriction (Section V).

II. Preliminaries

Suppose that $G = (V, E)$ is a connected unweighted graph. For every two nodes $p, q \in V$, denote by $\tau(p, q)$ the length of the shortest path between p and q. Let $B(p, r) = \{v \mid \tau(p, v) \leq r\}$ denote the radius r ball centered at p. We say G has *density* ρ and *growth rate* k if for every $p \in V$ and every $r \geq 1$, $|B(p, r)| \leq \rho r^k$.

We should note that our definition includes more graphs than several other definitions for capturing metrics with smooth growth. For example, in [20], a metric has *expansion rate* k if $|B(v, 2r)| \leq k|B(v, r)|$; and in [17], a metric has *doubling constant* k if $B(v, 2r)$ is contained in the union of at most k balls with radius r. If the shortest path metric of an unweighted graph has expansion rate k, the size of every ball with radius r is bounded by $O(k^{\log r}) = O(r^{\log k})$. Similarly, for the shortest path metric of an unweighted graph with doubling constant k, the size of every ball with radius r is bounded by $O(k^{\log r}) = O(r^{\log k})$. Therefore, in the graph setting, both definitions imply that the graph has a constant bounded growth rate. On the other hand, we can construct a family of graphs, e.g. the comb graphs as shown in Figure 1, with constant density and growth rate but unbounded expansion rate and doubling constant. Therefore, under our definition (and the one in [29]), more graphs are considered growth restricted.

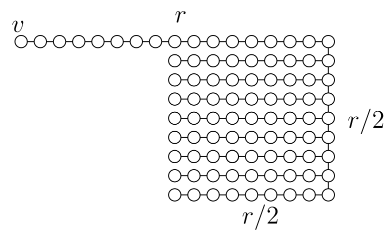


Fig. 1. The "comb" graph is a unit disk graph with constant bounded density. Therefore it is a graph with growth rate 2. Since $|B(v, 2r)| = \Theta(r^2)$ and $|B(v, r)| = \Theta(r)$, its expansion rate and doubling constant are both unbounded.
figure

Denote by $|P|$ the number of nodes on a path P in the graph G. For every two nodes $p, q \in V$, and a path P between p, q, the *stretch factor* $\omega(P)$ of P is defined to be $|P|/\tau(p, q)$. P is called *c-short* if $\omega(P) \leq c$.

A *routing request* is of the form $r = (s_r, t_r, \ell_r)$ where s_r, t_r, ℓ_r represent the source, destination, and the packet size, respectively. For a set of requests R, a set of paths $\mathcal{P}$ satisfy R, denoted by $\mathcal{P} \models R$, if $\mathcal{P} = \{P_r \mid r \in R\}$ where P_r is a path between s_r and t_r. We define the stretch factor $\omega(\mathcal{P})$ of $\mathcal{P}$ to be $\max_{r \in R} \omega(P_r)$. A routing algorithm is called a *c-short-path* (or c-short) routing if it only uses paths with stretch factor at most c. For example, shortest path routing algorithm is a 1-short path routing algorithm.

For a set of requests R and paths $\mathcal{P}$ that satisfy R, the *load* $\ell(v)$ on v is the total size of the packets that pass v, i.e. $\ell(v) = \sum_{v \in P_r} \ell_r$. The *load* $\ell(\mathcal{P})$ *of* $\mathcal{P}$ is then defined to be $\max_{v \in V} \ell(v)$. Define $\ell^*(R) = \min_{\mathcal{P} \models R} \ell(\mathcal{P})$ to be the optimal load for satisfying R and $\ell^c(R) = \min_{\mathcal{P} \models R \wedge \omega(\mathcal{P}) \leq c} \ell(\mathcal{P})$ the optimal load by any c-short-path routing algorithm. For example, $\ell^1(R)$ is the load created by a shortest path routing algorithm. For a routing algorithm $\mathcal{A}$, denote by $\mathcal{A}(R)$ the set of paths produced by $\mathcal{A}$ to satisfy R. Then $\mathcal{A}$'s approximation ratio (if $\mathcal{A}$ is off-line) or competitive ratio (if $\mathcal{A}$ is on-line) is defined to be $\max_R \frac{\ell(\mathcal{A}(R))}{\ell^*(R)}$. We generally call it the *load-balancing ratio*. In this paper, our goal is to study the trade-off between the stretch factor and the load-balancing ratio of routing algorithms in a network.

We now give definitions that are specific for wireless networks. Let S be a set of n nodes in the plane which represent wireless nodes. Let $|pq|$ denote the Euclidean distance between two nodes p, q. The *communication graph* of S can be modeled by an unweighted unit-disk graph $U(S) = (S, E)$, where $(p, q) \in E$ if and only if $|pq| \le 1$. A more practical model, named the quasi-unit disk graph, connects p, q if $|pq| \le \alpha$, for a parameter $\alpha < 1$ and does not connect them if $|pq| > 1$, and may or may not connect p, q otherwise [25]. For both models, we define the *maximum density* (or density[1] in short), $\rho(S)$ of S, as the maximum number of nodes in S covered by any unit disk. For each $p \in S$, denote by $\rho(p) = |B(p, 1)|$ with $B(p, 1)$ denote the unit size disk centered at node p. The density of p is thus the degree of node p in the unit disk graph plus 1. Define the *average density* $\bar{\rho}(S)$ of S to be $\sum_{p \in S} \rho(p)/n$, i.e., the average node degree plus 1. As easily seen, the graph $U(S)$ has density $\rho(S)$ and growth rate 2.

III. Trade-off for growth restricted graphs

For general graphs, it is only possible to obtain a weak trade-off between the stretch factor and the load-balancing ratio. The simple example in Figure 2 shows that if we insist on using c-short paths, then the load-balancing ratio can be as bad as $\Omega(n/c)$. There are $4c + 1$ spots on a loop. Each spot contains n/c nodes, except one spot has only one node o. The total number of nodes is $4n+1$. Only the nodes in adjacent spots are visible. If we make n/c requests, each from a distinct node on spot p to a distinct node on spot q. Any path that does not pass through o has length at least $4c$, i.e., is not c-short. Therefore, any c-short routing algorithm has to route the requests through o, i.e. o has load $\Theta(n/c)$. On the other hand, the optimal load balancing routing algorithm can route the requests evenly along the path on the longer arc such that each node only passes $O(1)$ packets.

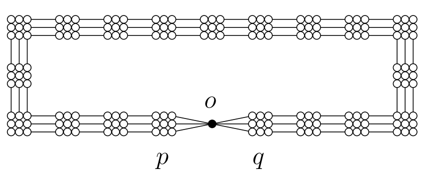


Fig. 2. Each spot contains n/c nodes. The loop has $4c+1$ spots. The packets from spot p to q either go through node o, thus causing the node o to be heavily loaded, or route along a long path with length $\Omega(n/c)$.
figure

The above graph has unbounded density or growth rate as the node o has $2n/c$ neighbors. For graphs with density ρ and growth rate k, we have that:

Theorem 1. *For a graph with density ρ and growth rate k, the load-balancing ratio of the optimal c-short routing is $O(\min((n/c)^{1-1/k}\rho^{1/k}, n/c)$. In particular, the load-balancing ratio of (any) shortest path routing for a graph with constant density and growth rate k is $O(n^{1-1/k})$.*

For notational simplicity, we will present the proof for $k = 2$. The extension to general k is straight forward. When $k = 2$, we will show that for any set of requests R, $\ell^c(R)/\ell^*(R) = O(\min(\sqrt{\rho n/c}, n/c))$, and the bound is tight in the worst case. As a special case, when the density is bounded by a constant the load-balancing ratio of the optimal c-short path routing is bounded by $O(\sqrt{n/c})$. In another special case, when $c = 1$, the load-balancing ratio for shortest path routing is $O(\sqrt{\rho n})$. So shortest path routing on nodes with constant density achieves a load balancing ratio of $O(\sqrt{n})$. We first prove the case of shortest path routing (Theorem 2) and extend the technique to prove Theorem 1.

Theorem 2. *For any n nodes with maximum density ρ and any set of requests R, $\ell^1(R)/\ell^*(R) = O(\sqrt{\rho n})$.*

Proof: Suppose that p is the node with the maximum load if we use shortest path routing. Without loss of generality, we can assume that all the requests in R are routed through p by shortest path routing, because otherwise we can safely delete those requests that do not — this does not change the maximum load by shortest path routing[2] but can only decrease the maximum load of the optimal load. Suppose that the set of requests is $R = \{r_1, \ldots, r_m\}$ where $r_i = (s_i, t_i, \ell_i)$ is a request from s_i to t_i with packet size ℓ_i. We denote by ℓ^* the maximum load of the optimal load balanced routing algorithm $\ell^*(R)$. Since all the requests in R pass through p in shortest path routing scheme, the maximum load of shortest path routing, $\ell^1(R) = \ell = \sum_{i=1}^{m} \ell_i$. We now wish to upper-bound $\alpha = \ell/\ell^*$.

The intuition of the proof is that shortest path routing is optimal in the sense of the *total loads* it creates. If the load on p is high, the total load a shortest path routing creates is also necessarily high. This causes the optimal algorithm to create high total loads as well. The average load therefore cannot be too low, even if those loads can be evenly distributed. This intuition is made concrete by the following lemma.

We first give some notations. For each node $q \in V$, denote by $R(q)$ all the requests that originate at q and by $\tilde{\ell}(q)$ the total size of those packets, i.e. $\tilde{\ell}(q) = \sum_{r_i \in R(q)} \ell_i$. Define $\beta(q) = \tilde{\ell}(q)/\ell$, where $\ell = \sum_{i=1}^{m} \ell_i$. Clearly $\sum_q \beta(q) = 1$.

Lemma 3. *Suppose that D_τ is the disk with radius $\tau \ge 1$ centered at p, then $\sum_{q \in D_\tau} \beta(q) \le c_0 \rho \tau / \alpha$, for some constant $c_0 > 0$.*

Proof: We partition D_τ into a set of $\log \tau$ disjoint sets B_k, $0 \le k \le \log \tau$, where B_0 is the unit disk centered at p and for $k \ge 1$, B_k is an annulus[3] with an inner radius of 2^{k-1} and an outer radius of 2^k. See Figure 3. Consider the set R_k of the requests originating at some node in B_k and a request $r_j = (s_j, t_j, \ell_j) \in R_k$. Since the shortest path between s_j and t_j passes the node p, the length of the shortest path between s_j and t_j is at least the shortest path length between p and s_j, i.e., $\tau(s_j, t_j) \ge \tau(p, s_j) \ge |ps_j| \ge 2^{k-1}$. Now, suppose that P_j is the path from s_j to t_j produced by the optimal load-balanced routing algorithm. The number of nodes on P_j is at least 2^{k-1}. Let A_j be the first 2^{k-1} nodes on P_j. Denote by S_k the union of all the A_j, i.e., $S_k = \bigcup_{r_j \in R_k} A_j$. We study the total load produced by the optimal load balanced routing algorithm on the nodes inside S_k. We denote by $\ell_{opt}(v)$ the load on node v by the optimal load balanced routing algorithm. $\ell_{opt}(v) \le \ell^*$. Firstly we have

$$\sum_{v \in S_k} \ell_{opt}(v) \ge \sum_{r_j \in R_k} \ell_j |A_j| = 2^{k-1} \sum_{r_j \in R_k} \ell_j \,. \tag{1}$$

[1]The notation density is slightly abused. As we shall see, the two definitions of density coincide.

[2]When there are multiple shortest paths between two nodes, with shortest path routing the nodes can have different load distributions. Here in this argument we compare the 'most unbalanced' load distribution by shortest path routing against the optimal load balanced routing. This is all right as this theorem concerns about upper-bounding the worst case load of shortest path routing.

[3]The inner boundary is included in B_k and the outer boundary is not.

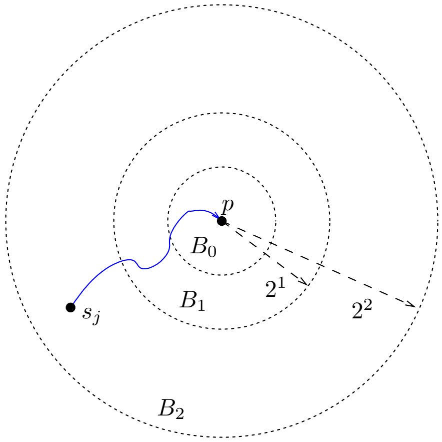


Fig. 3. Division of D_τ into a set of disjoint sets B_i. All the traffic pass through the center p by shortest path routing.
figure

On the other hand, for every node $a \in A_j$, $|pa| \le |ps_j| + |as_j| \le 2^k + 2^{k-1} = 3 \cdot 2^{k-1}$. That is, all the nodes in A_j are inside a disk with radius $3 \cdot 2^{k-1}$ centered at p. Since the nodes have maximum density ρ, $|S_k| = O(\rho(3 \cdot 2^{k-1})^2)$. That is, there is a constant $c_0 > 0$ such that $|S_k| = c_0\rho(3 \cdot 2^{k-1})^2$. Since each node has load at most $\ell^* = \ell/\alpha$, we have that

$$\sum_{v \in S_k} \ell_{opt}(v) \le |S_k|\ell^* \le c_0\rho(2^{k-1})^2\ell/\alpha\,. \quad (2)$$

Combining (1) and (2), we have that

$$\sum_{r_j \in R_k} \ell_j \le c_0\rho 2^{k-1}\ell/\alpha\,.$$

Thus $\sum_{r_j \in R_k} \beta_j = \sum_{r_j \in R_k} \ell_j/\ell \le c_0\rho 2^{k-1}/\alpha$, for $1 \le k \le \log\tau$. For the unit disk B_0, we have that

$$\begin{aligned}\textstyle\sum_{q \in B_0} \beta(q) &= \textstyle\sum_{q \in B_0} \tilde{\ell}(q)/\ell \le |B_0|\ell^*/\ell \\ &\le \rho\ell^*/\ell = \rho/\alpha\,.\end{aligned}$$

By summing up over all the k's, we have that

$$\begin{aligned}\textstyle\sum_{q \in D_\tau} \beta(q) &= \textstyle\sum_{q \in B_0} \beta(q) + \sum_{k=1}^{\log\tau} \sum_{r_j \in R_k} \beta_j \\ &\le \rho/\alpha + \textstyle\sum_{k=1}^{\log\tau} c_0\rho 2^{k-1}/\alpha \\ &\le c_0\rho\tau/\alpha\,.\end{aligned}$$

The last inequality follows from the sum of a geometric sequence. □

Now we proceed to prove Theorem 2. We can assume that for every $q \in V$, $\beta(q) \le 1/3$; otherwise $\ell^* \ge \tilde{\ell}(q) > \ell/3$, i.e. $\alpha < 3$. Now, consider the smallest disk D centered at p such that

$$\sum_{q \in D} \beta(q) \ge 1/2\,.$$

We assume that there is only one node on the boundary of D — otherwise we can perturb (conceptually) the nodes so that the assumption is valid. Since $\beta(q) \le 1/3$ for every q, we have that

$$\sum_{q \notin D} \beta(q) \ge 1/6\,.$$

Let τ^* denote the radius of D. Then, by Lemma 3,

$$c_0\rho\tau^*/\alpha \ge \sum_{q \in D} \beta(q) \ge 1/2\,,$$

i.e.

$$\alpha \le 2c_0\rho\tau^*\,. \quad (3)$$

On the other hand, for every node $q \notin D$, $|pq| \ge \tau^*$. By the same argument used in the proof of Lemma 3, for any algorithm, the loads incurred by those requests originating at q are at least $\tilde{\ell}(q)\tau^*$. Therefore, the total loads caused by such requests are at least

$$\sum_{q \notin D} \tilde{\ell}(q)\tau^* = \sum_{q \notin D} \beta(q)\ell\tau^* \ge \ell\tau^*/6\,.$$

Hence, the optimal load balancing routing algorithm can do no better than distributing these loads evenly on the n nodes. That is, $\ell^* \ge \ell\tau^*/6n$, i.e.

$$\alpha = \ell/\ell^* \le 6n/\tau^*\,. \quad (4)$$

By combining (3) and (4), we have that

$$\alpha \le \min(2c_0\rho\tau^*, 6n/\tau^*) \le c_1\sqrt{\rho n}\,,$$

for $c_1 = \sqrt{12c_0}$. This proves Theorem 2. □

Now, we extend the result to c-short routing.

PROOF OF THEOREM 1. We show that, for any set of requests R, we can construct a set of c-short paths that achieve the claimed upper bound. Consider the optimal routing that minimizes the maximum load, regardless of path length. We divide R into two subsets R_1 and R_2, where R_1 contains the requests that are routed by c-short paths in the optimal algorithm, and R_2 contains those requests routed by non-c-short paths. We construct a set of paths $\mathcal{P}$ as follows. We include in $\mathcal{P}$ the paths that the optimum algorithm produced for requests in R_1. For each request in R_2, we add to $\mathcal{P}$ (any) shortest path between the source and the destination of that request. Clearly, all the paths in $\mathcal{P}$ are c-short. We now show that the maximum load caused by $\mathcal{P}$, denoted by $\ell(\mathcal{P})$, is at most $O(\min(\sqrt{\rho n/c}, n/c)\ell^*(R))$.

For each node $q \in V$, denote by $\ell_1^*(q), \ell_2^*(q)$, the loads on q caused by, respectively, routing R_1 and R_2 by the optimal algorithm. Let $\ell_1^* = \max_q \ell_1^*(q)$ and $\ell_2^* = \max_q \ell_2^*(q)$. Clearly, $\ell^* \ge \max(\ell_1^*, \ell_2^*) \ge (\ell_1^* + \ell_2^*)/2$. For each node $q \in V$, denote by $\ell_2(q)$ the loads on q caused by routing R_2 by using shortest path routing. Let $\ell_2(R) = \max_q \ell_2(q)$. Clearly, $\ell^c(R) \le \ell(\mathcal{P}) \le \ell_1^* + \ell_2(R)$.

We now bound $\ell_2(R)/\ell_2^*$ by using almost the same argument as in the proof of Theorem 2. The only difference is that all the paths used to route requests in R_2 by the optimal algorithm are not c-short. Therefore, all the requests originating at nodes outside the disk D generate a total load of at least $\sum_{q \notin D} \tilde{\ell}(q) \cdot c\tau^*$, which is equal or greater than $\ell c\tau^*/6$. Then we can replace (4) with the following inequality

$$\ell_2(R)/\ell_2^* \le 6n/(c\tau^*)\,.$$

Since (3) is still valid, we have that

$$\begin{aligned}\ell_2(R)/\ell_2^* &= \min(2c_0\rho\tau^*, 6n/(c\tau^*)) \\ &= O(\min(\sqrt{\rho n/c}, n/c))\,.\end{aligned}$$

Therefore,

$$\begin{aligned}\ell^c(R) &\le \ell_1^* + \ell_2(R) = O(\min(\sqrt{\rho n/c}, n/c))(\ell_1^* + \ell_2^*) \\ &= O(\min(\sqrt{\rho n/c}, n/c)) \cdot \ell^*\,.\end{aligned}$$

This proves the upper bound in Theorem 1.

In the following, we show a lower bound construction. We only describe the lower bound construction for $\rho c \le n$, i.e. $\sqrt{\rho n/c} \le$

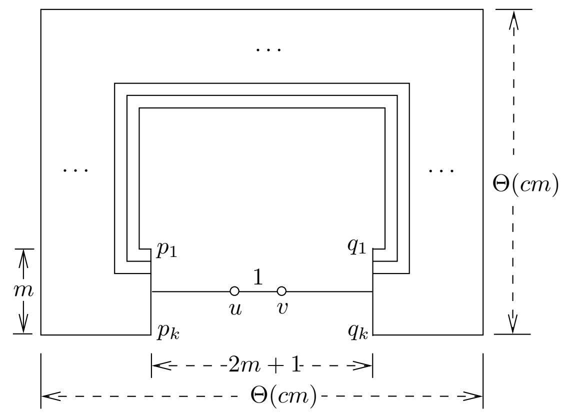


Fig. 4. Lower bound of the load-balancing ratio for the optimal c-short routing with density ρ.
figure

n/c. The other case is similar. We actually give a construction based on the wireless communication graph of a set of nodes in the plane. Consider the example illustrated in Figure 4. The distance between u, v is 1. Take a parameter $m > 1$ which will be determined later, we place $k = \rho m$ nodes $p_1, \ldots, p_k$ on a vertical line segment with length m and distance m away from u. Similarly, we create $q_1, \ldots, q_k$ with respect to v. On the horizontal line segment through u, v, we place about $2m$ nodes evenly. In addition, there is a path between every pair of p_i and q_i as drawn in Figure 4. Each path is about $4cm$ long and has $4cm$ nodes on it. Clearly, the maximum density of the node set is $O(\rho)$. The shortest path between p_i and q_i goes through u, v and has length at most $3m$. On the other hand, any other path connecting u, v has to go through the outside loop with length $4cm$. So all the c-short paths connecting p_i, q_i have to pass u and v. Therefore, if we request to send a unit packet from p_i to q_i, for $1 \leq i \leq k$, then the c-short path routing causes load $k = \rho m$ on u, v. On the other hand, we can use the outer path to route each packet, creating load 1 on each node. Thus, the load-balancing ratio of any c-short path routing of this example is $\Omega(\rho m)$. The total number of nodes in the example is about $\Theta((\rho m) \cdot (cm)) = \Theta(\rho c m^2)$. Setting $m = \sqrt{n/(c\rho)}$, we obtain the desired lower bound for $k = 2$. The extension to general k is straightforward. □

We should emphasize that in the proof of Theorem 2, we do not restrict which shortest path to use when there are more than one shortest paths. That is, the bound holds no matter which shortest paths are used when there exist multiple shortest paths. However, the proof of Theorem 1 does use a set of c-short paths produced by the optimal algorithm. Therefore, the bound does not hold for arbitrary c-short paths. Actually, if we choose bad c-short paths, we may end up with a bound even worse than that of the shortest path routing. In section V, we will present an algorithm to discover a set of c-short paths with the maximum load within $O(\log n)$ factor of the optimum load using c-short paths.

IV. Applications and extensions

A. *Trade-off based on average density for wireless networks*

The previous result immediately applies to the wireless network with maximum density ρ as the communication graph of such a network has density ρ and growth rate 2. In this section, we will show that we can obtain a weaker but similar bound in terms of average density for wireless networks. The benefit of considering average density is clear — it is applicable to a wider family of distributions including some uneven distributions.

Corollary 4. *Given a set of n nodes V in the plane with average density $\bar{\rho}$, for any set of requests R,*

$$\ell^1(R)/\ell^* = O(\min(\sqrt{\bar{\rho} n} \log n, n))\,.$$

In addition, there exists example such that

$$\ell^c(R)/\ell^* = \Omega(\sqrt{\bar{\rho} n / \max(1, \log c)})\,.$$

Proof: The proof for the upper bound is similar to the proof of Theorem 2. We use the notation in the proof of Lemma 3. We take τ to be the diameter of the communication graph. Since the graph is connected, $\tau = O(n)$. So D_τ, a disk with radius τ centered at p, covers all the n nodes. Then we partition D_τ into a set of $\log \tau$ disjoint sets B_k, $0 \leq k \leq \log \tau$, where B_0 is the unit disk centered at p and for $k \geq 1$, B_k is an annulus with an inner radius of 2^{k-1} and an outer radius of 2^k. The only difference is that with the average density, we can no longer bound the size of S_k as in Lemma 3. Instead, by using average density, we can obtain a weaker bound by the following fact.

Lemma 5. *For any planar point set V and disk B with radius $r \geq 1$, $|B \cap V| = O(r\sqrt{n\bar{\rho}(V)})$.*

Proof: We can partition $B \cap V$ into $O(r^2)$ disjoint subsets such that all the points in one subset are within distance 1 from each other. Suppose that those sets are $S_1, \ldots, S_m$, $B \cap V = \cup_{i=1}^m S_i$. Therefore, $\sum_i |S_i|^2 \leq n\bar{\rho}(V)$. By the Cauchy-Schwartz inequality, we have that $|B \cap V|^2 = (\sum_i |S_i|)^2 \leq m(\sum_i |S_i|^2) \leq mn\bar{\rho}(V)$. Since $m = O(r^2)$, $|B \cap V| = O(r\sqrt{n\bar{\rho}(V)})$. □

By the above lemma, for $1 \leq k \leq \log \tau$,

$$|S_k| = |\cup_{r_j \in R_k} A_j| = O(\sqrt{n\bar{\rho}} 2^k)\,.$$

Since each node has load at most ℓ^*, we have that

$$2^{k-1} \sum_{r_j \in R_k} \ell_j \leq c_0 \sqrt{\bar{\rho} n} 2^{k-1} \ell^*\,,$$

for some constant $c_0 > 0$. Thus $\sum_{r_j \in R_k} \ell_j \leq c_0 \sqrt{\bar{\rho} n} \ell^*$, for $1 \leq k \leq \log \tau$. We also know that $\sum_{r_j \in R_0} \ell_j \leq \bar{\rho} \ell^* \leq \sqrt{\bar{\rho} n} \ell^*$, since $\bar{\rho} \leq n$. By summing up for all the k's, we have that

$$\ell^1(R) = \ell = \sum_{k=0}^{\log \tau} \sum_{r_j \in R_k} \ell_j \leq c_1 \sqrt{\bar{\rho} n} \ell^* \log n\,,$$

for some constant c_1.

As for the lower bound, consider the example shown in Figure 5. In the figure, the distance between u, v is 1. There are c vertical bars with length $1, 2, \ldots, c$ and with distance $0.5, 1.0, 1.5, \ldots$ away from u. We place k nodes on each of the line segments evenly with k determined later. Symmetrically, we place nodes with respect to the node v. Label those nodes on the outside bars $p_1, \ldots, p_k$ and $q_1, \ldots, q_k$, respectively, and those nodes on the bar closest to u, v to be $u_1, \ldots, u_k$, and $v_1, \ldots, v_k$, respectively. Again, we place nodes to connect every pair p_i, q_i as shown in the figure. The length of those paths is $\Theta(c)$. Now, we request to send a packet from p_i to q_i, for $1 \leq i \leq k$. Again, each c-short path routing has to use the path through the nodes u, v, causing a load of k on u, v. On the other hand, the optimal algorithm can route the requests through the outside paths and create only load 1 to each node. Thus, the load-balancing ratio of any c-short routing algorithm is $\Omega(k)$. The total number of nodes

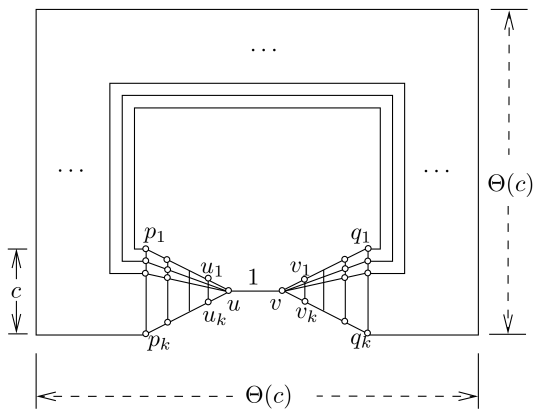


Fig. 5. Lower bound of the load-balancing ratio for the optimal c-short routing with average density $\bar{\rho}$.
figure

in the figure is bounded by $O(c \cdot k)$. To bound the average density of the nodes, we consider two types of nodes. For a node x on a vertical bar with length h, the number of nodes it sees is about $\Theta(k/h)$. Thus,

$$\sum_x \rho(x) = \Theta(\sum_{h=1}^{c} k^2/h) = \Theta(\max(1, \log c) \cdot k^2) .$$

For a node y on the outside path, $\rho(y) = \Theta(k/c)$. Therefore, the average density $\bar{\rho}$ is

$$\Theta((\max(1, \log c) \cdot k^2 + ck(k/c))/n) = \Theta(\max(1, \log c) \cdot k^2/n) .$$

That is, $k = \Theta(\sqrt{\bar{\rho}n/\max(1, \log c)})$, and the load-balancing ratio is

$$\Omega(\sqrt{\bar{\rho}n/\max(1, \log c)}) .$$

□

B. VLSI routing

In VLSI routing, the task is to connect some given pairs of nodes by paths on a mesh. One important goal is to reduce the line width, i.e. the maximum number of paths that pass the same edge. Such a problem has been studied extensively [35], [36], [27], [7]. A mesh can be realized as a unit disk graph of a set of nodes with constant bounded density. Thus, we have the following extension of our result to bound the line width in VLSI routing.

Corollary 6. *If we are restricted to use c-short paths to route wires in a mesh, then the line width is within $O(\sqrt{n/c})$ factor of the optimum solution. In particular, if we use (any) shortest paths, the approximation factor is $O(\sqrt{n})$.*

V. An algorithm for short path load balancing routing

In the previous section, we showed a combinatorial bound on the load balancing ratio for the optimal c-short routing algorithm. However, it is NP-hard to compute the set of c-short paths (actually even the shortest paths) that minimizes the maximum load [21]. Here, we describe an algorithm that computes c-short paths with maximum load within an $O(\log n)$ factor of the optimum.

One general approach for computing routes with a good load balancing ratio is to use the randomized rounding technique [35], [34]. But that technique cannot be directly applied to our case because of the restriction on the stretch factor — the size of the linear programming problem is exponential in the size of the network. Here we apply the on-line virtual circuit routing algorithm by Aspnes *et al.* [4] to our problem and obtain an $O(\log n)$ approximation ratio.

Theorem 7. *There is a polynomial time on-line c-short routing algorithm with load balancing competitive ratio $O(\log n)$ when compared to the optimal off-line c-short routing algorithm. The competitive ratio is tight in the worst case.*

Proof: We assume that the routing requests come in an on-line fashion. Upon the receipt of each routing request, one has to design a route which is no longer than c times the shortest path length. The goal is to minimize the maximum load on any node. We apply the method in [4] with slight modification. In the algorithm in [4], a weight is assigned to each edge (or vertex/node in our case) according to the current load on the edge and the size of the request. Then for every new request, the lightest path[4] with respect to this weighting function is used to satisfy the request. Similarly, for c-short routing, we use the lightest path among all the c-short paths. We just need to show that this modification can be done in polynomial time, and it does find us an $O(\log n)$ approximation.

To see the former, we can use dynamic programming: given a pair of nodes (s, t), we iteratively compute, for every node u in the graph, the lightest path from s to u with length exactly L (this may include non-simple paths) for $L = 1, 2, \ldots, c \cdot \tau(s, t)$, where $\tau(s, t)$ denotes the shortest distance between s, t. This will give us the lightest c-short path connecting s and t in polynomial time.

The proof of the $O(\log n)$ competitive ratio follows from the argument in the proof of Theorem 5.2 in [4]. By a close examination of that proof, we can see that it still holds even if we associate each request r with a subset of paths P_r such that only a path in P_r can be used to satisfy r. Therefore, restricting all the paths to be c-short is just a special case.

For a lower bound on the competitive ratio, we can show that even in a mesh, any on-line c-short routing algorithm is $\Omega(\log n)$ competitive compared to the optimal c-short routing algorithm. The lower bound construction is very similar with a construction in [8].

We assume a set of n nodes on a grid of size $\sqrt{n} \times \sqrt{n}$. The adversary makes $\log n$ rounds of requests. Each request has size 1. Divide the square uniformly into 16 sub-squares. The first set of requests are from the nodes on the line segment a_1b_1 to the corresponding nodes on c_1d_1. It is clear that any routing algorithm must use the nodes in the 8 sub-squares in the middle 2 rows. The adversary chooses one of 8 such sub-squares, say S_2, and recurses.

For the optimal routing, according to which sub-square is selected, we can route the first round requests by using nodes outside the sub-square S_2, as shown by Figure 6. Therefore, the nodes carrying the paths for requests in the first round do not overlap with the nodes carrying the paths for the later rounds. Thus every node has load at most 1. On the other hand, the average load on the sub-square S_2 is at least $1/2$. To make the online algorithm with a heavy maximum load, an adversary can

[4]we call it the lightest path, to be distinguished from the shortest path in the graph.

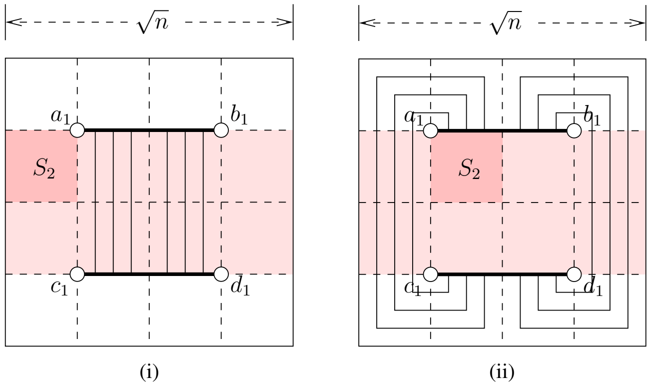


Fig. 6. A lower bound $\Omega(\log n)$ on the competitive ratio of any online algorithm compared with the optimal off-line algorithm with c-short paths. The figure shows two ways of routing packets from a_1b_1 to c_1d_1. If the adversary choose a subsquare S_2 as one of the 4 pink subsquares on the perimeter, as shown in (i), we can route the requests from a_1b_1 to c_1d_1 to avoid S_2. Similarly, if the adversary choose a subsquare S_2 as one of the 4 pink subsquares at the center, as shown in (ii), we can route the requests from a_1b_1 to c_1d_1 to avoid S_2.
figure

always select the sub-square S_2 to be the one that already carries average load at least $1/2$ for the requests in the first round. After $\log_2 n/4$ rounds, the node in the sub-square $S_{\log_2 n/4}$ has average load at least $\log_2 n/8$, i.e., one node will have load $\Omega(\log n)$ while the optimal routing can always spread out the load. □

The above algorithm only discovers the paths but does not deal with the scheduling in the routing, e.g., the queueing principle when multiple packets need to be delivered from the same node at the same time, or the interference resolution when multiple nearby nodes transmit packets. The methods in [26] or [31] can be used for such scheduling after the path selection step.

VI. Load-balancing ratio of routing on spanners

One method to reduce the complexity of routing in wireless networks is to construct a sparse spanner graph and route on the spanner graph [14], [28]. A sub-graph G of a unit-disk graph $U(V)$ is a c-spanner if the shortest path between any two nodes in G is c-short compared with $U(V)$. Since a spanner graph has fewer edges than the unit-disk graph, the load balancing ratio on a spanner graph might be high. The following theorem provides a worst case tight bound.

Theorem 8. *Suppose V is a set of n nodes in the plane with density ρ, and G is a c-spanner of $U(V)$, for any requests R, $\ell^*_G(R)/\ell^* = O(\rho c^2)$, where $\ell^*_G(R)$ (ℓ^*) is the maximum load resulted by the optimal load-balancing routing algorithm on G ($U(V)$). The bound is tight in the worst case.*

Proof: For a set of requests R, consider the optimal solution $\mathcal{P}^*$ on the unit-disk graph U. We now construct a solution on G from $\mathcal{P}^*$. For an edge uv on a path in $\mathcal{P}^*$, if it is not in G, then there must exist a path with length c in G because G is a c-spanner. We can then reroute the packet on that path. Clearly, this way we obtain a set of paths $\mathcal{P}'$ in G that satisfy R. Now, consider a node $p \in V$. A packet can be redirected to it only if it is routed in the optimal solution through a node u which is at most distance c away from p. Or, u is in the disk with radius c and centered at p. There are $O(\rho c^2)$ such nodes. Therefore, the load on p is $O(\rho c^2 \ell^*)$.

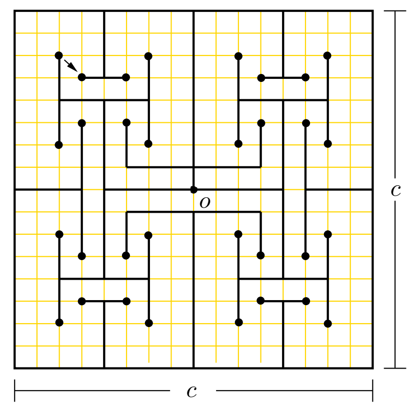


Fig. 7. Lower bound $\Omega(c^2)$ on the competitive ratio of load balanced routing algorithms on c-spanners.
figure

As for the lower bound, we use the classic H-tree construction [30]. We only show the construction for nodes with constant density. The extension to nodes with density ρ is easy – we just put ρ copies on each grid node. Consider $\Theta(c^2)$ nodes positioned on a grid as shown in Figure 7. Each little square of the grid has side length $1/2$. The spanner G is composed of an H-tree and a "complement" skeleton joined by a single edge at the center of the grid o. So any path from a node on the H-tree to a node in the complement H-tree has to go through o. Clearly, G is a $\Theta(c)$-spanner graph. Now we make a request from each leaf node of the H-tree to its nearby node on the complement part of the H-tree (see little arrow in Figure 7). The optimal solution can send the requests directly. However, in G, all the requests have to be routed through the node o. Therefore, the load-balancing ratio of the routing on this c-spanner is $\Omega(c^2)$. □

VII. Conclusion

In this paper, we study the trade-off between two important quality measures of routing algorithms for growth-restricted networks: the stretch factor for measuring the path length and the load balancing ratio for measuring the load balance. We show several trade-offs based on the density and growth rate of a graph. We extend the result to wireless networks by considering average density of the wireless nodes. There is still a gap for the trade-off when considering average density. Besides, all of our results are based on the worst case analysis. It would be interesting to study the trade-off under some reasonable traffic model.

References

[1] I. Abraham, C. Gavoille, A. V. Goldberg, and D. Malkhi. Routing in networks with low doubling dimension. In *Proc. IEEE International Conference on Distributed Computing Systems*, page 75, 2006.

[2] I. Abraham and D. Malkhi. Name independent routing for growth bounded networks. In *Proc. ACM symposium on Parallelism in algorithms and architectures*, pages 49–55, 2005.

[3] M. Andrews, D. Bhatia, T. Leighton, F. Makedon, C. H. Norton, and L. Zhang. Improved algorithms for routing on two-dimensional grids. Unpublished manuscript.

[4] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line machine scheduling with applications to load balancing and virtual circuit routing. In *Proc. ACM Symposium on Theory of Computing*, pages 623–631, 1993.

[5] B. Awerbuch, Y. Azar, and S. A. Plotkin. Throughput-competitive on-line routing. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 32–40, 1993.

[6] Y. Azar. On-line load balancing. In A. Fiat and G. Woeginger, editors, *On-line Algorithms: The State of the Art*, pages 178–195. LNCS 1442, Springer, 1998.

[7] H. Badr and S. Podar. An optimal shortest-path routing policy for network computers with regular mesh-connected topologies. *IEEE Transactions on Computers*, 38(10):1362–1371, 1989.

[8] Y. Bartal and S. Leonardi. On-line routing in all-optical networks. *Theoretical Computer Science*, 221(1–2):19–39, 1999.

[9] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[10] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1994.

[11] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[12] D. Eppstein. Spanning trees and spanners. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*. North-Holland, 2000.

[13] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5:691–703, 1976.

[14] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanners for routing in mobile networks. *IEEE Journal on Selected Areas in Communications Special issue on Wireless Ad Hoc Networks*, 23(1):174–185, 2005.

[15] J. Gao and L. Zhang. Load balanced short path routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communications*, 17(4):377–388, April 2006.

[16] A. Goldsmith and S. Wicker, editors. *Special Issue: Energy-aware ad hoc wireless networks*, volume 9. Proc. IEEE Wireless Comm., Auguest 2002.

[17] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543, 2003.

[18] K. Hildrum, R. Krauthgamer, and J. Kubiatowicz. Object location in realistic networks. In *Proc. ACM symposium on Parallelism in algorithms and architectures*, pages 25–35, 2004.

[19] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J.-C. Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks*, 7(4):343–358, 2001.

[20] D. Karger and M. Ruhl. Find nearest neighbors in growth-restricted metrics. In *Proc. ACM Symposium on Theory of Computing*, pages 741–750, 2002.

[21] J. Kleinberg. *Approximation Algorithms for Disjoint Paths Problems*. PhD thesis, Dept. of EECS, MIT, 1996.

[22] G. Konjevod, A. W. Richa, and D. Xia. Optimal-stretch name-independent compact routing in doubling metrics. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 198–207, 2006.

[23] G. Konjevod, A. W. Richa, D. Xia, and H. Yu. Compact routing with slack in low doubling dimension. In *PODC '07: Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing*, pages 71–80, 2007.

[24] G. Konjevod, A. W. Richa, D. Xia, and H. Yu. Optimal scale-free compact routing schemes in networks of low doubling dimension. In *SODA'07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 939–948, 2007.

[25] F. Kuhn and A. Zollinger. Ad-hoc networks beyond unit disk graphs. In *Proc. Joint Workshop on Foundations of Mobile Computing*, pages 69–78, 2003.

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

[27] T. Leighton. Methods for message routing in parallel machines. *Theoretical Computer Science*, 128(1–2):31–62, 1994.

[28] X.-Y. Li, G. Calinescu, and P.-J. Wan. Distributed construction of planar spanner and routing for ad hoc networks. In *Proc. IEEE INFOCOM*, pages 1268 – 1277, 2002.

[29] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15:215–245, 1995.

[30] C. Mead and L. Conway. *Introduction to VLSI systems*. Addison-Wesley, 1980.

[31] F. Meyer auf de Heide, C. Schindelhauer, K. Volbert, and M. Grünewald. Energy, congestion and dilation in radio networks. In *Proc. ACM Symposium on Parallel Algorithms and Architectures*, pages 230–237, 2002.

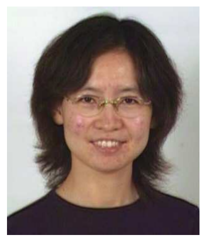
[32] E. Ng and H. Zhang. Predicting Internet network distance with coordinates-based approaches. In *Proc. IEEE INFOCOM*, pages 170–179, 2002.

[33] C. G. Plaxton, R. Rajaraman, and A. W. Richa. Accessing nearby copies of replicated objects in a distributed environment. In *Proc. ACM Symposium on Parallel Algorithms and Architectures*, pages 311–320, 1997.

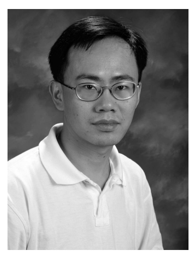
[34] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer programs. *J. Comp. and System Sciences*, pages 130–143, 1988.

[35] P. Raghavan and C. D. Thompson. Provably good routing in graphs: regular arrays. In *Proc. ACM Symposium on Theory of Computing*, pages 79–87, 1985.

[36] T.-H. Yeh, C.-M. Kuo, C.-L. Lei, and H.-C. Yen. Competitive source routing on tori and meshes. In *Proc. International Symposium on Algorithms and Computation (ISAAC)*, pages 82–91, 1997.



Jie Gao received her Ph.D in computer science from Stanford University in 2004, and her BS degree from University of Science and Technology of China in 1999. She is currently a postdoc fellow at the center for the mathematics of information at California Institute of Technology, and will join the State University of New York, Stony Brook as an assistant professor in Fall 2005. Her research interests are algorithms, ad hoc communication and sensor networks, and computational geometry.



Li Zhang received his BE and ME degrees, both in Computer Science, from Tsinghua University, Beijing, China, in 1991 and 1993, respectively, and the PhD degree in Computer Science from Stanford University, California, in 2000. He is currently a researcher at Information Dynamics Laboratory in HP Labs, Palo Alto, California. His research interests include computational geometry, algorithms for ad-hoc and sensor networks, and game theoretical analysis of resource allocation.