Bounding the Locality of Distributed Routing Algorithms

[Extended Abstract]

Prosenjit Bose
School of Computer Science
Carleton University
Ottawa, Canada
jit@scs.carleton.ca

Paz Carmi
Dept. of Computer Science
Ben-Gurion Univ. of the Negev
Beer-Sheva, Israel
carmip@cs.bgu.ac.il

Stephane Durocher
Dept. of Computer Science
University of Manitoba
Winnipeg, Canada
durocher@cs.umanitoba.ca

ABSTRACT

We examine bounds on the locality of routing. A local routing algorithm makes a sequence of distributed forwarding decisions, each of which is made using only local information. Specifically, in addition to knowing the node for which a message is destined, an intermediate node might also know a) the subgraph corresponding to all network nodes within k hops of itself, for some value of k, b) the node from which the message originated, and c) which of its neighbours last forwarded the message. Our objective is to determine which of these parameters are necessary and/or sufficient to permit local routing as k varies on a network modelled by a connected undirected graph. In particular, we establish tight bounds on k for the feasibility of deterministic k-local routing for various combinations of these parameters, as well as corresponding bounds on dilation (the worst-case ratio of actual route length to shortest path length).

Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Computer-Communication Networks—*Network Architecture and Design*; F.2.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*; G.2.2 [**Mathematics of Computing**]: Discrete Mathematics—*Graph Theory*

General Terms

Algorithms, Theory

Keywords

distributed algorithms, local routing, dilation

1. INTRODUCTION

Local Routing. Unicast communication in a network is achieved by a routing algorithm that computes a sequence of forwarding decisions that determine the route followed by a message (e.g., a packet) as it travels to its destination. In many settings, centralized routing algorithms or, more generally, routing algorithms that require complete knowledge of the network topology are impractical; reasons include that the network is too large, that the topology of the entire graph is unknown, or that the network changes dynamically [13]. Alternatively, a *local* routing algorithm makes a series of distributed forwarding decisions, computed at each of the intermediate nodes along the route. When a node receives a message, it selects a port (i.e., one of its neighbours) to which to forward the message using only local information. In particular, each node is only aware of the subset of the network consisting of nodes within k hops from itself, for some k. Consequently, the route cannot be precomputed entirely in general. Furthermore, message overhead and local memory are often limited [11]. In particular, a network node cannot be expected to maintain a history of messages that have passed through it (i.e., the network is *memoryless*). Similarly, the message overhead cannot store the set of nodes visited by the message (i.e., the routing algorithm is *stateless*).

Although a straightforward flooding algorithm is possible, such a strategy has obvious drawbacks, including high traffic loads [13], cyclic behaviour (if the network is memoryless), and requiring knowledge of an upper bound on the diameter of the network to ensure both termination and successful delivery. In this paper we consider single-path deterministic routing algorithms.

We represent a network by a connected, unweighted, undirected graph $G = (V, E)$ with unique vertex labels. A network node (graph vertex) is identified by its label. In some networks, a node's label may provide information about its neighbourhood in the network (e.g., a grid graph node can be labelled by its grid coordinates). In general, suppose that the vertex labelling is independent of the graph; that is, a node's label does not encode additional information about the topology of the graph or the node's neighbourhood. Equivalently, we consider routing algorithms that succeed on any permutation of the vertex labels of G. We assume that every node knows its own label as well as the labels of its neighbours. A message also requires a destination node, identified by the node's label. Some or all of the following additional information may be available to an intermediate node u to compute the next node to which a message should be forwarded:

1. **origin-awareness**: knowledge of the node from which the message originated,

$T(n)$	origin-aware	origin-oblivious
predecessor-aware	$n/4$	$n/3$
predecessor-oblivious	$n/2$	$n/2$

Table 1: Main result: there exists a k-local routing algorithm when $k \geq T(n)$, but no k-local routing algorithm exists when $k < T(n)$, where n denotes the number of network nodes.

2. **predecessor-awareness**: knowledge of the incoming edge (port) along which the message was forwarded to u (equivalently, the neighbour of u that last forwarded the message), and
3. ***k*-locality**: knowledge of the k-neighbourhood of u (i.e., the subgraph of G consisting of all paths rooted at u with length at most k).

Our objective is to determine which of these parameters are necessary and/or sufficient to permit local routing as k varies.

Overview of Results. We identify tight bounds on the value of the locality parameter k for the feasibility of k-local routing in each of the four combinations of constraints: predecessor-aware or predecessor-oblivious, and origin-aware or origin-oblivious. In each case, let $T(n)$ denote the corresponding threshold. That is, for every $k < T(n)$, every k-local routing algorithm is defeated by some connected graph on n vertices. Similarly, for every $k \geq T(n)$, there exists a k-local routing algorithm that succeeds on all connected graphs on n vertices. Our main result is the identification of the values of $T(n)$; see Table 1. In addition, we establish a lower bound of $S(k) = 2-3k/n$ on the worst-case dilation of any k-local routing algorithm and show this bound is tight for three of the four combinations of constraints.

2. MODELLING LOCAL ROUTING

In this section we formalize our model for local routing.

***k*-Local Routing Functions.** Given a graph $G = (V, E)$, we employ standard graph-theoretic notation, where for each vertex $v \in V$, $\text{Adj}(v) = \{u \mid \{u, v\} \in E\}$ denotes the set of vertices adjacent to v and $\deg(v) = |\text{Adj}(v)|$ denotes its degree. Let $\text{dist}(u, v)$ denote the (unweighted) graph distance between vertices u and v.

The *k-neighbourhood* of a vertex $v \in V$, denoted $G_k(v)$, is the subgraph of G that contains all paths rooted at v with length at most k. A routing algorithm is *origin-aware*, *predecessor-aware*, and *k-local* if it can be defined as a function $f(s, t, u, v, G_k(u))$, where

- $s \in V$ is the origin node,
- $t \in V$ is the destination node,
- the message is currently at node $u \in V$,
- node u received the message from its neighbour $v \in \text{Adj}(u)$,
- $G_k(u)$ is the k-neighbourhood of node u, and
- $f(s, t, u, v, G_k(u))$ returns the neighbour of u to which the message should be forwarded (i.e., the port to which node u must forward the packet).

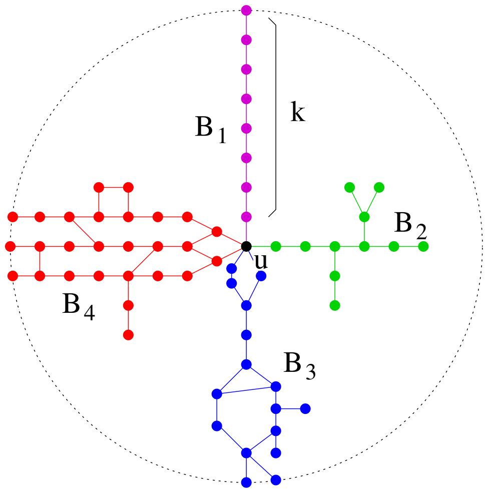

Figure 1: In this example, $G_8(u)$ consists of four local components, corresponding to the four connected components of $G_8(u)\setminus\{u\}$. B_1, B_3, and B_4 are frontier components but B_2 is not. B_1 and B_3 are constrained frontier components but B_2 and B_4 are not.

Say $v = \varnothing$ if the message has not yet been forwarded (i.e, the message leaves node s for the first time). Every k-local routing algorithm $\mathcal{A}$ has a corresponding routing function f. A sequence of calls to function f returns a sequence of forwarding decisions that corresponds to a walk through G originating at s (i.e., the route). We consider two constraints on k-local routing algorithms: an *origin-oblivious* k-local routing algorithm is not provided the parameter s, and a *predecessor-oblivious* k-local routing algorithm is not provided the parameter v. To simplify notation for predecessor-aware algorithms, let $f_u(v)$ denote the *local routing function* at node u for a given s, t, u, and $G_k(u)$, where $f_u(v) = f(s, t, u, v, G_k(u))$. That is, $f_u(v)$ returns the neighbour of u to which the message is forwarded as a function of the neighbour v from which it is received.

Naturally, not all routing functions can be implemented efficiently as local routing algorithms. The routing function model allows stronger negative results to be established for a more general class of routing algorithms, regardless of implementation concerns. With respect to positive results, the routing algorithms we present can be implemented efficiently locally; implementation details are not the focus of this paper.

Let C denote a connected component of $G_k(u) \setminus \{u\}$. We refer to C as a *local component* of u. If $v \in C \cap \text{Adj}(u)$, then we say C is rooted at v (C can have multiple roots). If C contains a vertex v such that $\text{dist}(u, v) = k$, then (relative to u) C is a *frontier component*, v is a *frontier vertex*, and a shortest path from u to v is a *frontier path*. In other words, C extends to the limit of u's knowledge: node v may have neighbours outside C, but this information is not known locally at u. If C is a frontier component of u and every frontier path in C passes through some vertex w, then C is a *constrained frontier component* and w is a *constraint vertex*. See Figure 1.

Evaluating Routing Algorithms. A routing algorithm $\mathcal{A}$ defined by a routing function f *succeeds* (synonymously, *guarantees delivery*) if for all graphs G and all origin-destination pairs (s, t) in G, the sequence of values returned by f corresponds to a walk from s to t in G. Otherwise, $\mathcal{A}$ is *defeated* by some graph G and some pair (s, t) in G. A routing algorithm $\mathcal{A}$ has *dilation* bounded by δ if for all graphs G and all origin-destination pairs (s, t) in G, $r_{\mathcal{A}}(s, t)/\operatorname{dist}(s, t) \leq \delta$, where $r_{\mathcal{A}}(s, t)$ denotes the length of the route from s to t returned by $\mathcal{A}$.

3. RELATED WORK

In *position-based* routing, network nodes are embedded in some space (typically $\mathbb{R}^2$ or $\mathbb{R}^3$) and each node knows its spatial coordinates (i.e., nodes are *location-aware*). Position-based routing is also known as geo-routing, geographic routing, or geometric routing. Many recent results related to local routing are position based. We briefly describe some of these related results and discuss the interdependence between position-based and *position-oblivious* routing.

Greedy routing [7] (forward the message to the neighbour closest to the destination), compass routing [11] (forward the message along the edge that forms the smallest angle with the line segment to the destination), and greedy-compass routing [1] (apply greedy routing to the two edges adjacent to the line segment to the destination) are three well-known position-based routing algorithms, each of which succeeds on specific classes of graphs but is defeated by some planar graph [2]. All three algorithms are predecessor-oblivious, origin-oblivious, and 1-local.

To show that a routing algorithm fails on some class of graphs $\mathcal{G}$, it suffices to identify a graph in $\mathcal{G}$ on which the algorithm cycles infinitely without reaching the destination. Stronger negative results are those that apply to all routing algorithms, showing that no routing algorithm succeeds on a given class of graphs. Bose et al. [1] show that every position-based, predecessor-oblivious, origin-oblivious, 1-local routing algorithm is defeated by some convex subdivision.

Face routing [11] was one of the first position-based 1-local routing algorithm discovered to succeed on more general classes of graphs embedded in the plane. In brief, face routing forwards the message in a clockwise direction along the edges of a face, and along the sequence of faces that intersect the line segment between the origin and destination nodes. Forward progress is guaranteed by storing a parameter such as the furthest intersection of the line segment with a visited face. As such, face routing is not stateless since it requires $\Theta(\log n)$ bits to be stored with the message. Face routing succeeds on planar graphs [11], on unit disc graphs [3], and on d-quasi unit disc graphs for any $d \in [1/\sqrt{2}, 1]$ [12]. See [3] and [12] for definitions of unit disc graphs and quasi unit disc graphs, respectively. Fraser considers a generalization of face routing to graphs embedded on tori [9].

Although our discussion focuses on deterministic routing algorithms, we briefly note that randomized solutions permit k-local routing on more general classes of graphs. Flury and Watterhofer consider the problem of randomized local routing on unit ball graphs [8] and show that any randomized position-based local routing algorithm has expected route length $\Omega(l^3)$, where l denotes the length of the shortest path.

Durocher et al. [6] show that for every fixed k, every origin-aware, predecessor-aware, k-local routing algorithm fails on some unit ball graph. The proof has two parts. First, the corresponding position-oblivious result is proven: for every fixed k, every origin-aware, predecessor-aware, k-local routing algorithm fails on some graph. Next, a k-local reduction from (unembedded) graphs to unit ball graphs is used to show that if some (possibly position-based) k-local routing algorithm succeeds on unit ball graphs, then some (position-oblivious) k-local routing algorithm succeeds on all graphs. This interdependence between position-based and position-oblivious routing algorithms motivates the question of exploring the boundary between feasibility and impossibility of local routing algorithms as a function of the local information available. In this paper we consider the position-oblivious case.

4. WHEN LOCAL ROUTING IS IMPOSSIBLE: NEGATIVE RESULTS

In this section we present negative results: every k-local routing algorithm fails on some graph when the degree of locality k is less than the given bound. For each combination of origin-awareness/obliviousness and predecessor-awareness/obliviousness, we demonstrate a counter-example consisting of a set of graphs such that any k-local routing algorithm fails on at least one of the graphs in the set.

4.1 Properties of Local Routing Functions

The proofs of Theorems 3 through 6 refer to Lemma 1 and Corollary 2, which generalize an observation of Durocher et al. [6] showing that if a k-local routing algorithm guarantees delivery, then each local routing function corresponds to a circular permutation (under certain conditions). Recall that a circular permutation of n distinct elements is an ordering of these elements in a cycle.

LEMMA 1. *Let u denote a node such that*

1. $\deg(u) \geq 2$,
2. *for all $\{a, b\} \subseteq \operatorname{Adj}(u)$, a and b belong to different frontier components of u, and*
3. *neither the origin node s nor the destination node t is in $G_k(u)$.*

If $\mathcal{A}$ is an origin-aware, predecessor-aware, k-local routing algorithm that guarantees delivery, then the local routing function of $\mathcal{A}$ at u is a circular permutation of $\operatorname{Adj}(u)$.

PROOF. Choose any $k \geq 1$, any node u, and any k-neighbourhood $G_k(u)$ such that Properties 1 through 3 hold. Suppose $\mathcal{A}$ is any k-local routing algorithm that guarantees delivery and that the local routing function f_u is not a circular permutation.

Case 1. Suppose f_u is not a permutation. That is, f_u is not surjective. Therefore, there exists some $v \in \operatorname{Adj}(u)$ such that for all $w \in \operatorname{Adj}(u)$, $f_u(w) \neq v$. Let B_1 denote the local component of u that contains v and let B_2 denote any other local component of u. By Property 2, each local component of u extends to the frontier of $G_k(u)$. Let G denote a graph that contains $G_k(u)$ such that node t has degree one and is the only node adjacent to B_1 outside $G_k(u)$. Similarly, let node s have degree one such that it is the only node adjacent to B_2 outside $G_k(u)$. See Figure 2. Since for all $w \in \operatorname{Adj}(u)$, $f_u(w) \neq v$, the message will never enter B_1 and, consequently, will never reach t. Therefore, algorithm $\mathcal{A}$ fails on graph G, deriving a contradiction.

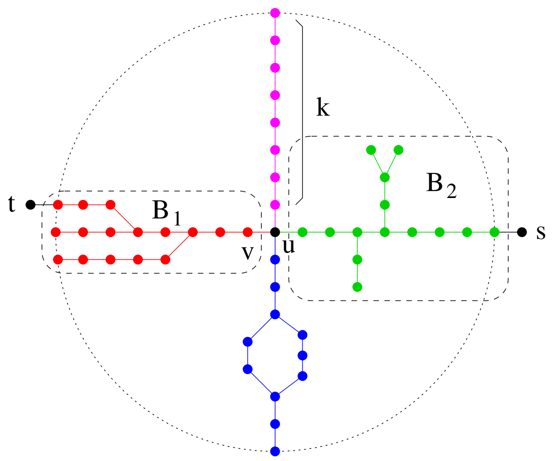

Figure 2: $G_k(u)$ consists of frontier components, each of which is rooted at a unique neighbour of u. This example illustrates the graph constructed in Case 1 for a given $G_k(u)$ when $k = 8$.

Case 2. Suppose f_u is a permutation but not a derangement (a derangement is a complete permutation). Therefore, $f_u(v) = v$ for some $v \in \text{Adj}(u)$. Let G be a graph as defined in Case 1, with the exception that nodes s and t are interchanged. It follows that the message will never enter any local component other than B_1 and, consequently, will never reach t. Therefore, algorithm $\mathcal{A}$ fails on graph G, deriving a contradiction.

Case 3. Suppose f_u is a derangement but not a circular permutation. Therefore, f_u cannot be expressed as a single permutation cycle. Let $(a_1 \ldots a_k)$ and $(b_1 \ldots b_j)$ denote any two permutation cycles of f_u. Observe that $\{a_1, \ldots a_k\}$ and $\{b_1, \ldots, b_j\}$ are disjoint subsets of $\text{Adj}(u)$. Let G be a graph as defined in Case 1, with the exception that node s is adjacent to a local component B_1 rooted at a node in $\{a_1, \ldots a_k\}$ and t is adjacent to a local component B_2 rooted at a node in $\{b_1, \ldots, b_j\}$. It follows that the message will never enter B_2 and, consequently, will never reach t. Therefore, algorithm $\mathcal{A}$ fails on graph G, deriving a contradiction.

All three cases derive a contradiction and our assumption must be false. Therefore, the local routing function f_u must be a circular permutation. □

In other words, without additional information on which to base a local routing decision, an intermediate node u must try all possibilities and sequentially forward the message to each of its neighbours. When node u has degree two, only one circular permutation is possible: a message received from one neighbour of u must be forwarded to the opposite neighbour. If node u has degree j, then $(j-1)!$ circular permutations are possible. If routing algorithm $\mathcal{A}$ is origin oblivious, then Lemma 1 gives:

Corollary 2. *Let u denote a node such that*

1. $\deg(u) \geq 2$,
2. *for all $\{a, b\} \subseteq \text{Adj}(u)$, a and b belong to different frontier components of u, and*
3. *the destination node t is not in $G_k(u)$.*

If $\mathcal{A}$ is an origin-oblivious, predecessor-aware, k-local routing algorithm that guarantees delivery, then the local routing function of $\mathcal{A}$ at u is a circular permutation on $\text{Adj}(u)$.

routing strategy	circular permutation	succeeds	fails
1	$(P_1P_2P_3P_4)$	G_1, G_3	G_2
2	$(P_1P_2P_4P_3)$	G_1, G_2	G_3
3	$(P_1P_3P_2P_4)$	G_2, G_3	G_1
4	$(P_1P_3P_4P_2)$	G_1, G_2	G_3
5	$(P_1P_4P_2P_3)$	G_2, G_3	G_1
6	$(P_1P_4P_3P_2)$	G_1, G_3	G_2

Table 2: Each routing strategy corresponds to a circular permutation of the neighbours of u.

Proof. Given any node u, any $k \geq 1$, and any k-neighbourhood $G_k(u)$, if Properties 1 through 3 hold (as defined in Lemma 1), then the local routing function f_u is a circular permutation by Lemma 1. Since $\mathcal{A}$ is origin oblivious, function f_u remains constant for any given $G_k(u)$ and t, regardless of s. In particular, f_u is a circular permutation regardless of whether or not s is contained in $G_k(u)$. The result follows. □

Theorems 3 through 6 establish lower bounds corresponding to each of the four combinations of k-local routing algorithms: origin-aware/oblivious and predecessor-aware/oblivious.

4.2 Predecessor Aware and Origin Aware

Theorem 3. *For every $k < \lfloor (n+1)/4 \rfloor$, every origin-aware, predecessor-aware, k-local routing algorithm fails on some connected graph.*

Proof. Choose any $k < \lfloor (n+1)/4 \rfloor$, $k \in \mathbb{Z}^+$. Therefore, $k \in \{1, \ldots, r\}$, where $r = \lfloor (n-3)/4 \rfloor$. Let G_1, G_2, and G_3 denote the graphs illustrated in Figure 3, such that each path P_1 through P_4 consists of r vertices that are labelled consistently relative to node u in all three graphs. In each graph, $G_k(u)$ is a tree consisting of four paths of length k rooted at u, none of which contain s nor t. In addition to the $4r$ nodes in paths P_1 through P_4, each graph includes nodes u, s, and t. Depending on the value of $n \bmod 4$, between zero and three extra nodes remain; these are added between s and P_1 to bring the total number of nodes to n. Any successful routing algorithm must pass the message across P_1 to node u. Since u has degree four, its local routing function is one of six possible circular permutations by Lemma 1. The remaining nodes have degree at most two. Therefore, when the message is passed to a node on a path that does not contain s nor t, by Lemma 1, the message must continue forward until it returns again to u. As shown in Table 2, for each of the six possible routing strategies, the message never enters the path containing t in at least one of the graphs G_1, G_2, or G_3. That is, for every routing strategy $\mathcal{A}$, there exists a graph on which $\mathcal{A}$ fails. □

4.3 Predecessor Aware and Origin Oblivious

Using an argument similar to the proof of Theorem 3, we now show that the lower bound on the locality parameter k increases to $\lfloor (n+1)/3 \rfloor$ for origin-oblivious k-local routing algorithms:

Theorem 4. *For every $k < \lfloor (n+1)/3 \rfloor$, every origin-oblivious, predecessor-aware, k-local routing algorithm fails on some connected graph.*

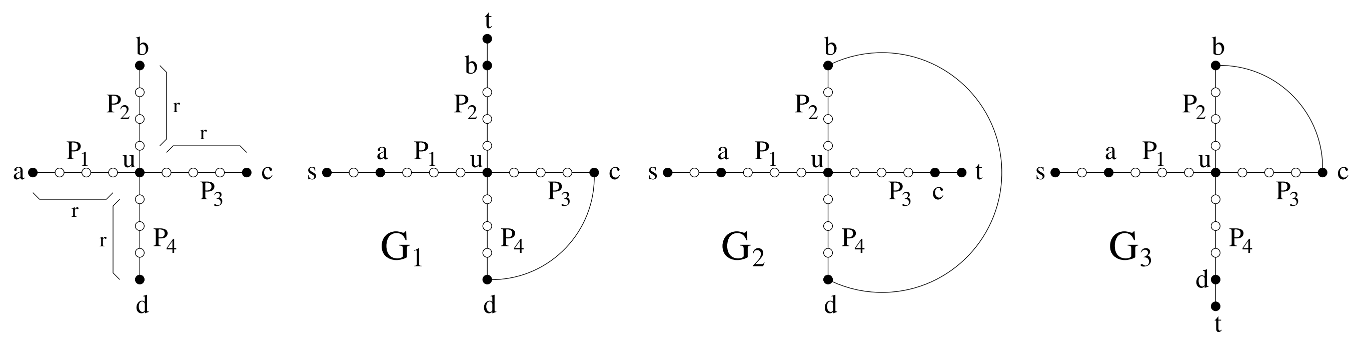

Figure 3: The k-neighbourhood $G_k(u)$ is identical in graphs G_1, G_2, and G_3. In this example, suppose $n \bmod 4 = 0$. Consequently, one extra node is added between s and P_1 such that the total number of nodes is n.

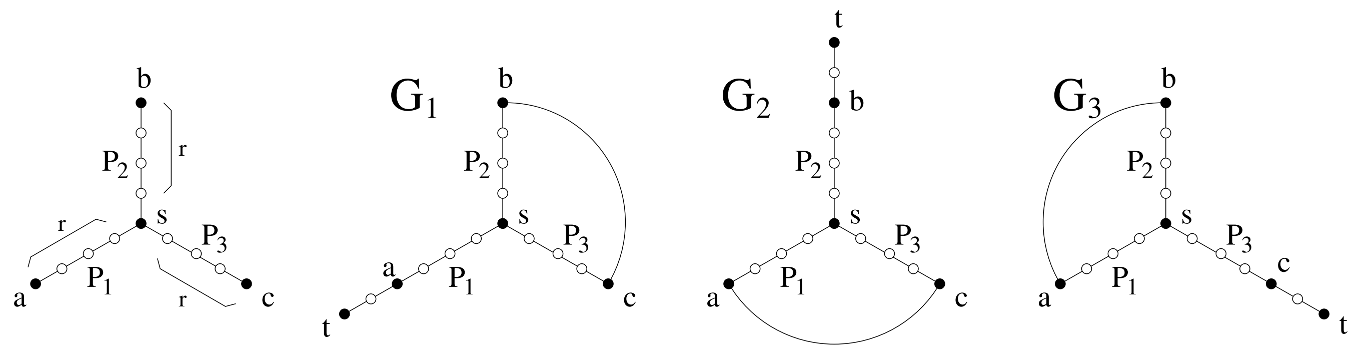

Figure 4: The k-neighbourhood $G_k(s)$ is identical in graphs G_1, G_2, and G_3. In this example, suppose $n \bmod 3 = 0$. Consequently, one extra node is added next to t such that the total number of nodes is n.

Proof. Choose any $k < \lfloor (n+1)/3 \rfloor$, $k \in \mathbb{Z}^+$. Therefore, $k \in \{1, \ldots, r\}$, where $r = \lfloor (n-2)/3 \rfloor\}$. Let G_1, G_2, and G_3 denote the graphs illustrated in Figure 4, such that each path P_1 through P_3 consists of r vertices that are labelled consistently relative to node s in all three graphs. In each graph, $G_k(s)$ is a tree consisting of three paths of length k rooted at s, none of which contain t. In addition to the 3r nodes in paths P_1 through P_3, each graph includes nodes s and t. Depending on the value of $n \bmod 3$, between zero and two extra nodes remain; these are added between t and the corresponding path P_i nearest to t to bring the total number of nodes to n. Since node s has degree three, its local routing function is one of two possible circular permutations by Corollary 2. A routing strategy must specify the direction in which a message initially leaves node s (three directions are possible). The remaining nodes have degree at most two. Therefore, when the message is passed to a node on a path that does not contain t, by Corollary 2, the message must continue forward until it returns again to node s. As shown in Table 3, for each of the six possible routing strategies, the message never enters the path containing t in at least one of the graphs G_1, G_2, or G_3. That is, for every routing strategy $\mathcal{A}$, there exists a graph on which $\mathcal{A}$ fails. □

4.4 Predecessor Oblivious and Origin Aware

When knowledge of the predecessor node is withheld, the lower bound on the locality parameter k increases to $\lfloor n/2 \rfloor$ for predecessor-oblivious k-local routing algorithms:

Theorem 5. *For every $k < \lfloor n/2 \rfloor$, every origin-aware, predecessor-oblivious, k-local routing algorithm fails on some connected graph.*

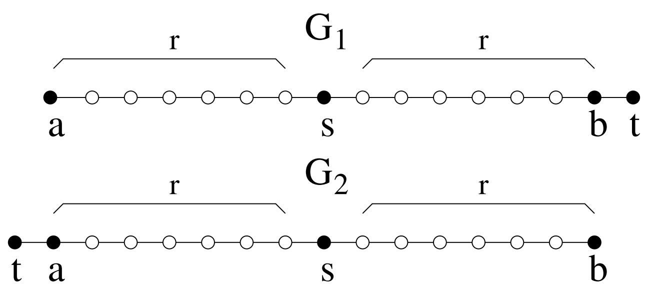

Figure 5: illustration in support of Theorem 5

Proof. Choose any $k < \lfloor n/2 \rfloor$, $k \in \mathbb{Z}^+$. Therefore, $k \in \{1, \ldots, r\}$, where $r = \lfloor n/2 \rfloor - 1$. Let G_1 denote a path of n vertices with the origin node s located at the $(r+1)$st vertex and the destination node t located at the far end. Let G_2 denote the analogous graph upon moving node t to the opposite end of the path. Let the remaining nodes be labelled consistently relative to node s in both graphs. See Figure 5. The k-neighbourhood $G_k(s)$ is identical in G_1 and G_2. If algorithm $\mathcal{A}$ sends the message right at s, then $\mathcal{A}$ fails on graph G_2 since it must eventually send the message left, at which point its behaviour becomes cyclic. Similarly, if algorithm $\mathcal{A}$ sends the message left at s, then it fails on graph G_1. □

routing strategy	circular permutation	initial direction	succeeds	fails
1	$(P_1P_2P_3)$	toward a	G_1, G_3	G_2
2	$(P_1P_2P_3)$	toward b	G_1, G_2	G_3
3	$(P_1P_2P_3)$	toward c	G_2, G_3	G_1
4	$(P_1P_3P_2)$	toward a	G_1, G_2	G_3
5	$(P_1P_3P_2)$	toward b	G_2, G_3	G_1
6	$(P_1P_3P_2)$	toward c	G_1, G_3	G_2

Table 3: Each routing strategy corresponds to a circular permutation of the neighbours of u paired with an initial direction.

4.5 Predecessor Oblivious and Origin Oblivious

Finally, if we further constrain the knowledge available to intermediate nodes, then the lower bound on the locality parameter k does not increase:

THEOREM 6. *For every $k < \lfloor n/2 \rfloor$, every origin-oblivious, predecessor-oblivious, k-local routing algorithm fails on some connected graph.*

PROOF. The lower bound of $\lfloor n/2 \rfloor$ follows by Theorem 5. □

4.6 Dilation

We now consider lower bounds on dilation for k-local routing algorithms.

THEOREM 7. *For any $k < n/2$, no k-local routing algorithm can guarantee dilation less than*

$$\frac{2n-3k-1}{k+1}. \tag{1}$$

PROOF. Choose any n and any $k < n/2$. Let P denote the set of paths of length n with vertex set $\{s, t, v_1, \ldots, v_{n-2}\}$. Choose any k-local routing $\mathcal{A}$ that succeeds on all paths in P. Suppose the origin and destination nodes are labelled s and t, respectively. Let $d_P(i) = \max\{\text{dist}(s, v) \mid v \in V_i\}$, where V_i denotes the set of vertices of P to which algorithm $\mathcal{A}$ has passed the message during the first i steps. Thus, $d_P(i) \le i$. Choose any $i \in \{0, \ldots, n-2k-2\}$. Observe that there exist paths P_1 and P_2 in P such that for all $v \in V_i$, a) $G_k(v)$ is identical in P_1 and P_2, b) $G_k(v)$ has two frontier components, neither of which contains t, and c) t lies at opposite endpoints of paths P_1 and P_2 relative to s. The nodes of V_i correspond to a subpath containing at most $i+1$ nodes; each endpoint of the subpath has an unexplored frontier component (k nodes each) and node t remains out of sight. Summing these gives $i + 1 + 2k + 1 \le n$ nodes.

When the message reaches the corresponding node at distance $d_{P_1}(i) = d_{P_2}(i)$, $\mathcal{A}$ sends the message in the same direction in P_1 and P_2, i.e., toward t in one path and away from t in the other. In the latter case, $\mathcal{A}$ must eventually pass the message back to s before it can reach t. Returning the message to s requires at least $i+2$ steps and reaching t requires at least $k+1$ additional steps. Therefore, the total number of steps is at least $2i + k + 3$. This gives $r_{\mathcal{A}}(s,t) \ge 2n - 3k - 1$ when $i = n - 2k - 2$. Since $\text{dist}(s,t)$ can be as little as $k+1$, the results follows. □

The bound on dilation (1) is perhaps more clearly expressed by taking the limit as the number of nodes approaches infinity (and $k = c \cdot n$ for some constant c). We denote this limit by $S(k)$:

$$S(k) = \lim_{n\to\infty} \frac{2n-3k-1}{k+1} = \frac{2n}{k} - 3.$$

Of particular interest are the values of $k \in \{n/4, n/3, n/2\}$, for which the corresponding bounds on dilation are 5 (when $k = n/4$), 3 (when $k = n/3$), and 1 (when $k \to n/2$). These bounds are tight for $k = n/3$ and $k = n/2$ as shown in Theorems 14, 16, and 17.

5. WHEN LOCAL ROUTING IS POSSIBLE: ROUTING STRATEGIES

In this section we present positive results: there exists a successful k-local routing algorithm when the degree of locality k exceeds the given bound. We describe a k-local routing algorithm for each combination of origin-awareness/obliviousness and predecessor-awareness/obliviousness.

5.1 Predecessor Aware and Origin Aware

We describe an $(n/4)$-local predecessor-aware and origin-aware routing algorithm. Given any $k \ge n/4$, the algorithm begins with a k-local preprocessing step to identify the edges on which routing takes place. Every node $u \in G$ builds a subgraph of $G_k(u)$ by deleting an edge from every cycle that contains u in $G_k(u)$. The order in which the cycles are broken is important. By our assumption that nodes have unique labels, we obtain an ordering on the edges of G. In particular, any set of edges has a minimum edge. The node u considers every cycle that contains it in $G_k(u)$ and deletes the minimum edge e among all cycles. See Figure 6. The node u then computes all remaining cycles in $G_k(u) \setminus \{e\}$ that contain u and deletes the minimum edge among these cycles. This process is repeated until no cycles containing u remain. The final graph denoted $G'_k(u)$ is a spanning subgraph of $G_k(u)$ in which u is a cut vertex.

Consider the local components in $G'_k(u)$. The edges in $G'_k(u)$ joining u to a local component are called *routing edges*. If a local component in $G'_k(u)$ is a frontier component, the component is called *active*. Otherwise, both the component and routing edge are called *passive*. The number of active edges adjacent to a vertex is its *active degree*. Since an active edge joins u to a component with at least $n/4$ vertices, a node can have active degree at most 3. An edge $\{a, b\}$ is called *consistent* provided that both a and b consider the edge between them to be active. A node is called *consistent* provided that all of its active edges are consistent. Otherwise, the node is called *inconsistent*

OBSERVATION 8. *An inconsistent node has active degree 1.*

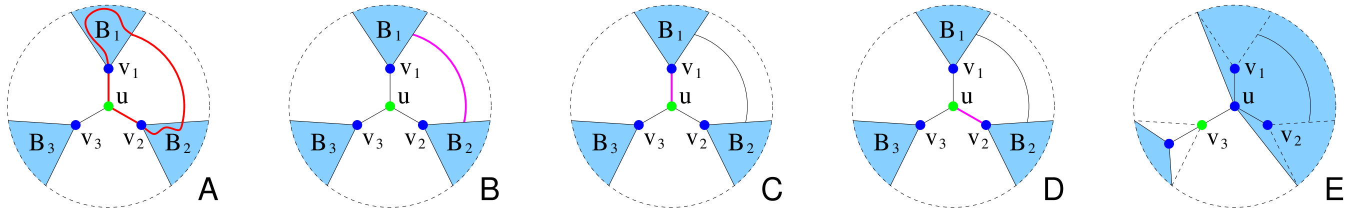

Figure 6: k-local preprocessing. Suppose the current node u has three neighbours, v_1 through v_3, and $G_k(u)$ contains a cycle that includes vertices v_1, u, and v_2 (A). The preprocessing step classifies one of the edges on the cycle as a non-routing edge (magenta). The selected edge may be distant from u (B) or adjacent to u (C and D). The choice of routing edge does not affect nodes not on the cycle (e.g., v_3) since the edges of the cycle all lie in the same local component from the point of view of the corresponding vertex (E); in particular, none of the cycle's edges are adjacent to v_3.

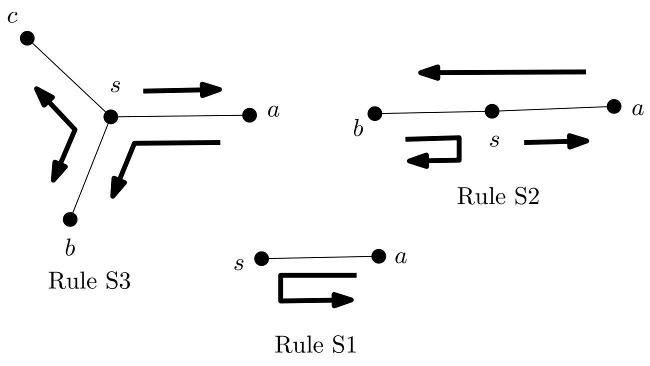

Figure 7: Algorithm 1, Case 2: Message received/forwarded by s.

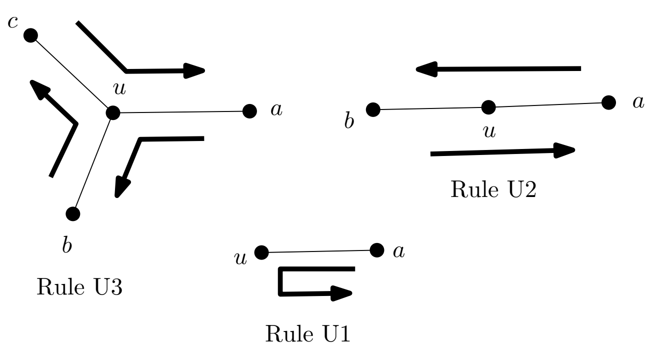

Figure 8: Algorithm 1, Case 3: Message received/forwarded by a node u that cannot see t and can possibly see s in an active component.

Once each node has identified its routing edges, a simple set of rules determine the forwarding decisions. The success of the routing algorithm relies on the property that each node has active degree at most 3. Notice that the lower bound argument of the proof of Theorem 3 consists of graphs that have one node with active degree 4; this cannot occur when $k > n/4$. The routing algorithm consists of four cases outlined below.

Algorithm 1: $(n/4)$-local, origin-aware, predecessor-aware routing algorithm.
Case 1. Suppose $\text{dist}(u,t) \le k$. That is, $t \in G_k(u)$. The algorithm forwards the message to any neighbour of u on a shortest path from u to t until the message arrives at t.
Case 2. Suppose $\text{dist}(u,t) > k$ and $u = s$. Forwarding decisions are illustrated in Figure 7.
Case 3. Suppose $\text{dist}(u,t) > k$, $u \ne s$, and either $\text{dist}(u,s) > k$ or s is in an active component of u. Forwarding decisions are illustrated in Figure 8.
Case 4. Suppose $\text{dist}(u,t) > k$, $u \ne s$, and s is in a passive component of u. Forwarding decisions are illustrated in Figure 9.

LEMMA 9. *Algorithm 1 successfully delivers the message from the origin node s to the destination node t.*

PROOF. Suppose, for sake of a contradiction, that a message from s does not reach t. Since the number of nodes in G is finite, the message must visit a repeating sequence of nodes of G. Denote by $X = x_1, \ldots, x_m$ the sequence of

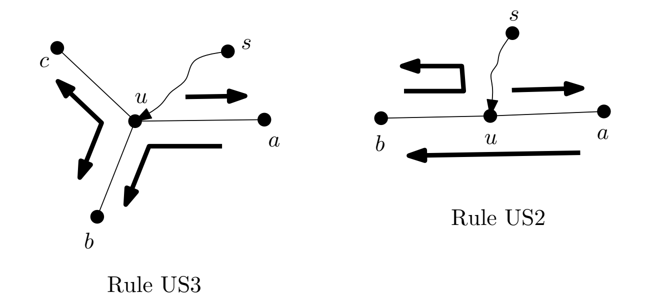

Figure 9: Algorithm 1, Case 4: Message received/forwarded by a node u that cannot see t and sees s in a passive component.

nodes visited, where $R = x_i, \ldots, x_m$ is a repeating sequence of nodes for some $1 \leq i < m$.

Claim 1: All nodes in sequence R are consistent. This follows from the property that an inconsistent node can only receive and forward a message at most once.

There is a set I of nodes in G that are *invisible* to all nodes in X. By invisible, we mean that $\forall x \in X,\ G_k(x) \cap I = \varnothing$. In particular, $t \in I$. Consider a shortest path from t to a node $v \in V \setminus I$. By construction, v must be on the frontier of some vertex $x_j \in X$, i.e., $\text{dist}(x_j, v) = k \geq n/4$. Otherwise, all neighbours of v would be visible to x_j, contradicting the fact that v is the closest visible vertex to t. Furthermore, no node on a shortest path between x_j and v can appear in X.

Since v is a frontier vertex of x_j, the frontier component of x_j containing v must be active. However, x_j never forwards the message to this component. Consequently, x_j has active degree at least 2, which by Observation 8 implies that x_j is a consistent node. Let $e = \{x_j, y\}$ denote the corresponding active edge (along which the message is not forwarded). To obtain a contradiction, we show that in all cases, x_j ultimately forwards the message along edge e.

Claim 2: $x_{j-1} \neq x_{j+1}$. Only forwarding rules S1, S2, U1, and US2 would have $x_{j-1} = x_{j+1}$. In all four cases, the message is forwarded to all active edges adjacent to x_j, contradicting the fact that v is the closest visible vertex to t.

Claims 1 and 2 imply that x_j has active degree 3. Consider the undirected subgraph S of G that consists only of the edges followed by the algorithm in the repeating sequence R. Observe that by construction, S has maximum degree 3.

Case 1. Suppose S contains no cycle. That is, S is a tree. Root this tree at x_j. When x_j forwards the message to x_{j+1}, the message is passed into the subtree of S rooted at x_{j+1}. By Claim 2, since S is a tree, all paths from a node in the subtree rooted at x_{j+1} to a node in the subtree rooted at x_{j-1} must go through x_j. This implies that at some point in the sequence R, the node x_{j+1} forwards the message to x_j, which will then forward it to y, deriving a contradiction.

Case 2. Suppose S contains a simple cycle. By Claim 1, all edges in the cycle are consistent. Every cycle of length at most $2k$ contains at least one passive edge. Therefore, the cycle must contain greater than $2k \geq n/2$ vertices.

The node x_j must lie on such a cycle, otherwise, as in Case 1, forwarding the message along x_{j+1} will cause it to return to x_j via the edge $\{x_{j+1}, x_j\}$, deriving a contradiction.

We now show that the message cannot be passed around this cycle. The key to breaking the cycle is the location of the origin node s. Consider two cases: either node s is invisible to every node in the cycle or it is visible to at least one node in the cycle.

Case 2a. Suppose s is not visible to any node in the cycle. Therefore, when s initially forwards the message, the message follows a path from s to some node on the cycle. This path contains at least $k \geq n/4$ distinct nodes that are not part of the cycle since no node of the cycle can see s. The cycle contains more than $n/2$ distinct nodes and the path from x_j to v contains at least $n/4$ nodes, none of which are visited by the sequence X. Therefore, we obtain the desired contradiction since there are exactly n nodes in G.

Case 2b. Suppose s is visible to some x_k on the cycle. Consider two further subcases.

Case 2b1. Suppose s is contained in the cycle. Node s must have active degree 2 or 3. If it has active degree 2, then Rule S2 implies that the message is sent along the cycle in two different directions, contradicting the fact that the message is not forwarded along edge $\{x_j, y\}$. If s has active degree 3, then only one of three pairs of nodes adjacent to s can be involved in the cycle. Let us examine each pair in turn. For what follows, consider the labelling given in Figures 7. The nodes a and c cannot be involved in the cycle because when the message is received by s from c, it is forwarded to b and when it is received from s by a, it is also forwarded to b. Nodes a and b cannot be involved in the cycle because s initially forwards the message to a, which implies that to complete the cycle, it must receive the message from b. However, when s receives a message from b, it is forwarded to c. Therefore, according to Rule S3, if s part of the cycle, then on the cycle, it receives the message from b and forwards it to c or vice versa. However, initially s forwards the message to a and the message returns to s. This implies that none of the nodes in the active component of s rooted at a are part of the cycle. This component has size at least $n/4$. Moreover, none of these nodes can be part of the path from x_j to v. Therefore, we again contradict the fact that G has n nodes.

Case 2b2. Suppose s is not contained in the cycle. The argument is analogous to Case 2b1 upon substituting rules US2 and US3 for rules S2 and S3, respectively. □

LEMMA 10. *Algorithm 1 has dilation at most 8.*

PROOF. If a message from s is sent to t and t is visible to s, then the path followed by Algorithm 1 has dilation 1. When t is not visible to s, then the length of the shortest path from s to t is at least $n/4 + 1$. By looking at the proof of Lemma 9, we see that if a message follows a cycle, then it visits every edge of the cycle once. Otherwise, it visits every edge at most twice. This implies that the length of the path followed by Algorithm 1 is at most $2n$. Since the length of the shortest path is at least $n/4 + 1$, we get the desired result. □

THEOREM 11. *For every $k \geq n/4$, there exists an origin-aware, predecessor-aware, k-local routing algorithm that succeeds on all connected graphs while guaranteeing dilation at most 8.*

5.2 Predecessor Aware and Origin Oblivious

We describe a $(n/3)$-local predecessor-aware and origin-oblivious routing algorithm.

Choose any integer $k \geq n/3$. Therefore, $3k+1 > n$. Since a frontier component contains at least k nodes, every node has at most two frontier components. In particular, it has at most two constrained frontier components. Let u denote the current node.

The algorithm requires a k-local preprocessing step similar to that described in Algorithm 1 to identify a set of active edges in $G_k(u)$ which we denote by $G'_k(u)$. Analogously, after applying the preprocessing step, every active neighbour of u is the root of a unique constrained component of u in $G'_k(u)$ (there can be at most two).

The following algorithm is then applied to $G'_k(u)$ to make the forwarding decision.

Algorithm 2: $(n/3)$-local, origin-oblivious, predecessor-aware routing algorithm.

Case 1. Suppose $\text{dist}(u,t) \leq k$. That is, $t \in G_k(u)$. The algorithm forwards the message to any neighbour of u on a shortest path from u to t until the message arrives at t.
Case 2. Suppose $\text{dist}(u,t) > k$. Forwarding decisions are illustrated by Rules U1 and U2 in Figure 8. If the message originated at u or arrived at u via an inconsistent edge, then the algorithm forwards the message along any active edge of u.

LEMMA 12. *Algorithm 2 successfully delivers the message to the destination node t.*

PROOF. *(Sketch)* Since $3k+1 > n$, a node cannot have three or more active components. Therefore, each node has active degree at most 2. Suppose Algorithm 2 does not deliver the message to node t. The message must visit a repeating sequence of nodes of G. Using an argument similar to (but simpler than) the proof of Lemma 9, we derive a contradiction by showing that this repeating sequence must include a node that has an active component that is not visited (and eventually leads to node t). Since each node has active degree at most 2, this component must be visited according to rule U2. □

LEMMA 13. *Algorithm 2 has dilation at most 3.*

We omit the proof of Lemma 13 due to space constraints. Theorem 14 follows from Lemmas 12 and 13:

THEOREM 14. *For every $k \geq n/3$, there exists an origin-oblivious, predecessor-aware, k-local routing algorithm that succeeds on all connected graphs while guaranteeing dilation at most 3.*

5.3 Predecessor Oblivious and Origin Oblivious

We describe a $(n/2)$-local predecessor-oblivious and origin-oblivious routing algorithm.

Choose any $k \geq \lfloor n/2 \rfloor$. Therefore, $2k+1 \geq n$. Since a frontier component contains at least k nodes, every node has at most two frontier components. Let u denote the current node.

Algorithm 3: $(n/2)$-local, origin-oblivious, predecessor-oblivious routing algorithm.
Case 1. Suppose $\text{dist}(u,t) \leq k$. That is, $t \in G_k(u)$. The algorithm forwards the message to any neighbour of u on a shortest path from u to t until the message arrives at t.
Case 2. Suppose u has zero or two frontier components. Since $2k+1 \geq n$, the entire network is contained in $G_k(u)$ and, therefore, $d(u,t) \leq k$. Routing proceeds as in Case 1.
Case 3. Suppose u has one unconstrained frontier component. Since every unconstrained frontier component contains at least $2k$ vertices, the entire network is contained in $G_k(u)$ and, therefore, $d(u,t) \leq k$. Routing proceeds as in Case 1.
Case 4. Suppose $\text{dist}(u,t) > k$ and u has one constrained frontier component. Let v denote the constraint vertex in $G_k(u)$ that is furthest from u. The algorithm forwards the message to any neighbour of u that reduces the distance to v. This procedure continues until the algorithm enters Cases 1, 2, or 3.

LEMMA 15. *Algorithm 3 successfully delivers the message to the destination node t along a shortest path from s to t.*

PROOF. In Cases 1 through 3, the distance from the current node to the destination, $\text{dist}(u,t)$, decreases by one unit each time the message is forwarded. In Case 4, observe that $\text{dist}(u,t) = \text{dist}(u,v) + \text{dist}(v,t)$. Therefore a decrease in $\text{dist}(u,v)$ implies an equal decrease in $\text{dist}(u,t)$. It follows that the route from s to t has length $\text{dist}(s,t)$. Therefore, Algorithm 3 finds a shortest path from s to t. □

Theorem 16 follows from Lemma 15:

THEOREM 16. *For every $k \geq n/2$, there exists an origin-oblivious, predecessor-oblivious, k-local routing algorithm that succeeds on all connected graphs and finds a shortest path from the origin to the destination.*

Algorithm 3 can be defined analogously to Algorithms 1 and 2. That is, when at a node u, the same preprocessing step could be included to classify edges as active or passive. Upon doing so, a node has active degree at most 3 in Algorithm 1 ($k \geq n/4$), active degree at most 2 in Algorithm 2 ($k \geq n/3$), and active degree at most 1 in Algorithm 3 ($k \geq n/2$). Ignoring predecessor- and origin-awareness/obliviousness, this trend suggests that the locality parameter k is inversely proportional to the number of possible forwarding decisions that a k-local routing algorithm must consider at each node.

5.4 Predecessor Oblivious and Origin Aware

As we now show, an origin-aware k-local routing algorithm does not require a greater locality parameter k than does an origin-oblivious k-local routing algorithm to guarantee delivery.

THEOREM 17. *For every $k \geq n/2$, there exists an origin-aware, predecessor-oblivious, k-local routing algorithm that succeeds on all connected graphs and finds a shortest path from the origin to the destination.*

PROOF. For every k-local origin-oblivious routing algorithm $\mathcal{A}$ there is a corresponding k-local origin-aware algorithm $\mathcal{A}'$ whose routing function matches that of $\mathcal{A}$. In other words, providing knowledge of the origin cannot hinder an origin-oblivious routing algorithm. The result follows by Theorem 16. □

6. DIRECTIONS FOR FUTURE RESEARCH

Directed Graphs. The results of this paper concern k-local routing on undirected graphs. Of course, the analogous questions can be posed in the setting of directed graphs, many of which remain open. Preliminary investigations of local routing on directed graphs have been made by Chávez et al. [5] who describe 1-local routing algorithms for Eulerian graphs and outerplanar graphs and Fraser et al. [10] who show that every 1-local routing algorithm requires $\Omega(n)$ bits of memory on directed graphs (i.e., no stateless 1-local routing algorithm exists).

Using Additional Memory. Relaxing constraints and allocating additional memory the message overhead to store state information allows more general solutions to the local routing problem. Braverman [4] shows that there exists a position-oblivious 1-local routing algorithm using $\Theta(\log n)$ state bits that succeeds on all graphs. An interesting open question is to determine whether there is a corresponding lower bound. Alternatively, does there exist a position-oblivious, origin-aware, predecessor-aware, k-local routing algorithm with $o(\log n)$ state bits for $k \in O(1)$? In general, can we identify tight bounds on memory requirements for deterministic k-local routing under various models?

Acknowledgements

The authors wish to thank Therese Biedl who observed that our origin-oblivious and predecessor-oblivious $(n/2)$-local routing algorithm (Algorithm 3) identifies a shortest path.

7. REFERENCES

[1] P. Bose, A. Brodnik, S. Carlsson, E. D. Demaine, R. Fleischer, A. López-Ortiz, P. Morin, and I. Munro. Online routing in convex subdivisions. *International Journal of Computational Geometry and Applications*, 12(4):283–295, 2002.

[2] P. Bose and P. Morin. Online routing in triangulations. *SIAM Journal on Computing*, 33(4):937–951, 2004.

[3] P. Bose, P. Morin, I. Stojmenović, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.

[4] M. Braverman. On ad hoc routing with guaranteed delivery. In *Proceedings of the ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, volume 27, page 418, 2008.

[5] E. Chávez, S. Dobrev, E. Kranakis, J. Opatrny, L. Stacho, and J. Urrutia. Route discovery with constant memory in oriented planar geometric networks. *Networks*, 48(1):7–15, 2006.

[6] S. Durocher, D. Kirkpatrick, and L. Narayanan. On routing with guaranteed delivery in three-dimensional ad hoc wireless networks. *Wireless Networks*, 2008. To appear.

[7] G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, Information Sciences Institute, 1987.

[8] R. Flury and R. Wattenhofer. Randomized 3D geographic routing. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pages 834–842, 2008.

[9] M. Fraser. Local routing on tori. In *Proceedings of the Conference on Ad-Hoc, Mobile, and Wireless Networks*, volume 4686 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2007.

[10] M. Fraser, E. Kranakis, and J. Urrutia. Memory requirements for local geometric routing and traversal in digraphs. In *Proceedings of the Canadian Conference on Computational Geometry (CCCG)*, volume 20, 2008.

[11] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proceedings of the Canadian Conference on Computational Geometry (CCCG)*, volume 11, pages 51–54, 1999.

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

[13] I. Stojmenović. Position based routing in ad hoc networks. *IEEE Communications Magazine*, 40(7):128–134, 2002.