Tradeoffs between Stretch Factor and Load Balancing Ratio in Routing on Growth Restricted Graphs

Jie Gao* Li Zhang†

ABSTRACT

A graph has growth rate k if the number of nodes in any subgraph with diameter r is bounded by $O(r^k)$. The communication graphs of wireless networks and peer-to-peer networks often have small growth rate. In this paper we study the tradeoff 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 how evenly the traffic is distributed. We show that if the routing algorithm is required to use paths with stretch factor c, then its load balancing ratio is bounded by $O((n/c)^{1-1/k})$, where k is the graph's growth rate. We illustrate our results by focusing on the unit disk graph for modeling wireless networks in which two nodes have direct communication if their distance is under certain threshold. We show that if the maximum density of the nodes is bounded by ρ, there exists routing scheme such that the stretch factor of routing paths is at most c, and the maximum load on the nodes is at most $O(\min(\sqrt{\rho n/c}, n/c))$ times the optimum. In addition, the bound on the load balancing ratio is tight in the worst case. As a special case, when the density is bounded by a constant, the shortest path routing has a load balancing ratio of $O(\sqrt{n})$. The result extends to k-dimensional unit ball graphs and graphs with growth rate k. We also discuss algorithmic issues for load balanced short path routing and for load balanced routing in spanner graphs.

Categories and Subject Descriptors: E.1 [Data]: Data Structures—*graphs and networks*; F.2.2 [Theory of Computation]: analysis of algorithms and problem complexity—*non-numerical algorithms and problems*; F.2.3 [Theory of Computation]: analysis of algorithms and problem complexity—*Tradeoffs between Complexity Measures*

General Terms: Theory, Algorithms

*Department of Computer Science, Stanford University, Stanford, CA 94305. E-mail: jgao@cs.stanford.edu. Work was done when she was working at the HP Systems Research Center.

†Information Dynamics Laboratory, HP Labs, Palo Alto, CA 94304. E-mail: l.zhang@hp.com.

PODC'04, July 25–28, 2004, St. Johns, Newfoundland, Canada.

Keywords: Routing, load balancing, wireless networks, growth restricted graphs

1 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 [9] and the load balanced routing algorithms [5, 8]. These two families can be regarded as to minimize two quality measures: the stretch factor of the paths, 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 studied separately in the past. This probably should not be so surprising as they are conflicting goals to some extent: for a general graph, 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 long paths.

In this paper, we study the tradeoffs between those two measures for the family of *growth restricted* graphs. In our definition, a graph has bounded growth rate k if the number of nodes in any subgraph with diameter r is bounded by $O(r^k)$. 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 [23, 24, 16, 17]. For example, the unit disk graph has been used extensively to model a wireless network, which consists of a set of nodes with direct communication between those pairs of nodes within distance 1 from each other. When the maximum density of the nodes, i.e., the maximum number of nodes covered by a unit disk, is a constant, the unit disk graph has growth rate 2. As another example [23], the Internet network distance defined by round-trip propagation and transmission delay forms a metric with restricted growth rate. Therefore, many algorithms for peer-to-peer networks, such as media file sharing on Internet, content addressable overlay networks, exploit this property [24, 16, 17]. In this paper we show that for growth restricted graphs, there exists a tradeoff between the stretch factor and load balancing ratio with a dependency on the growth rate. Since the direct motivation of our work is from wireless networks, we will illustrate our results by using wireless network routing and then show the extension to general growth restricted networks.

The nodes in a wireless network are usually energy constrained as they are normally powered by batteries. Therefore it is crucial to balance the load on the nodes because an uneven use of the nodes may cause some nodes die much earlier, thus creating holes in the network. In addition, unbalanced use of the nodes may discourage them to participate in the routing. For these reasons, there has been extensive work on energy-aware routing in wireless networks [15, 14]. But most previous work focus on minimizing the overall energy consumption. In this paper, we are more concerned about minimizing the maximum energy consumption of nodes in the network. Since a major amount of energy consumed by a wireless node is on communication, we measure the load of a routing scheme to be the maximum total size of packets that pass any node in the network. Then the load balancing problem can be formulated as minimizing the load and is exactly the classic load balancing routing for connections with permanent duration [8].

The ideal algorithm would be the one that achieves 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 [13], it is possible to design a routing algorithm achieving both a constant stretch factor and a constant load balancing ratio. This is however 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 (or c-short paths) necessarily causes some node to have $\Omega(n/c)$ loads while the optimal load-balancing algorithm only loads $O(1)$ packets on each node. Such an example, however, uses highly crowded nodes. In this paper, we show that if the nodes are not so crowded, then it is impossible to construct such a bad example. We use two metrics to measure the density of wireless nodes: the *maximum density* of a set of nodes is defined as the maximum number of nodes covered by any unit disk in the plane, and the *average density* as the average number of nodes covered by the unit disks centered at the nodes in the set. Both definitions appear naturally in analyzing wireless networks. In practice, we can expect the density of the wireless nodes to be low. Indeed, using multi-hop peer-to-peer communication to reduce the deployment and communication cost is exactly a goal of a wireless network. In a dense network, techniques such as clustering are often used to reduce routing complexity by routing on a smaller set of "backbone" nodes. The "backbone" nodes usually have constant density [11, 12].

In this paper, we show tradeoffs between the stretch factor and the load balancing ratio with a dependency on the density of the wireless nodes. Specifically, if the maximum density of a point set is ρ, then a c-short path routing can achieve a load balancing ratio of $O(\min(\sqrt{\rho n/c}, n/c))$. In particular, if we use shortest path routing, i.e., when the stretch factor c is 1, the load balancing ratio is $O(\sqrt{\rho n})$. When ρ is constant, it is $O(\sqrt{n})$. Furthermore, all those bounds are tight asymptotically in the worst case. When nodes are not evenly distributed, the average density is a more appropriate measure. We obtain similar tradeoffs in terms of the average density.

Our tradeoffs rely on the fact that the number of nodes inside any disk is polynomial in the radius of the disk. Therefore, the results naturally extend to higher dimensional unit ball graphs and to graphs with restricted growth rate. We show that for a k dimensional unit-ball graph with maximum density ρ, the load balancing ratio of the optimal routing algorithms with a stretch factor c is bounded by $O((n/c)^{1-1/k}\rho^{1/k})$, and for a graph with growth rate k, the load balancing ratio is $O((n/c)^{1-1/k})$.

Another application of our results is in global routing in VLSI design [26, 25]. In VLSI routing, given a graph (typically a mesh) which represents the physical wiring paths of a chip and a set of vertex pairs, one needs to connect every pair by a path in the graph. The goal is to minimize the line width on each vertex. The seminal work of Raghavan and Thompson [26] 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 [1], 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 the graph in VLSI routing is a graph with restricted growth rate due to physical constraint, our $O(\sqrt{n})$ upper bound implies that the construction in [1] is actually the worst possible case. Further, the bound holds as long as the underlying graph is similar to the unit disk graph with constant-bounded node density, which is the case due to the physical constraint.

In addition to the combinatorial bounds, we also observe that the on-line algorithm for virtual-circuit routing developed in [3, 4] can be adapted to obtain an on-line c-short-path routing algorithm that is $O(\log n)$ competitive in terms of load balancing ratio, compared with the optimal c-short-path routing algorithm. Another common approach of reducing routing complexity in the wireless network is to extract a sparse spanner graph [10] from the unit-disk graph and only route packets on the spanner [12, 20]. 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.

2 Preliminaries

Given an unweighted graph $G = (S, E)$, we denote by $|P|$ the number of points on a path P in the graph G. We assume that G is connected, otherwise we can consider each connected component individually. For any two points $p, q \in S$, denote by $\tau(p, q)$ the length of the shortest path between p and q. For any 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 $\omega(P) \leq c$. We say G has growth rate k if for any $p \in S$ and any $r \geq 1$, the number of nodes in $B = \{q|\tau(p, q) \leq r\}$ is bounded by $O(r^k)$.

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 $\mathcal{P} \models R$, if $\mathcal{P} = \{P_r \,|\, r \in R\}$ 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 $\max_{v\in S}\ell(v)$. 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 tradeoff between the stretch factor and the load-balancing ratio of routing algorithms in a network.

We now give definitions that are particular for wireless networks. Let S be a set of n points in the plane which represent wireless nodes. Let $|pq|$ denote the Euclidean distance between two nodes p, q. The *communication graph* of S is an unweighted unit-disk graph $U(S) = (S, E)$ of S, where $(p, q) \in E$ iff $|pq| \leq 1$. We say two points p, q *see* each other if $|pq| \le 1$. The *maximum density* (or density in short), $\rho(S)$ of S, is defined as the maximum number of nodes in S covered by any unit disk (disk with radius 1). For each $p \in S$, denote by $\rho(p)$ the number of points p sees (including p itself). Define the *average density* $\bar{\rho}(S)$ of S to be $\sum_{p \in S} \rho(p)/n$. Clearly, $\bar{\rho}(S) \le \rho(S)$. The following facts will be useful later.

Lemma 1. *For any disk B with radius $r \ge 1$,*

1. $|B \cap S| = O(\rho(S) r^2)$;
2. $|B \cap S| = O(r\sqrt{n\bar{\rho}(S)})$.

PROOF. Since B can be covered by $O(r^2)$ unit disks, we have $|B \cap S| = O(\rho(S) r^2)$. For the second claim, suppose that there are x points in $B \cap S$. We can partition $B \cap S$ into $O(r^2)$ disjoint subsets such that all the points in one subset are mutually visible[1]. Suppose that those sets are $S_1, \ldots, S_m$, and let $n_i = |S_i|$. Therefore, $\sum_i n_i^2 \le n\bar{\rho}(S)$. By the Cauchy-Schwartz inequality, we have that $x^2 = (\sum_i n_i)^2 \le m(\sum_i n_i^2) \le mn\bar{\rho}(S)$. Since $m = O(r^2)$, $x = O(r\sqrt{n\bar{\rho}(S)})$. □

3 Tradeoffs in wireless network routing

If we consider the general case, it is only possible to obtain a weak tradeoff between the stretch factor and the load-balancing ratio. The simple example in Figure 1 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 $3c + 1$ spots on a loop. Each spot contains n/c wireless nodes, except one spot has only one node o. The total number of nodes is $3n + 1$. 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 doesn't pass through o has length at least $3c$, i.e., is not c-short path. 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.

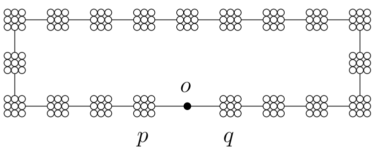

Figure 1. Each spot contains n/c nodes. The loop has $3c + 1$ spots. The packets from spot p to q either go through node o, thus causing the node o to be heavily loaded, or route along a long path with length $\Omega(n/c)$.

The above configuration uses a point set with high density. The main result of this paper is to show that there is a tradeoff between the stretch factor and the load-balancing ratio dependent on the density of the point set. In this section, we first present a tight tradeoff based on the maximum density. Then, we show a slightly weaker bound dependent on the average density.

3.1 Tradeoff based on the maximum density

Our main result for the maximum density is as follows:

Theorem 2. *For any n nodes with the maximum density ρ and any set of requests R, $\ell^c(R)/\ell^*(R) = O(\min(\sqrt{\rho n/c}, n/c))$. This bound is tight in the worst case.*

[1] It's possible that two points in different subsets are visible.

As a special case of the above theorem, when the set of nodes has constant bounded density, then the load-balancing ratio of the optimal c-short path routing is bounded by $O(\sqrt{n/c})$. In another special case, $c = 1$, 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 above theorem for the case of shortest path routing and extend the technique to prove Theorem 2.

Theorem 3. *For any n nodes with the 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 but can only decrease the maximum load of the optimal routing algorithm. 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 point $q \in S$, denote by $R(q)$ all the requests that originate at q and by $\ell(q)$ the total size of those packets, i.e. $\ell(q) = \sum_{r_i \in R(q)} \ell_i$. Write $\beta(q) = \ell(q)/\ell$, where $\ell = \sum_{i=1}^m \ell_i$. Clearly $\sum_q \beta(q) = 1$.

Lemma 4. *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 with an inner radius of 2^{k-1} and an outer radius of 2^k. See Figure 2. Consider the set R_k of the requests originating

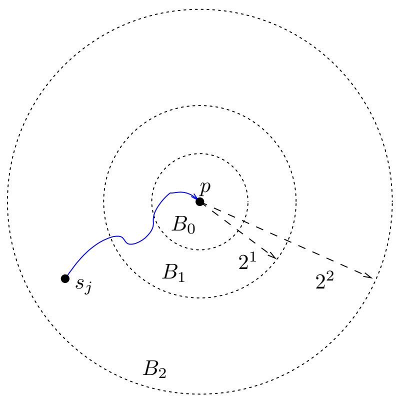

Figure 2. Division of D_τ into a set of disjoint sets B_i. All the traffic pass through the center p by shortest path routing.

at some point 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 point 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., $\tau(s_j, t_j) \geq \tau(p, s_j) \geq |ps_j| \geq 2^{k-1}$. 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 points on P_j is at least 2^{k-1}. Let A_j be the first 2^{k-1} points 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. Firstly we have

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

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

$$\sum_{v \in S_k} \ell(v) \leq |S_k| \ell^* \leq c_0 \rho (2^{k-1})^2 \ell/\alpha \,, \tag{2}$$

for some constant $c_0 > 0$. Combining (1) and (2), we have that

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

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

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

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

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

□

Now we proceed to prove Theorem 3. We can assume that for any $q \in S$, $\beta(q) \leq 1/3$; otherwise $\ell^* \geq \ell(q) > \ell/3$, i.e. $\alpha < 3$. Now, consider the smallest disk D centered at p such that

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

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 $\beta(q) \leq 1/3$ for any q, we have that

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

Let τ^* denote the radius of D. Then, by Lemma 4,

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

i.e.

$$\alpha \leq 2 c_0 \rho \tau^* \,. \tag{3}$$

On the other hand, for any point $q \notin D$, $|pq| \geq \tau^*$. By the same argument used in the proof of Lemma 4, for any algorithm, the loads incurred by those requests originating at q are at least $\ell(q)\tau^*$. Therefore, the total loads caused by such requests are at least

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

The last inequality is due to that $\sum_{q \notin D} \beta(q) \geq 1/6$. Hence, the optimal load balancing routing algorithm can do no better than distributing these loads evenly on the n nodes. That is, $\ell^* \geq \ell \tau^*/6n$, i.e.

$$\alpha = \ell/\ell^* \leq 6n/\tau^* \,. \tag{4}$$

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

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

for $c_1 = \sqrt{12 c_0}$. This proves Theorem 3. □

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

PROOF OF THEOREM 2. 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. 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 $\mathcal{P}$ is at most $O(\min(\sqrt{\rho n/c}, n/c)\ell^*(R))$.

For each point $q \in S$, 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, $\ell^* \geq \max(\ell_1^*, \ell_2^*) \geq (\ell_1^* + \ell_2^*)/2$. For each point $q \in S$, 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, $\ell^c(R) \leq \ell(\mathcal{P}) \leq \ell_1^* + \ell_2(R)$.

We now bound $\ell_2(R)/\ell_2^*$ by using almost the same argument as in the proof of Theorem 3. 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 $\sum_{q \notin D} \ell(q) \cdot c\tau^*$, which is equal or greater than $\ell c \tau^*/6$. Then we can replace (4) with the following inequality

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

Since (3) is still valid, we have that

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

Therefore,

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

This proves the upper bound in Theorem 2.

In the following, we show a lower bound construction. We only describe the lower bound construction for $\rho c \leq n$, i.e. $\sqrt{\rho n/c} \leq n/c$. The other case is similar. Consider the example illustrated in Figure 3. The distance between u, v is 1. Take a parameter $m > 1$ which will be determined later, we place $k = \rho m$ points $p_1, \ldots, p_k$ 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 $2m$ points evenly. In addition, there is a path between every pair of p_i and q_i as drawn in Figure 3. Each path is about $4cm$ long and has $4cm$ points on it. Clearly, the maximum density of the point 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

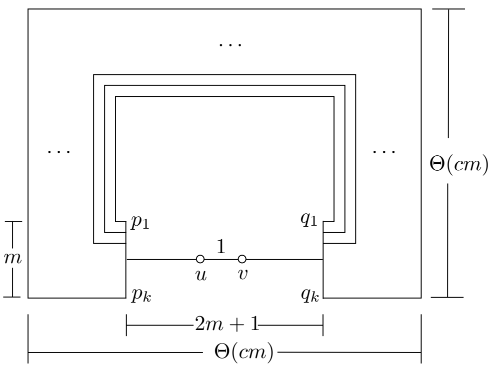

Figure 3. Lower bound of the load-balancing ratio for the optimal c-short routing with maximum density ρ.

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 $1 \le i \le k$, 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 point. Thus, the load-balancing ratio of any c-short path routing of this example is $\Omega(\rho m)$. The total number of points 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. □

We should emphasize that in the proof of Theorem 3, 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 2 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 5, 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.

3.2 Tradeoff based on average density

We now show the tradeoff based on the average density of the point set. The benefit of considering average density is clear — it is applicable to a wider family of point sets, in particular to the point sets with uneven distribution.

Theorem 5. *Given a set of n nodes S 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 3. We use the notation in the proof of Lemma 4. We take τ to be the diameter of the communication graph. $\tau = O(n)$. 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, $0 \le k \le \log\tau$, where B_0 is the unit disk centered at p and for $k \ge 1$, B_k is an annulus with an inner radius of 2^{k-1} and an outer radius of 2^k. The only difference is that with the average density, by Lemma 1, we can only bound $|\cup_{r_j \in R_k} A_j| = O(\sqrt{\bar{\rho}n}2^k)$, for $1 \le k \le \log\tau$. Since each node has load at most ℓ^*, we have that

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

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

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

for some constant c_1.

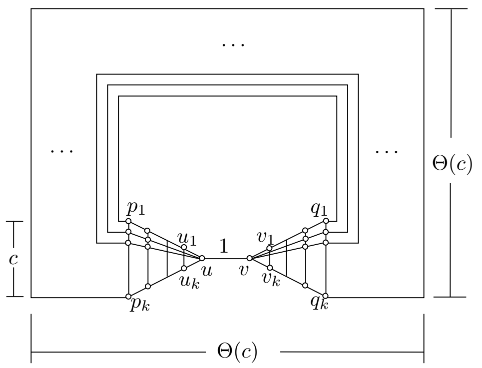

Figure 4. Lower bound of the load-balancing ratio for the optimal c-short routing with average density $\bar{\rho}$.

As for the lower bound, consider the example shown in Figure 4. 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 $1 \le i \le k$. 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 in the figure is bounded by $O(c \cdot k)$. To bound the average density of the point set, we consider two types of points. For a point x on a vertical bar with length h, the number of points 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 point 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 $\Omega(\sqrt{\bar{\rho}n/\max(1, \log c)})$. □

4 Extensions

The above results naturally extend to other routing problems and to a larger family of growth-restricted graphs.

4.1 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 [26, 27, 19, 6]. A mesh can be realized as a unit disk graph of a set of points 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 $O(\sqrt{n})$.*

4.2 Unit ball graph in higher dimensions

Similar results hold for unit ball graphs in higher dimensions. The definitions in Section 2 extend naturally to points in higher dimensions. We can apply the same technique to obtain the following.

Theorem 7. *For n point in $\mathbb{R}^k$ with maximum density ρ, the load-balancing ratio of the optimal c-short routing is $O((n/c)^{1-1/k}\rho^{1/k})$. In particular, the load-balancing ratio of (any) shortest path routing is $O(n^{1-1/k}\rho^{1/k})$.*

4.3 Growth restricted graphs

If we examine the proof of Theorem 2, we can see that the only property we needed for the proof is that there are $O(\rho r^2)$ nodes inside any disk with radius r. Thus, the result extends immediately to graphs with small growth rate. In our definition, a graph has *density* ρ and *growth rate* k (or growth dimension k) if for any vertex v and any $r > 1$, $|B_r(v)| \le \rho r^k$, where $B_r(v) = \{u|\tau(u,v) \le r\}$, the ball with radius r centered at v. By using exactly the same argument, we have that

Theorem 8. *For a graph with density ρ and growth rate k, the load-balancing ratio of the optimal c-short routing is $O((n/c)^{1-1/k}\rho^{1/k})$. In particular, the load-balancing ratio of (any) shortest path routing for a graph with constant density and growth rate k is $O(n^{1-1/k})$.*

We should note that there are several other definitions for capturing metrics with slow growth. For example, in [16], a metric has *expansion rate* k (or KR-dimension $\log k$) if $|B_{2r}(v)| \le k|B_r(v)|$; and in [2], a metric has *doubling constant* k (or doubling dimension $\log k$) if $B_{2r}(v)$ is contained in the union of at most k balls with radius r. Both definitions imply that the size of B_r is bounded by $O(k^{\log r}) = O(r^{\log k})$. On the other hand, we can construct a family of graphs, e.g. the comb graphs as shown in Figure 5, with constant density and growth rate but unbounded KR-dimension and unbounded doubling dimension. Therefore, our definition is broader in the sense that any graph with KR-dimension or doubling dimension d also has a growth dimension d.

5 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. Here, we describe an algorithm that computes c-short paths with maximum load within an $O(\log n)$ factor of the optimum.

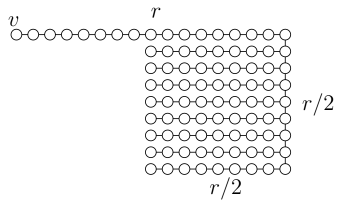

Figure 5. The "comb" graph is a unit disk graph with constant bounded density. Therefore it's a graph with growth rate 2. It is not a metric with constant KR-dimension since $|B_{2r}(v)| = \Theta(r^2)$ and $|B_r(v)| = \Theta(r)$. And, it is not a metric with constant doubling dimension: the comb graph has diameter $2r$, it can not be covered by a constant number of balls with diameter r, since the diameter of a set including two teeth of the comb is at least $r+1$.

One general approach for approximation is by the randomized rounding technique [26, 25]. But that technique cannot be directly applied to our case because of the restriction on the stretch factor — it will make the size of the linear programming problem exponentially large. But we show that the on-line virtual circuit routing algorithm by Aspnes *et al.* [3] applies to our problem to obtain an $O(\log n)$ approximation ratio.

Theorem 9. *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 apply the method in [3] with slight modification. In the algorithm in [3], a weight is assigned to each edge (or vertex in our case) according to the current load on the edge and the size of the request. Then for any new request, the lightest path[2] 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 [3]. 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.

The lower bound construction in [7] can be used to 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. Actually, we can show a stronger result where any on-line algorithm is $\Omega(\log n)$ competitive even when compared against the optimal off-line algorithm that only uses the shortest paths. The details will appear in the full version. □

The above algorithm only discovers the paths but doesn't deal with the scheduling in the routing, for examples, the queueing principle when multiple packets need to be delivered from the same

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

node at the same time, or the interference resolution when multiple nearby nodes transmit packets. The methods in [18] or [22] can be used for such scheduling after the path selection step.

6 Load-balancing ratio of routing on spanners

One important method to reduce the complexity of routing in wireless network is to construct a sparse spanner graph and route on the spanner graph [12, 20]. A sub-graph G of a unit-disk graph $U(S)$ is a c-spanner if the shortest path between any two points in G is c-short compared with $U(S)$. 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 10. *Suppose S is a set of n points in the plane with density ρ, and G is a c-spanner of $U(S)$, 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 ($U(S)$). 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 point $p \in S$. A packet can be redirected to it only if it is routed in the optimal solution through a point 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 points. Therefore, the load on p is $O(\rho c^2 \ell^*)$.

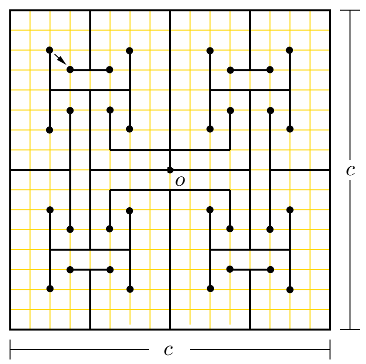

Figure 6. Lower bound $\Omega(c^2)$ on the competitive ratio of load balanced routing algorithms on c-spanners.

As for the lower bound, we use the classic H-tree construction [21]. We only show the construction for points with constant density. The extension to points with other density is easy – we just put ρ copies on each grid node. Consider $\Theta(c^2)$ points positioned on a grid as shown in Figure 6. 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 point of the H-tree to its nearby point on the complement part of the H-tree (see little arrow in Figure 6). The optimal solution can send the requests directly. However, in G, all the requests have to be routed through the point o. Therefore, the load-balancing ratio of the routing on this c-spanner is $\Omega(c^2)$. □

7 Conclusion

In this paper, we study the tradeoff between two important quality measures of routing algorithms for wireless networks and growth-restricted networks: the stretch factor for measuring the path length and the load balancing ratio for measuring the load balance. We show several tradeoffs based on the maximum and the average density of the wireless nodes. There is still a gap for the tradeoff when considering average density. Besides, all of our results are based on the worst case analysis. It would be interesting to study the tradeoff under some reasonable traffic model. The issue of interference between wireless communication links is not considered in this paper. A future direction is to study the tradeoff under proper interference models such as the one in [22].

Acknowledgments: The authors wish to thank the anonymous reviewers for suggesting relevant work and improving the writing of this paper.

References

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

[2] R. K. Anupam Gupta and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proc. 44th Symposium on Foundations of Computer Science (FOCS '03)*, pages 534–543, 2003.

[3] 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. 25th ACM Symposium on Theory of Computing*, pages 623–631, 1993.

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

[5] 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.

[6] 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.

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

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

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

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

[11] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Discrete mobile centers. *Discrete and Computational Geometry*, 30(1):45–63, 2003.

[12] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *Proc. 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 01')*, pages 45–55, 2001.

[13] J. Gao and L. Zhang. Load balanced short path routing in wireless networks. In *Proc. IEEE INFOCOM'04*, 2004.

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

[15] 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.

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

[17] R. Krauthgamer and J. R. Lee. The intrinsic dimensionality of graphs. In *Proceedings of the 35th ACM symposium on Theory of computing*, pages 438–447. ACM Press, 2003.

[18] 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.

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

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

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

[22] F. Meyer auf de Heide, C. Schindelhauer, K. Volbert, and M. Grünewald. Energy, congestion and dilation in radio networks. In *Proceedings of the 14th ACM symposium on Parallel algorithms and architectures*, pages 230–237. ACM Press, 2002.

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

[24] 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.

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

[26] P. Raghavan and C. D. Thompson. Provably good routing in graphs: regular arrays. In *Proceedings of the 17th annual ACM Symposium on Theory of Computing*, pages 79–87, 1985.

[27] T.-H. Yeh, C.-M. Kuo, C.-L. Lei, and H.-C. Yen. Competitive source routing on tori and meshes. In *ISAAC: 8th International Symposium on Algorithms and Computation*, pages 82–91, 1997.