Tradeoff between Stretch Factor and Load Balancing Ratio in Wireless Network Routing

Jie Gao*  Li Zhang†

Abstract

A wireless multi-hop network consists of a set of nodes in the plane where two nodes can directly communicate to each other if their distance is at most 1. We consider two quality measures for routing in wireless networks. One is the stretch factor of the paths used by a routing algorithm, and the other is the load balancing ratio, which measures how evenly the traffic is distributed. In this paper, we show a trade-off between the two measures dependent on the density of the point set. When the maximum density of the point set is $\rho$, the optimal algorithm by using only paths with stretch factor no more than $c$ generates the maximum load on the nodes at most $O(\min(\sqrt{pn/c}, n/c))$ times that of the optimal algorithm without path length restriction. In particular, when the density is bounded by a constant, the shortest path routing has an approximation ratio of $O(\sqrt{n})$ compared to the optimal algorithm. This bound is tight in the worst case. The result can be extended to $k$-dimensional unit-ball graphs and graphs with “growth rate” $k$. We also present a tradeoff when considering the average density $\bar{\rho}$. We show a bound of $O(\sqrt{pn \log n})$ and $\Omega(\sqrt{pn / \log c})$ for this case. We also discuss issues such as algorithms for computing load-balanced short path routing and load-balanced routing in spanner 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 the shortest-path routing algorithms [11] and the load-balanced routing algorithms [4, 7]. These two families can be regarded as to minimize two different quality measures: the stretch factor of the paths, defined to be the 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 ratio between the maximum load incurred by the algorithm and that of the optimal load-balancing routing algorithm. Both families of routing algorithms have been studied extensively in the literature. However, there has been virtually no work done on studying the tradeoffs between these two measures. This probably should not be surprising as they are conflicting goals to some extent: the shortest-path routing restricts the resources that can be used, while a load-balanced routing tries to use all the resources available to even the load. One can easily construct examples to show that these two goals sometimes are indeed conflicting, i.e. a shortest-path routing algorithm necessarily creates heavily loaded nodes, and a load-balanced

*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.
†Systems Research Center, Hewlett-Packard Labs, 1501 PageMill Road, Palo Alto, CA 94304. E-mail: l.zhang@hp.com.
routing algorithm necessarily uses long paths. In this paper, we study tradeoffs between these two measures for routing in wireless networks. We show that there exist non-trivial tradeoffs dependent on the density of the points in the network.

A wireless network consists of a set of wireless nodes in the plane. Two nodes can directly communicate with each other if their Euclidean distance is no more than 1. The graph formed by connecting those pairs of nodes within distance 1 is also known as the unit-disk graph [10]. Such graphs have recently attracted much attention due to the increasing interests in ad hoc networks and sensor networks [8, 31, 17, 32, 16, 9]. One important restriction of a wireless network is that the nodes are energy constrained as they are normally powered by batteries. It is crucial to balance load on the nodes because an uneven use of the nodes may cause some nodes die much earlier, thus creating holes in the network, or worse, leaving the network disconnected. Due to its importance, there has been extensive work on energy-aware routing in practice [18, 15]. Since the energy consumed by a node is largely determined by the size of packets it ever relayed, we can define 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 can be formulated to minimize the load and is exactly the classic load balancing routing for connections with permanent duration [7]. We then measure the load-balancing ratio of a routing algorithm by comparing its performance to the optimal load-balancing solution.

Of course, the ideal algorithm would be to achieve good performance in terms of both the stretch factor and the load balancing ratio. This is however in general impossible as it is not difficult to construct a set of nodes and a set of routing requests such that a routing algorithm limited to using paths with stretch factor $c$ (or $c$-short paths) will necessarily cause some node to have $\Omega(n/c)$ loads while the optimal load-balancing algorithm only create $O(1)$ load on every node. This example, however, is only possible with highly crowded nodes. In practice, the wireless nodes prefer to use short range communication and therefore tend to distribute sparsely, compared to their communication range. Indeed, using multi-hop peer-to-peer communication to reduce the deployment and communication cost is exactly a goal of ad-hoc networks and sensor networks.

In this paper, we use density as a sparsity measure for a point set, where the maximum density is defined as the maximum number of points covered by any unit disk in the plane, and the average density as the average number of points covered by the unit disks centered at the points in the set. We show that there are non-trivial tradeoffs involving the density of the wireless nodes. For example, if the maximum density of a point set is $\rho$, then a $c$-short path routing can achieve a load balancing ratio of $O(\min(\sqrt{m/c}, n/c))$. In particular, if we use shortest path routing, i.e., when $c = 1$, the load balancing ratio is $O(\sqrt{m})$. When $\rho$ is constant (this includes the graphs such as meshes), the bound is then $O(\sqrt{n})$. Furthermore, all those bounds are tight asymptotically in the worst case. When points are not evenly distributed, the average density is a more appropriate measure. We also obtain similar tradeoffs involving the average rather than the maximum density. Taking density into account appears naturally in analyzing wireless networks, even in a dense network. Techniques such as clustering are often used to reduce the routing complexity by reducing the problem on the original point set to the routing on a smaller set of “backbone” nodes. The “backbone” nodes usually have small density [13, 14].

Recently, there have been increasing attention paid to metrics with restricted growth rate [26]. People have shown efficient algorithms or better properties of such metrics [22, 28, 21, 19, 23]. Wireless networks for points with bounded density can be naturally regarded as a special type of growth-restricted network. The tradeoffs we show for wireless networks extend to points in higher dimensions and other growth-restricted network. Therefore, our work adds another set of
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 (spanner graph short-path routing algorithm when compared to the on-line optimal P-path density.

Let \( \rho(S) \) of \( S \) be defined to be the maximum number of points 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 \( p \)). Define the average density \( \bar{\rho}(S) \) of \( S \) to be \( \sum_{p \in S} \rho(p)/n \). Clearly, \( \bar{\rho}(S) \leq \rho(S) \). The following properties are useful:

**Lemma 1.** For any disk \( B \) with radius \( r \geq 1 \), \( |B \cap S| = O(\rho(S)r^2) \) and \( O(n \sqrt{\bar{\rho}(S)}) \).

**Proof:** It is clear that \( |B \cap S| = O(\rho(S)r^2) \) since \( B \) can be covered by \( O(r^2) \) unit disks. For the other 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 two points see each other if they are in the same subset. Suppose that those sets are \( S_1, \ldots, S_m \), and let \( n_i = |S_i| \). Therefore, \( \sum_i n_i^2 \leq \bar{\rho}(S) \). By Cauchy-Schwartz inequality, we have that \( X^2 = (\sum_i n_i)^2 \leq m(\sum_i n_i^2) \leq m\bar{\rho}(S) \). Since \( m = O(r^2) \), \( X = O(r \sqrt{n \bar{\rho}(S)}) \).

The length \( |P| \) of a path \( P \) in the graph \( U(S) \) is the number of points on the path. For any two points \( p, q \in S \), denote by \( d(p, q) \) 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 \( |P|/d(p, q) \). \( P \) is called c-short if \( \omega(P) \leq c \).

A **routing request** is of the form \( r = (s_r, t_r, \ell_r) \) where \( s_r, t_r, \ell_r \) 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 \( \mathcal{P} = \{ P_r \mid r \in R \} \) 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

\[ \begin{align*}
\omega(\mathcal{P}) & \leq c \\
\text{for all } \mathcal{P} \models R.
\end{align*} \]
it only uses paths with stretch factor at most $c$. For example, the 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 $\ell(v)$ incurred to $v$ is the total size of the packets that pass $v$, i.e. $\ell(v) = \sum_{v \in P, v \not\in P, v \in E, \ell_v}$. The load $\ell(P)$ of $\mathcal{P}$ is then defined to be $\max_{v \in S} \ell(v)$. Define $\ell^*(R) = \min_{\mathcal{P} = R} \ell(P)$ to be the optimal load for satisfying $R$ and $\ell^c(R) = \min_{\mathcal{P} = R, \omega(\mathcal{P}) \leq c} \ell(P)$ the optimal load by any $c$-short routing algorithm. For example, $\ell^1(R)$ is the optimal 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 $\max_{R} \ell(\mathcal{A}(R))/\ell^*(R)$. We generally call it a 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 for wireless networks.

3 Tradeoff based on the maximum density

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 following simple example shows that if we insist to use $c$-short paths, then the load-balancing ratio can be as bad as $\Omega(n/c)$: Consider two points $u, v$ with distance 1 apart and a circle with radius $\Theta(c)$ and passing through both $u$ and $v$. We distribute $n$ points uniformly on the longer arc between $u$ and $v$ on the circle. If we make $n/c$ requests, each from a point close to $u$ to another point close to $v$. By choosing constant appropriately, we can force any path along the longer arc $uv$ to have more than $3c$ points. Therefore, any $c$-short routing algorithm has to route the requests through $u$ and $v$, i.e. $u$ and $v$ have load $\Theta(n/c)$. On the other hand, the optimal solution can route the requests evenly along the path on the longer arc such that each node only passes $O(1)$ packets.

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

Our main result for maximum density is as follows:

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

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

**Theorem 3.** For any $n$ nodes with the maximum density $\rho$ and any set of requests $R$, $\ell^1(R)/\ell^*(R) = O(\sqrt{n})$.

**Proof:** Suppose that $p$ is the point with the maximum load when using the shortest path routing. Without loss of generality, we can assume that all the requests in $R$ are routed through $p$ by the shortest path routing because otherwise we can safely delete those requests that do not — this does not change the maximum load by the shortest path routing but can only decrease the maximum load of the optimal routing algorithm. Suppose that $R = \{r_1, \ldots, r_m\}$ where $r_i = (s_i, t_i, \ell_i)$ is a request from $s_i$ to $t_i$ with packet size $\ell_i$. For each point $q \in S$, denote by $R(q)$ all the requests
that are originated from \( q \) and by \( \ell(q) \) the total size of those packets, i.e. \( \ell(q) = \sum_{i \in R(q)} \ell_i \). Write 
\[
\beta(q) = \ell(q)/\ell, \quad \text{where } \ell = \sum_{i=1}^{m} \ell_i. 
\]
Clearly \( \sum_q \beta(q) = 1. \) Write \( \ell^* = \ell^*(R) \). We now wish to upper-bound \( \alpha = \ell/\ell^* \).

The intuition of the proof is that the 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 optimal algorithm to create high total loads as well. The average load therefore cannot be too low, even those loads can be evenly distributed. This intuition is made concrete by the following lemma.

**Lemma 4.** Suppose that \( D_\tau \) is the disk with radius \( \tau \geq 1 \) centered at \( p \), then \( \sum_{q \in D_\tau} \beta(q) \leq c_1 \rho \tau/\alpha \), for some constant \( c_1 > 0 \).

**Proof:** We partition \( D_\tau \) into a set of \( \log \tau \) disjoint annuli \( B_k \), \( 0 \leq k \leq \log \tau \), where \( B_0 \) is the unit disk centered at \( p \) and for \( k \geq 1 \) \( B_k \) has an inner radius of \( 2^{k-1} \) and an outer radius of \( 2^k \). Consider the set \( R_k \) of the requests originated from 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 number of the points in the shortest path is at least \( \delta(p, s_j) \geq 2^{k-1} \). 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 \( P_j \). Thus,
\[
\sum_{r_j \in R_k} \ell_j|A_j| = 2^{k-1} \sum_{r_j \in R_k} \ell_j. 
\]

On the other hand, for any point \( a \in A_j \), \( \delta(p, a) \leq \delta(p, s_j) + \delta(s_j, a) \leq 2^k + 2^{k-1} = 3 \cdot 2^{k-1} \). That is, all the points in \( A_j \) are inside a disk with radius \( 3 \cdot 2^{k-1} \) centered at \( p \). Therefore, \( |\cup_{r_j \in R_k} A_j| = O(\rho(3 \cdot 2^{k-1})^2) \). Since each node has load at most \( \ell^* = \ell/\alpha \), we have that
\[
2^{k-1} \sum_{r_j \in R_k} \ell_j \leq c_0 \rho(3 \cdot 2^{k-1})^2 \ell/\alpha, 
\]
for some constant \( c_0 > 0 \).

Thus \( \sum_{r_j \in R_k} \beta_j = \sum_{r_j \in R_k} \ell_j/\ell \leq c_0 \rho 2^{k-1}/\alpha. \) By summing up for all the \( k \)'s, we have that \( \sum_{q \in D_\tau} \beta(q) \leq 4c_0 \rho \tau/\alpha. \) Setting \( c_1 = 4c_0 \), we proved the claim. \( \Box \)

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. \( \alpha < 3 \). We also assume that all the points have different distances to \( p \); otherwise we can perturb (conceptually) the points so the assumption is valid. Now, consider the minimum disk \( D \) centered at \( p \) such that \( \sum_{q \in D} \beta(q) \geq 1/2 \). Since \( \beta(q) \leq 1/3 \), we have that \( \sum_{q \in D} \beta(q) \geq 1/6 \). Let \( \tau^* \) denote the radius of \( D \). Then, by Lemma 4, \( c_1 \rho \tau^*/\alpha \geq \sum_{q \in D} \beta(q) \geq 1/2 \), i.e.
\[
\alpha \leq 2c_1 \rho \tau^*. \tag{1}
\]

On the other hand, for any point \( q \notin D \), \( \delta(p, q) \geq \tau^* \). By the same argument used in the proof of Lemma 4, for any algorithm, the loads incurred by those requests originated from \( q \) are at least \( \ell(q) \tau^* \). Therefore, the total loads caused by such requests are at least \( \sum_{q \notin D} \ell(q) \tau^* = \sum_{q \notin D} \beta(q) \ell \tau^* \geq \ell \tau^*/6 \). That is, \( \ell^* \geq \ell \tau^*/6 \), i.e.
\[
\alpha = \ell/\ell^* \leq 6n/\tau^*. \tag{2}
\]
By combining (1) and (2), we have that 
\[ \alpha \leq \min(2c_1\rho \tau^*, 6n/\tau^*) \leq c_2 \sqrt{pm} \text{ for } c_2 = \sqrt{12c_1}. \] This proves Theorem 3. \[\square\]

Now, we extend the result to \(c\)-short routing. 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 \(R_2\) those requests routed by non-\(c\)-short paths. We construct a set of paths \(P\) as follows. We include in \(P\) the paths that the optimum algorithm produced for requests in \(R_1\). For each request in \(R_2\), we add to \(P\) (any) shortest path between the source and destination of that request. Clearly, all the paths in \(P\) are \(c\)-short. We now show that the maximum load caused by \(P\) is at most \(O(\min(n/c, c/n)\ell^*(R))\).

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 the shortest path routing. Let \(\ell_2(R) = \max_q \ell_2(q)\). Clearly, \(\ell^c(R) \leq \ell(P) \leq \ell^*_1 + \ell_2(R)\).

We now bound \(\ell_2(R)/\ell^*_2\) by using almost the same argument in the proof of Theorem 3. The only difference is that since all the paths used to route requests in \(R_2\) by the optimal algorithm are not \(c\)-short, we can replace (2) with the following inequality
\[
\ell^c_2(R) \leq \frac{6n}{c \tau^*}
\]
Since (1) is still valid, we have that \(\ell_2(R)/\ell^*_2 = \min(2c_1\rho \tau^*, 6n/(c \tau^*)) = O(\min(\sqrt{pm}/c, n/c))\). Therefore,
\[
\ell^c(R) \leq \ell^*_1 + \ell_2(R) = O(\min(\sqrt{pm}/c, n/c))(\ell^*_1 + \ell^*_2) \\
= O(\min(\sqrt{pm}/c, n/c)) \cdot \ell^*.
\]

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. \(\sqrt{pm}/c \leq n/c\). The other case is similar. Consider the example illustrated in Figure 1. The distance between \(u, v\) is 1. Take a

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{Lower bound of the load-balancing ratio for \(c\)-short routing with maximum density \(\rho\).}
\end{figure}
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 1. Each path is about $\Theta(cm)$ long and has $\Theta(cm)$ 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 $u,v$ has to go through the outside loop with length $\Theta(cm)$. If we choose constant large enough, then 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 those outside 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 $\Theta((\rho m) \cdot (cm)) = \Theta(\rho cm^2)$. Setting $m = \sqrt{n/(c\rho)}$, we obtain the desired lower bound. 

We should remark 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 for any shortest path. However, the proof of Theorem 2 does use a set of $c$-short paths produced by the optimal algorithm. Therefore, the bound only holds for the optimal $c$-short routing algorithm. We will discuss the algorithmic issues later.

The load balanced routing using shortest paths on meshes has been studied extensively in the area of global wiring in VLSI and parallel computing [30, 33, 24, 5]. It is easy to see that the 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 the line width in VLSI routing.

**Corollary 5.** If we are restricted to use $c$-short path 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 $O(\sqrt{n})$.

### 3.1 Extensions

The above results can be extended to higher dimensions and to graphs with bounded growth-rate. The definitions in Section 2 extend naturally to points in higher dimensions. It is not difficult to apply the same technique to obtain the following.

**Theorem 6.** For $n$ point in $\mathbb{R}^k$ with maximum density $\rho$, the load-balancing ratio of 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 $O(n^{1-1/k}\rho^{1/k})$.

Another interesting extension is to graphs with small growth rate. Following the definition in [26], the growth rate of an unweighted graph is defined to be $\max_{v, r > 1} \frac{\log |B_r(v)|}{\log r}$, where $B_r(v)$ denote the set of vertices in $G$ that is within distance $r$ from the vertex $v$. So, if a graph has growth rate $k$, then for any vertex $v$ and any $r > 1$, $|B_r(v)| \leq r^k$. We have the following extension to graphs with restricted growth-rate. Again, the proof is similar and omitted.

**Theorem 7.** For an unweighted graph with growth rate $k$, the load-balancing ratio of optimal $c$-short routing is $O((n/c)^{1-1/k})$. The load-balancing ratio of (any) shortest path routing is $O(n^{1-1/k})$.
3.2 Algorithm for \(c\)-short load-balanced routing

In the above, we showed a worst case tight bound on the load balancing ratio for the optimal \(c\)-short routing algorithm. Here, we discuss how to compute \(c\)-short paths that approximate the optimal solution.

When \(c=1\), the routing is the shortest path routing, which can be done efficiently. As remarked before, the bound in Theorem 3 holds for any shortest path if there are multiple shortest paths between two points.

As for \(c\)-short path routing, the upper bound in Theorem 2 only holds for the optimal \(c\)-short routing algorithm. For the off-line case, one general approach to obtain an approximation by the randomized rounding technique [30, 29]. But that technique cannot be directly applied to our case because of the restriction on the stretch factor. We show that the on-line algorithm by Aspnes et al. [2] for virtual-circuit routing can be adapted to obtain an \(O(\log n)\) competitive (and therefore approximation) ratio.

**Theorem 8.** There is a polynomial time on-line \(c\)-short routing algorithm with the 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 [2] with slight modification. In the algorithm in [2], a weight was assigned to each edge (or vertex in our case) according to the current load on the edge and the size of the request. Then each time the shortest path with respect to this weight function is found (we call it the “lightest” path, to be distinguished from the shortest path in the graph) to satisfy a request. For \(c\)-short routing, we only need to find the lightest path among all the \(c\)-short paths. This can be done by 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 at most \(L\) for \(L = 1, 2, \ldots, c \cdot d(s,t)\). The proof of \(O(\log n)\) competitive ratio follows from the same argument in [2].

We observe that the lower bound construction in [6] can be used to show that even in a mesh, a unit-disk graph of points with constant density, 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 are omitted in this abstract. \(\square\)

4 Tradeoff based on average density

In this section, we 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 9.** Given a set of \(n\) nodes \(S\) in the plane with average density \(\hat{\rho}\), for any set of requests \(R\), \(\ell^*(R)/\ell^* = O(\min(\sqrt{\hat{\rho}} \log n, n))\). In the worst case, \(\ell^*(R)/\ell^* = \Omega(\sqrt{\hat{\rho} n / \log c})\).

**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. The only difference is that from the average density, by Lemma 1, we can only bound \(|\bigcup_{r_j \in R_k} A_j| = O(\sqrt{\hat{\rho} m^2})\). Since each node has load at most \(\ell^*\), we have that

\[
2^{k-1} \sum_{r_j \in R_k} \ell_j \leq c_0 \sqrt{\hat{\rho} m^2}^{k-1} \ell^*,
\]
for some constant $c_0 > 0$.

Thus $\sum_{r_j \in R_k} \ell_j \leq c_0 \sqrt{ml^*}$. By summing up for all the $k$'s, we have that $\sum_{r_j \in R} \ell_j \leq c_1 \sqrt{ml^*} \log n$, for $c_1 = 2c_0$.

As for the lower bound, consider the example shown in Figure 2. 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 $k$ points on each of those line segments evenly with $k$ determined later. Symmetrically, we place points with respect to the point $v$. Label those points on the outside bars $p_1, \ldots, p_k$ and $q_1, \ldots, q_k$, respectively, and those points on the bar closest to $u, v$ to be $u_1, \ldots, u_k$, and $v_1, \ldots, v_k$, respectively. Again, we place points 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 points $u, v$, 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 point. Thus, the load-balancing ratio of any $c$-short routing algorithm is $\Omega(k)$. The total number of points 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,

$$\sum_{x} \rho(x) = \Theta(\sum_{h=1}^{c} \frac{k^2}{h}) = \Theta(k^2 \log c).$$

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

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

If we set $k = \Theta(\sqrt{mn / \log c})$ with proper constant, the average density is upper bounded by $\bar{\rho}$, and the load-balancing ratio is $\Omega(\sqrt{mn / \log c})$.

## 5 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 [20, 14, 25]. A sub-graph $G$ of a unit-disc

![Figure 2. Lower bound of the load-balancing ratio for $c$-short routing with average density $\bar{\rho}$.](image)
graph $U(S)$ is said to be a $c$-spanner if the shortest path between any two points in $G$ is $c$-short. 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 $\rho$, and $G$ is a $c$-spanner on $S$, for any requests $R$, $\ell^*_c(R)/\ell^* = O(\rho c^2)$, where $\ell^*_c(R)$ is the maximum load resulted by the optimal load-balancing routing algorithm on $G$. The bound is tight in the worst case.

**Proof:** For a set of requests $R$, consider the optimal solution $P^*$ on the unit-disk graph $U$. We now construct a solution on $G$ from $P^*$. For an edge $uv$ on a path in $P^*$, 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 $P'$ in $G$ that satisfy $R$. Now, consider a point $p \in S$. A packet can be rerouted 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 3](image_url)

**Figure 3.** Lower bound $\Omega(c^2)$ on the competitive ratio on $c$-spanners.

As for the lower bound, we use the classic H-tree construction [27]. We only show the construction for constant density. The extension to arbitrary density is easy – we just put $\rho$ copies on each node. Consider $\Theta(c^2)$ points positioned on a grid shown in Figure 3. 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 $o$. 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 the little arrow in Figure 3. 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 $\Omega(c^2)$.

### 6 Conclusion

In this paper, we study the tradeoff between two important quality measures of routing algorithms for wireless 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. In practice, the shortest path routing seems to perform well in terms of balancing the load in spite of the $\Omega(\sqrt{n})$ lower bound as shown in our paper. This is probably due to that the worst case traffic pattern never happens in practice. It would be interesting to study the tradeoff under some reasonable traffic model.

References


