On Local Algorithms for Topology Control and Routing in Ad Hoc Networks

Lujun Jia*
College of Computer & Information Science
161 Cullinane Hall
Northeastern University
Boston, MA 02115, USA
lujunjia@ccs.neu.edu

Rajmohan Rajaraman†
College of Computer & Information Science
161 Cullinane Hall
Northeastern University
Boston, MA 02115, USA
rraj@ccs.neu.edu

Christian Scheideler
Department of Computer Science
Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218, USA
scheideler@cs.jhu.edu

ABSTRACT

An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any fixed infrastructure. Indeed, an important task of an ad hoc network is to determine an appropriate topology over which high-level routing protocols are implemented. Furthermore, since the underlying topology may change with time, we need to design routing algorithms that effectively react to dynamically changing network conditions.

The aim of this paper is to explore the limits of communication in wireless mobile networks, concentrating on local-control algorithms for topology control and routing. We analyze the performance of the algorithms under three measures: *throughput*, which is the rate at which packets can be delivered, *space overhead*, i.e. the space necessary to buffer packets, and the total *energy* consumed due to packet transmissions. Energy consumption is an important performance measure for ad hoc networks since the battery power of mobile nodes is usually limited.

Towards topology control, we show that for *any* distribution of nodes in the 2-dimensional Euclidean plane, a simple local algorithm allows to establish and maintain a connected constant degree overlay network that contains energy-efficient paths between every pair of nodes. Towards routing, we present a local routing algorithm that works for arbitrary overlay networks without transmission interference. We show that for *any* sequence of network changes and packet injections the algorithm is within a constant factor of the optimal, with respect to both throughput and energy, when compared to what a *best possible* routing algorithm can achieve under the same sequence of network changes and injection. We then combine the topology control and routing algorithms to obtain competitive wireless communication algorithms that account for transmission interference, an important performance-limiting aspect of wireless communication.

*Supported by NSF CAREER award NSF CCR–9983901.
†Supported by NSF CAREER award NSF CCR–9983901.

SPAA'03, June 7–9, 2003, San Diego, California, USA.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on discrete structures, geometrical problems and computations, routing and layout

General Terms

Algorithms, theory

Keywords

Distributed algorithms, mobile computing and communication, ad hoc wireless networks, routing, spanners, adversarial model, competitive analysis

1. INTRODUCTION

An ad hoc wireless network consists of a collection of geographically dispersed nodes communicating with one another over a wireless medium using paths that may traverse multiple nodes. An ad hoc network differs from both wired and cellular networks in that there is no wired infrastructure and the communication capabilities of the network are usually constrained by the limited battery power of the nodes. While primary applications of ad hoc networks are in the military domain [18], the rapid advent of mobile telephony and a plethora of personal digital assistants has brought to the fore a number of potential commercial applications of ad hoc networks. Examples are disaster relief, conferencing, home networking, sensor networks, personal area networks, and embedded computing applications [33].

The absence of a fixed infrastructure in ad hoc networks implies that an ad-hoc network does not have an associated fixed topology. Hence, the nodes themselves have to form a connected topology to enable communication among them. There are several factors that influence the topology of an ad hoc network. Some are controllable such as the transmission power of individual nodes and antenna direction, while others are uncontrollable such as node mobility, weather, and noise. Furthermore, since wireless nodes transmit by broadcasting within a certain (potentially variable) transmission range, two different simultaneous transmissions may *interfere*, and neither may succeed. For a given topology, we also need to identify routes and schedule packet movements so to ensure high throughput and to minimize energy consumption, an important measure for communication in ad hoc networks.

In general, designing optimal communication protocols in ad hoc networks is hard. For instance, it is known that finding a schedule for a set of packets in an ad hoc network of n nodes that completes in time within even an $O(n^{1-\varepsilon})$ factor of optimal, is NP-hard, for any constant $\varepsilon > 0$ [1]. The sheer complexity of establishing communication in ad hoc networks suggests a layered approach, addressing the following questions:

- **How to set up a topology that guarantees connectivity?** Distributed algorithms that address this problem will be called *topology control protocols*. A naive solution that is wasteful in both energy consumption and maintenance overhead is to simply connect each node to all other nodes within its maximum transmission range. In order to increase scalability and reduce interference, it is more desirable to maintain only a constant number of direct links for each node at any point of time, while trying to ensure that the topology offers energy-efficient routes between any pair of nodes. Note that just connecting each node to its closest k neighbors may provide energy-efficient routes but does *not* guarantee connectivity or a constant degree per node.

- **How to select connections provided by the topology to allow non-interfering transmissions of packets?** A topology that ensures connectivity necessarily contains edges that interfere with each other. Thus, a way has to be found to schedule the use of these edges. Algorithms for this problem will be called *medium access control (MAC) protocols*.

- **How to route packets along non-interfering connections?** Given an underlying topology, which may be dynamically changing, we need to determine routes for individual packets and decide which packet to schedule if several packets contend to use an edge at the same time. We will refer to algorithms for this problem as *routing protocols*.

1.1 Our results

In this paper, we consider the performance of simple local algorithms for topology control, medium access, and routing in ad hoc wireless networks. To the best of our knowledge, this is the first study in which all of these issues have been addressed and analyzed.

Our first result concerns a local algorithm for computing a *constant-degree, energy-efficient* topology for an arbitrary distribution of ad hoc network nodes in the 2-dimensional Euclidean plane. Let V be a set of nodes in the 2-dimensional plane. We adopt the following standard model for energy consumption. The energy consumed due to a direct transmission from u to v is given by $|uv|^\kappa$, where $\kappa \geq 2$ is a constant and $|uv|$ is the Euclidean distance between u and v. The preceding formula for energy consumption, which is discussed in more detail in Section 2.2, follows from a standard power attenuation model adopted for wireless transmissions [35, 41]. The total energy used for delivering a packet from source s to destination t along a path P is simply the sum of the energy used for all the edges in P. We define the *energy-stretch* of a path P between vertices u and v to be the ratio of the energy of P to the energy of the minimum-energy path between u and v. The energy-stretch is a variant of the well-known measure of distance-stretch, which for a path P is the ratio of the length of P to the the minimum distance between u and v.

- We show that a simple local-control algorithm, proposed by Li et al [32], identifies an $O(1)$-degree graph $\mathcal{N}$ on V such that for any two nodes u and v, there exists a path in $\mathcal{N}$ between u and v that has $O(1)$ energy-stretch. For the special case of *civilized graphs*, in which it is assumed that the ratio of the maximum edge length to the minimum edge length is bounded by a constant, we show that the same algorithm achieves $O(1)$ distance-stretch for any two nodes u and v. Our result, which is presented in Section 2, is related to work done on proximity graphs in computational geometry and may be of independent interest.

A topology control algorithm provides an underlying network over which a suitable routing mechanism can be implemented. Since this network is computed online and may further change due to uncontrollable factors (as discussed above), we need to design routing algorithms that react to dynamically changing network conditions.

First, we consider a scenario in which a topology control protocol and a MAC protocol are given that provides edges to the routing layer that can be used without interference. We investigate the performance of our routing algorithm under the situation that the MAC protocol and the packet injections show adversarial behavior. More precisely, we assume that there is an adversary that is allowed to inject an arbitrary number of packets and that can select an arbitrary set of non-interfering edges at any time step. We also associate a *cost* with each edge that represents, for example, the energy usage for transmission along the edge, and may change from one step to another.

- Under the above adversarial model, we present a routing algorithm in Section 3 that is based on a simple, local balancing approach. For any sequence of adversarial packet injections and edge activations and for any constant $\varepsilon > 0$, our algorithm successfully delivers a $1 - \varepsilon$ fraction of the packets at an average cost that is within an $O(1/\varepsilon)$ factor of optimal, assuming that the node buffer sizes in our algorithm are larger than the buffer sizes used in an optimal schedule by a factor of essentially $O(\bar{L}/\epsilon)$, where $\bar{L}$ is the average path length used for successful packets in an optimal solution. While algorithms based on local balancing have been extensively studied before, this is the first study that models transmission costs; it is somewhat surprising that a local-control algorithm achieves a constant-factor approximation with respect to *both throughput and average cost*, when compared with any other routing schedule. We also note that the generality of our adversarial model implies the applicability of our result in diverse scenarios involving dynamic networks.

An important assumption in the above result is that transmissions across all of the edges in the network can be scheduled simultaneously. As mentioned at the outset, wireless nodes transmit by broadcasting and, therefore, transmissions are prone to interference, even when the nodes are able to adjust their transmission ranges. We adopt a standard model for interference, that is described in Section 2.4. Our next set of results addresses the impact of interference on the throughput achievable on the topology $\mathcal{N}$ and on the throughput achieved by our local balancing algorithm.

- We show that for any communication pattern, the local balancing algorithm, when applied to network $\mathcal{N}$ using a simple randomized symmetry-breaking technique for resolving interference, achieves throughput within $\Omega(1/I)$ of the optimal achievable on *any topology*, where I is the maximum number of edges that any edge in $\mathcal{N}$ interferes with. If the n nodes are distributed uniformly at random in the plane, then we show that $I = O(\log n)$ whp[1], thus implying that our local algorithms achieve a throughput within an $O(\log n)$ factor of any other routing algorithm on any topology. These results follow from our analyses in Sections 2.4 and 3.3.

- Finally, we also show in Section 3.4 that for the special case where the transmission range of every node is uniform and *fixed*, one can achieve expected throughput which is optimal to within constant factors.

1.2 Related work

The topology control algorithm that we analyze in this paper was first proposed by Li et al [32] and is a variant of a graph introduced by Yao [44] for connecting nodes in Euclidean space. In the Yao graph, which is also commonly referred to as the θ-graph, one partitions the space around each node into *sectors* of a fixed angle and connects the node to the nearest neighbor in each sector. It can be easily shown that the Yao graph contains paths of $O(1)$ energy-stretch connecting any two nodes. In fact, the Yao graph satisfies the stronger property of being a *spanner*; that is, for any two nodes u and v, the Yao graph contains a path connecting u and v, the length of which is within a constant factor of the Euclidean distance between u and v. (Note that a spanner always has constant energy-stretch.) The maximum degree of the Yao graph is $\Omega(n)$ in the worst case, however. One can obtain a bounded-degree subgraph of the Yao graph that is also a spanner by processing the edges in order by length and adding an edge (u, v) to the subgraph if there is no other edge (u, w) or (v, w) already added and having an angle close to that of (u, v) [36] (a related idea is used in [7]). A topology control algorithm due to Wattenhoffer et al [43] (also see [31]) adopts a similar approach to convert the Yao graph to a constant-degree spanner. All of the suggested approaches, however, rely on a global ranking of the edges and it is not apparent how to implement such a postprocessing of the Yao graph edges without network-wide communication. In [42], the authors analyzed the topology control algorithm proposed in [32], and showed that this algorithm brings the maximum degree down to a constant and achieves the $O(1)$ energy-stretch for the special case of civilized graphs. In our paper, we show that the same topology control algorithm actually achieves the $O(1)$ energy-stretch for arbitrary graphs. We also analyze the throughput-efficiency of the resultant topology for arbitrary and random node distributions. Variants of the Yao graph, including the topology that we analyze in this paper, are also studied in [23]. We have recently learnt from Klaus Volbert, one of the authors of [23], that our result of $O(1)$ energy-stretch can also be proved using the analysis techniques presented in [23].

The Yao graph and the variants discussed above are closely related to a class of graphs referred to as *proximity graphs*, in which the graph edges are determined by the proximity among the nodes in Euclidean space. Proximity graphs include relative neighborhood graphs and Gabriel graphs [39]. While the relative neighborhood graph has polynomial energy-stretch, a Gabriel graph, by definition, has shortest paths with respect to the ℓ_2-norm and hence has optimal energy paths. The Gabriel graph, however, has $\Omega(n)$ degree in the worst case. Another geometric structure that leads to a spanner is the Delaunay triangulation of the set of points; without additional restrictions, however, the Delaunay triangulation graph may include edges much longer than the transmission range of a node. It has been shown that restricted Delaunay graphs [21], in which we only include Delaunay edges with a limited fixed transmission radius, are also spanners. The maximum degree of restricted Delaunay graphs is $\Omega(n)$ in the worst case, however. For a comprehensive survey on geometric spanners and other structures in geometric network design, see [17].

In recent years, a number of routing protocols have been proposed for ad hoc networks. A recent survey may be found in [38]. Most of these protocols rely on heuristics and, as such, do not provide provable worst-case guarantees. Our work is also related to routing protocols that exploit the underlying geometry of the network [25, 30].

Our results on adversarial routing build on a series of studies in adversarial queuing theory, which was initiated by Borodin et al. [15]. Other work on adversarial queuing theory includes [5, 19, 20, 22, 37, 40]. In these studies it is assumed that the adversary has to provide a path for every injected packet and reveals these paths to the system. The paths have to be selected in a way that they do not overload the system. Hence, it only remains to find the right queueing discipline (such as furthest-to-go) to ensure that all of the packets can reach their destination.

In the context of *packet routing algorithms*, the study of adversarial models was initiated by Awerbuch, Mansour and Shavit [13] and further refined by [4, 9, 11, 12, 20]. In the model adopted by these studies, the adversary *does not* reveal the paths to the system, and therefore the routing protocol has to figure out paths for the packets by itself. Based on work by Awerbuch and Leighton [12], Aiello et al. [4] show that there is a simple distributed routing protocol that keeps the number of packets in transit bounded in a dynamic network if, roughly speaking, in each window of time the paths selected for the injected packets require a capacity that is below what the available network capacities can handle in the same window of time.

Awerbuch et al. [9] study the problem of sending packets to a single destination in a dynamic network, using an adversarial model in which the adversary is allowed to control

[1] We use the abbreviation "whp" throughout the paper to mean "with high probability" or, more precisely, "with probability $1 - n^{-c}$, where n is the number of nodes in the network and c is a constant that can be set arbitrarily large by appropriately adjusting other constants defined within the relevant context."

the network topology and packet injections as it likes, as long as for every injected packet it can provide a schedule to reach its destination. They show that even for the case that the network capacity is fully exploited, the number of packets in transit is bounded at any time. Recently, Awerbuch et al. [10] extended these results to arbitrary anycasting situations and showed that simple balancing strategies achieve a throughput that can be brought arbitrarily close to a best possible throughput. Our work generalizes the results of [10] to incorporate edge costs. We also augment the algorithm to account for interference. We note that all of the above work in the adversarial routing area, including this current paper, is based on simple load balancing schemes first described in [13], and refined in [2, 3, 4, 6, 9, 10, 11, 12] for various routing purposes.

2. TOPOLOGY CONTROL

We consider a set V of n nodes in a 2-dimensional plane, in which each node can directly communicate with every node within a maximum distance D. Let $G^* = (V, E)$ denote the *transmission graph* that contains an edge between two nodes u and v if they can directly communicate with each other. We assume throughout this paper that G^* is connected. For each edge (u, v) in E, we associate a *cost* $c(u, v) = |uv|^\kappa$ for $\kappa \geq 2$, where $|uv|$ denotes the Euclidean distance between u and v. In the cost we consider only transmission energy, since receiving energy is independent of the transmission distance and normally smaller compared to transmission energy. We also assume that κ is the same for all the links in the network. The cost of a path P between u and v is the sum of the cost of the edges along P.

We now elaborate on the assumptions made in our model. The parameter D represents the maximum transmission range of any node. The cost assigned to an edge represents the transmission energy expended and is based on a standard power attenuation model [35], in which the receiving power at any receiver is given by $\Theta(\frac{P_T}{d^\kappa})$, where P_T is the transmission power, d is the distance between the transmitter and the receiver and $2 \leq \kappa \leq 4$ is a constant. Thus, if we assume that each node has the same power reception threshold for signal detection, then the transmission power, and hence the energy, required for transmission over edge (u, v) is proportional to $|uv|^\kappa$, which is what we assign as energy cost for edge (u, v). We assume that each node is equipped with a GPS receiver, which can provide the position information. We also assume that each node is able to adjust its transmission power according to the distance to its receiver [43, 34], as long as the power does not exceed the maximum power needed to transmit to a distance of D.

Given an arbitrary collection of nodes forming a transmission graph G^*, we seek a distributed algorithm that identifies a low-degree subgraph of G^* that contains energy-efficient paths and admits high throughput. We capture the energy-efficiency of a subgraph H by its energy-stretch, which we now define. For any subgraph H of G^* and any nodes u and v, define $E^H_{u,v}$ to be the cost of the path with least cost in H. We define the *energy-stretch* of a subgraph H to be maximum ratio, over all nodes u and v, of $E^H_{u,v}$ to $E^{G^*}_{u,v}$. The main result of this section is that for any distribution of the n nodes, a simple local algorithm computes an $O(1)$-degree topology $\mathcal{N}$ with $O(1)$ energy-stretch. We note that the results of Wang et al. [42] establish the $O(1)$ energy-stretch property of $\mathcal{N}$ for the special case of civilized graphs [27]. For this special case, we show in this section that the topology $\mathcal{N}$ actually achieves $O(1)$ distance-stretch, which directly implies the $O(1)$ energy-stretch result in [42]. For a general distribution of nodes, however, we have not been able to resolve whether $\mathcal{N}$ is a spanner and we leave this question as an open problem at this time. The algorithm is described in Section 2.1, and the analysis of energy-stretch and distance-stretch are presented in Section 2.2 and Section 2.3 respectively.

We also evaluate the topology $\mathcal{N}$ on the basis of the throughput achievable for arbitrary communication patterns. The throughput achievable on a topology depends on the degree to which the edges of the topology interfere. A formal definition of the interference model and the analysis of throughput are give in Section 2.4.

2.1 Algorithm

In this section, we describe the topology control algorithm proposed in [32]. The algorithm is parametrized by an angle $\theta \leq \pi/3$. We refer to the algorithm as ΘALG. Each node $u \in V$ divides the 360° space into $2\pi/\theta$ *sectors* or *cones*. For any two nodes u and v, we let $S(u, v)$ denote the sector of u containing node v. In the following description, we assume, without loss of generality, that all pairwise distances among the n nodes are unique. (If the distances are not unique, then a simple tie breaking scheme can be used to enforce the assumption.) The ΘALG determines a subgraph $\mathcal{N} = (V, E)$ of G in two phases:

1. Each node u computes $N(u)$ which consists of all nodes v such that v is the node nearest to u in $S(u, v)$.
2. Edge $(u, v) \in E$ is in E if v is the nearest node in $S(u, v)$ such that $u \in N(v)$ or u is the nearest node in $S(v, u)$ such that $v \in N(u)$.

Let $\mathcal{N}_1$ denote the graph obtained after the first phase of the algorithm; that is $\mathcal{N}_1 = (V, E_1)$ where $(u, v) \in E_1$ if $u \in N(v)$ or $v \in N(u)$. The graph $\mathcal{N}_1$ is identical to the Yao graph with θ-degree sectors. One can easily prove by an induction on pairwise distances that the distance between two nodes u and v in the graph is $O(|uv|)$, and hence that $\mathcal{N}_1$ is a spanner. It follows then that $\mathcal{N}_1$ also has $O(1)$ energy-stretch.

While the total number of edges in $\mathcal{N}_1$ is $O(n)$, the degree of a node may be as large as $\Omega(n)$ in the worst-case. One can construct a constant-degree subgraph of $\mathcal{N}_1$ by processing the edges in order of decreasing length, and eliminating edges that do not decrease the distance between endpoints by more than a constant-factor [43]. Such a postprocessing step, however, takes communication time proportional to the diameter of the network. Instead, the second phase of the algorithm above proposes a simple local step to eliminate certain edges from $\mathcal{N}_1$ so that the degree of each node is a constant. Note that if we assign direction to each edge in $\mathcal{N}_1$, i.e. u has a directed edge to v if u identifies v as its nearest neighbor in $S(u, v)$, then it is easy to see that each node in $\mathcal{N}_1$ has constant out-degree while in-degree can be $O(n)$. In the second phase of the algorithm, the in-degree of each node u is then reduced to constant by letting u admit only one incoming edge, the shortest one, in each of u's sector. In Section 2.2, we will show that the resultant graph $\mathcal{N}$ has $O(1)$ energy-stretch.

Before going on to the analysis, we note that ΘALG can be implemented by three rounds of local message broadcasting and computation. In the first round, each node broadcasts a *Position* message containing its position acquired from its GPS receiver, at maximum power P. After receiving the position information, each node u computes $N(u)$ based on angle information computed from the received *Position* messages. In the second round, each node u broadcasts a *neighborhood* message containing $N(u)$ to each node in $N(u)$. In the third round, each node u sends a *connection* message to the nearest node v (if any) in each sector such that u is in $N(v)$. The topology $\mathcal{N}$ has an edge (u, v) for any pair of nodes u and v that have exchanged a connection message. We note that the three rounds of message exchanges may take a variable amount of time due to the interference and confliction.

It can be easily shown that the topology $\mathcal{N}$ is connected and has constant degree [42]. The proof of 2.1 is presented in the full paper [28].

LEMMA 2.1 ([42]). *$\mathcal{N}$ is connected and the degree of each node is at most $4\pi/\theta$.*

2.2 Analysis of energy-stretch

In this section, we show that $\mathcal{N}$ has $O(1)$ energy-stretch. Let $E_{u,v}$ denote the cost of the minimum-cost path from u to v in $\mathcal{N} = (V, E)$. We will show that for any pair of nodes u, v, $E_{u,v}$ is within a constant factor of the minimum cost to transmit from u to v in G^*. Since, the transmission along any edge (u, v) in G^* incurs cost $|uv|^\kappa$, it suffices to show that for any edge $(u, v) \in G^*$, $E_{u,v}$ is $O(|uv|^\kappa)$. Our main theorem is as follows

THEOREM 2.2. *For θ sufficiently small, $E_{u,v} = O(|uv|^\kappa)$, for any edge (u, v) in G^*.*

The proof of Theorem 2.2 proceeds by induction on pairwise distances. A challenge in establishing Theorem 2.2 is that unlike proximity graphs such as the Yao graph [44], Gabriel graph and some of its variants (such as β-skeletons with $\beta < 1$) [17], the minimum-cost path in $\mathcal{N}$ from a node u to another node v may traverse nodes that are farther from v than u is. We are able to overcome this hurdle by sufficiently characterizing such a path so as to place an upper bound on the cost. For our proof of Theorem 2.2, we need a series of technical lemmas, which can be proved using elementary geometry, to establish relationships among node distances and relative orientation. We here give a proof sketch for Theorem 2.2, and proofs of the technical lemmas and Theorem 2.2 is presented in the full paper [28].

LEMMA 2.3. *For any $\triangle ABC$ with $|AC| \leq |BC|$ and $\angle ACB \leq \pi/3$, $c|AB|^2 + |AC|^2 \leq c|BC|^2$ for $c \geq \frac{1}{2\cos(\angle ACB)-1}$.*

LEMMA 2.4. *For any $\triangle ABC$ with $|BC| \leq |AC| \leq |AB|$ and $\angle BAC \leq \pi/6$, $|BC| \leq \frac{|AB|}{2\cos\angle BAC}$.*

LEMMA 2.5. *Let $A, A_1, A_2, \ldots, A_k$ be a set of points, such that $|AA_i| \geq |AA_{i+1}|$ and $0 \leq \angle A_iAA_{i+1} \leq \theta$. If $\angle A_1AA_k = \alpha$, then $\sum_{i=1}^{k-1} |A_iA_{i+1}|^2 \leq (|AA_1| - |AA_k|)^2 + 2|AA_1|^2 \frac{\alpha}{\theta}(1 - \cos\theta)$.*

LEMMA 2.6. *Let A and B be any two points, and O be the center of line segment (A, B). Let D be the point such that $|BD| = |AB|$ and $\angle DBA = \pi/6$. Let C be a point outside $C(O, |OA|)$ such that $|AC| \leq |AB|$ and $\angle CAB < \pi/12$ and C, D are on the same side of (A, B). Let E be the intersection of (C, D) with circle $C(O, |OA|)$. We have $\angle EAB \leq 2\angle CAB$.*

Proof Sketch: We prove that $E_{u,v} \leq c|uv|^\kappa$ for an appropriately chosen c, for any pair of nodes $u, v \in V$. For simplicity, we assume $\kappa = 2$ in the following analysis, and it is easy to see that the analysis holds for any $\kappa \geq 2$. We prove this theorem by an induction on $|uv|$ for any $u, v \in V$.

Consider the base case where $|uv|$ is the minimum among any pair of nodes in V. This is trivial, since it is clear that $(u, v) \in E$ and $E_{u,v} = |uv|^2 \leq c|uv|^2$, for $c \geq 1$. Consider nodes u, v. Assume for the purpose of induction that for any pair of nodes x, y with $|xy| < |uv|$, $E_{x,y} \leq c|xy|^2$. We show in the following that $E_{u,v} \leq c|uv|^2$. We distinguish between two cases.

Case 1: $v \in N(u)$, or $u \in N(v)$. In this case, u selects v as its nearest neighbor, so there exists node w ($w = v$ or $w \neq v$) in sector $S(v, u)$ with $\angle uvw \leq \theta$ such that $|wv| \leq |uv|$ and $(w, v) \in E$. Invoking Lemma 2.3, we have $E_{u,v} \leq E_{u,w} + |wv|^2 \leq c|uw|^2 + |wv|^2 \leq c|uv|^2$, for $c \geq \frac{1}{2\cos\theta - 1}$.

Case 2: $v \notin N(u)$ and $u \notin N(v)$. In this case, if there exists a node w with $\angle uwv \geq \pi/2$, we have $E_{u,v} \leq E_{u,w} + E_{w,v} \leq c|uw|^2 + c|wv|^2 \leq c|uv|^2$.

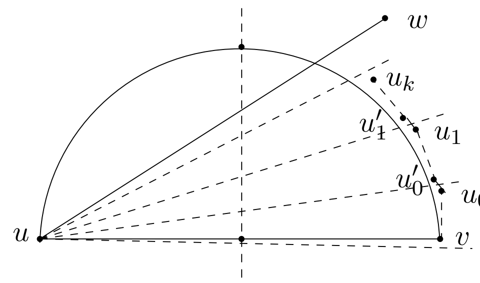

Figure 1: Sectors of node u.

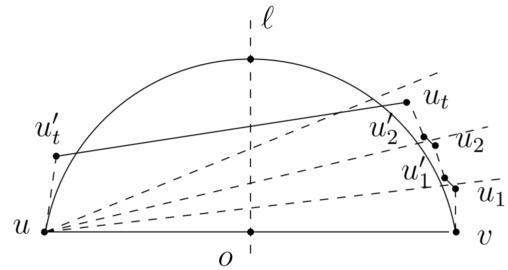

Figure 2: Node u_t has edge (u_t, u'_t).

We now assume that $\angle uwv < \pi/2$ for any $w \in V$. Consider Figure 1. Let $\angle wuv = \pi/6$. We number the consecutive sectors of u as $0, 1, \ldots, k$, starting from $S(u, v)$. Let k be the largest number k such that the total angle spanned by the sectors is at most $\pi/6$. We denote the nearest neighbor of u in sector i as u_i, and let u'_i be the node such that $u_i \in N(u'_i)$ and $(u_i, u'_i) \in E$, if such a node exists. We then let u_t be the first node in the sequence $u_0, u_1, \ldots, u_k$, such that $|uu'_t| < |u'_tu_t|$. We distinguish between the following two cases:

Case 2.1: u_t exists for $0 \leq t \leq k$. Consider Figure 2. By applying an induction on i for $1 \leq i \leq t$, we are able to show that there exists a sequence of nodes $v, u_1, u_2, \ldots, u_t$, such that their distance to u is decreasing and $\angle vuu_1, \angle u_iuu_{i+1} \leq 2\theta$ for $1 \leq i \leq t-1$. Specifically, we have an edge $(u_t, u'_t) \in E$ with a length of $\Omega(|uv|)$. Invoking Lemma 2.4 on edge (u, u'_t)

and invoking Lemma 2.5 on sequence $v, u_1, u_2, \ldots, u_t$, we can show that $E_{u,v} \le E_{u,u'_t} + E_{u'_t,u_t} + E_{u_t,v} \le c|uv|^2$, for some constant c and θ.

Case 2.2: u_t does not exist for $0 \le t \le k$. In this case, we have a sequence of nodes $v, u_1, u_2 \ldots, u_k$, by we are unable to identify an edge of length $\Omega(|uv|)$ from any of the u_i's. Instead, we identify another sequence of nodes on u's side. Consider Figure 3. First, we determine a node z with the following properties.:

1. $\theta \le \angle zuv \le 4\theta$;
2. the nearest neighbor of z in sector $S(z, u)$ is on the same side of (u, v) as z;
3. any neighbor of z between ray (z, u) and (z, w) is on the same side of ℓ as u.

This is done by invoking Lemma 2.6 (a detailed argument is given in the full paper [28]). Then, we number the consecu-

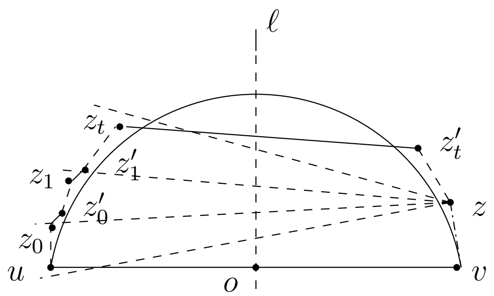

Figure 3: Node z_t has edge (z_t, z'_t).

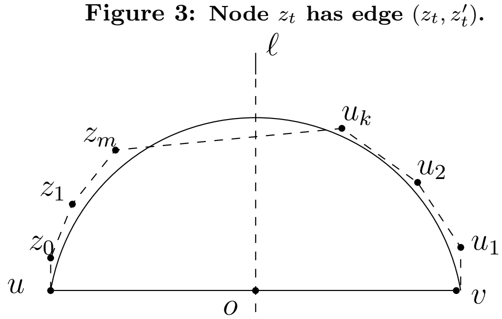

Figure 4: Node z_t does not exist.

tive sectors of z between ray (z, u) and (z, w) as $0, 1, \ldots, m$, starting from $S(z, u)$. The number m is selected to be the largest number such that the angles those sectors span is close to $\pi/6$. Similar to Case 2.1, we denote the nearest neighbor of z in sector i as z_i, and let z'_i denote the node such that $z_i \in N(z'_i)$ and $(z_i, z'_i) \in E$, if such a node exists. Let z_t be the first node in the sequence $z_0, z_1, \ldots, z_m$, such that $|z_t z'_t| > |z'_t z|$. We then further distinguish between the following two subcases:

Case 2.2.1: z_t exists for $0 \le t \le m$. Consider Figure 3. In this case, we have an edge (z_t, z'_t) with length $\Omega(|uv|)$ and, as in Case 2.1, we can show that $E_{u,v} \le E_{u,z_t} + E_{z_t,z'_t} + E_{z'_t,z} + E_{z,v} \le c|uv|^2$ for some constant c and θ.

Case 2.2.2: z_t does not exist for $0 \le t \le m$. Consider Figure 4. In this case, we have two sequences of nodes $v, u_0, u_1, \ldots, u_k$ and $u, z_0, z_1, \ldots, z_m$, in which the distance between u_k and z_m is considerably small compared with $|uv|$, which enables us to apply the induction step on $|u_k z_m|$ directly. Invoking Lemma 2.5 on the two sequences of nodes, we are able to show that $E_{u,v} \le E_{u,z_n} + E_{z_n,u_k} + E_{u_k,v} \le c|uv|^2$, for certain $\theta \le \pi/60$. This completes a sketch of the proof. We note that we have not attempted to optimize the value of θ in our analysis. □

2.3 Analysis of distance-stretch for civilized graphs

A *civilized graph*, also called λ-precision graph, satisfies the following property: for any nodes u_1, u_2, v_1, v_2 in the graph, $\min\{\frac{|u_1 v_1|}{|u_2 v_2|}\} \ge \lambda$, where $0 < \lambda \le 1$ is a constant. This is a commonly used model for wireless ad hoc networks [42, 23], since wireless devices typically are not too close to each other. Let Θ-*distance* $D^{\mathcal{N}}(u, v)$ denote the minimum distance between u, v in topology $\mathcal{N}$. Our main theorem in this section is as follows.

THEOREM 2.7. *If G^* is a civilized graph, then topology $\mathcal{N}$ has a distance-stretch of $O(1)$ for sufficiently small θ.*

Due to space constraints, we have placed the proof of Theorem 2.7 in the full paper [28].

2.4 Interference model and throughput analysis

Modeling interference in a wireless environment is a complex task. The wireless medium is susceptible to path loss, noise, interference and blockages due to physical obstructions. In this paper, we adopt a *pairwise* interference model, in which we specify conditions on the distances among participating nodes under which a given transmission is successfully received. Let $X_1, X_2, \ldots, X_k$ be the set of nodes transmitting simultaneously to receivers $Y_1, Y_2, \ldots, Y_k$, respectively, at some instant. Then the transmission by X_i is *successfully received* by node Y_i if $|X_j Y_i| \ge (1 + \Delta)|X_j Y_j|$, for every other node X_j, where $\Delta > 0$ models a protocol specified guard zone to prevent transmission interference. This means any node Y_i that falls into the guard zone $(1 + \Delta)|X_j Y_j|$ of transmission from X_j to Y_j will be interfered. Note that similar pairwise interference model is also adopted in [24] (the *protocol* model), and is a simplified version of the *physical* model [24], which considers a combined interference from all other simultaneous transmissions.

Let topology $\mathcal{N} = (V, E)$. We consider any message exchange between X_i and Y_i as a bidirectional communication consisting of a transmission from X_i to Y_i and another transmission from Y_i to X_i, to account for both data packets and control packets such as acknowledgments. We define

$$IR(X_i, Y_i) = C(X_i, (1 + \Delta)|X_i Y_i|) \bigcup C(Y_i, (1 + \Delta)|X_i Y_i|)$$

to be the *interference region* of transmission $X_i \leftrightarrow Y_i$, where $C(O, r)$ denotes the open disk with center O and radius r. Thus, $X_i \leftrightarrow Y_i$ is successful if and only if for any other transmission $X_j \leftrightarrow Y_j$, both X_i and Y_i are not in $IR(X_j, Y_j)$. We say that an edge e' *interferes* with $e \in E$ if the interference region of e' contains at least one end point of e. Following the recent work of Meyer auf der Heide et al [8], we define the *interference set* of e as $I(e) = \{e' \in E \mid e' \text{ interferes with } e, \text{ or vice versa}\}$, and call $\max_{e \in E}\{I(e)\}$ the *interference number* of the graph.

For an arbitrary communication pattern, the throughput achievable on a given topology depends on both the interference number of the topology and the congestion of the best path system connecting source-destination pairs, both of which in turn are a function of the distribution of the nodes in the plane. In the following theorem, we show the throughput achievable on $\mathcal{N}$ is essentially limited only by its interference number, when compared with an optimal schedule on G^*.

THEOREM 2.8. *Let I be the interference number of topology $\mathcal{N}$. Let W denote a set of packets that are successfully delivered by an arbitrary schedule of packet transmissions in G^* in t steps. Then, there exists a schedule of transmissions in $\mathcal{N}$ that delivers W in $O(tI + n^2)$ steps. Thus, for sufficiently large t and W, the throughput achievable on $\mathcal{N}$ is an $\Omega(1/I)$ fraction of the optimal.*

We here give an overview of the proof of Theorem 2.8. Let T be any set of edges in G^* such that the any two edges do not interfere with each other. We show in the following that any edge (u, v) in T can be replaced by a set of edges $\{(u, u_1), (u_1, u_2), \ldots, (u_k, v)\}$ from $\mathcal{N}$, such that any edge in $\mathcal{N}$ can be included in at most a constant number of such set of edges in T. We replace an edge (u, v) in G^* by a path P, which is recursively computed as follows. Initially, we have $P = \phi$. If $(u, v) \in E$, then P is simply the edge (u, v). Otherwise, we have two cases. First, if v is the nearest neighbor of u in $S(u, v)$, then let w be the node in $S(v, u)$ such that (v, w) is an edge. We set P to be the recursive path from u to w, which we call the θ-path, followed by the edge (w, v). Second, if v is not the nearest neighbor of u in $S(u, v)$, then let w be the nearest neighbor of u in $S(v, u)$. We set P to be the recursive path from u to w followed by the recursive path from w to v.

LEMMA 2.9. *Any edge in $\mathcal{N}$ can be selected by at most 6 θ-paths of edges in any T.*

Theorem 2.8 follows from Lemma 2.9 and a carefully designed scheduling [28]. While the minimum interference number can be large in the worst case, we show a logarithmic upper bound on the interference number of $\mathcal{N}$ for a random node distribution can be established.

LEMMA 2.10. *If the n nodes are placed independently and uniformly at random in a unit square, then the interference number of $\mathcal{N}$ is $O(\log n)$ whp.*

3. ROUTING

In this section we will show how to perform routing in wireless networks to ensure that, in conjunction with certain topology control and medium access control protocols, the throughput and energy efficiency is close to a best possible. After describing our basic model, we investigate various scenarios of comparing optimal algorithms with our algorithms:

1. First, we assume that protocols for topology control and the medium access control are already given. This means that in each step a set of edges is provided that do not interfere with each other and therefore can be used concurrently. MAC layer protocols that allow to achieve this are, for example, CSMA/CA[16], MACA [14, 29] and IEEE 802.11 [26]. Thus, it remains to perform routing decisions to achieve a throughput that is as high as possible.
2. Next, we assume that a topology control protocol is only given, and medium access control and routing protocols have to be designed. In this case, we compare the performance of our algorithm with a best possible routing strategy using the interference number of underlying topology.
3. Finally, we investigate two special cases, one when the ad hoc network nodes are randomly placed in a unit square and the other where the nodes are arbitrarily placed but have a fixed transmission strength and hence have to transmit every packet in the same range.

3.1 Analytical approach

We adopt a model in which the topology changes and packet injections are under adversarial control. That is, in each time step the adversary can specify a new topology with edge costs that may differ from previous edge costs, and it can inject an unbounded number of packets. Of course, in this case only some of the injected packets may be able to reach their destination, even when using a best possible strategy. For each of the successful packets a *schedule* can be specified. A schedule $S = (t_0, (e_1, t_1), \ldots, (e_\ell, t_\ell))$ consists of a sequence of movements by which the injected packet P can be sent from its source node to its destination node. It has the property that P is injected at time step t_0, the edges $e_1, \ldots, e_\ell$ form a connected path, with the starting point of e_1 being the source of P and the endpoint of e_ℓ being the destination of P, the time steps have the ordering $t_0 < t_1 < \ldots < t_\ell$, and edge e_i is active at time t_i for all $1 \le i \le \ell$.

We assume that at most one packet can be transmitted along any edge in each direction and require that no two schedules conflict with each other. That is, no edge is used by two schedules at the same time. When speaking about schedules in the following, we always mean a delivery strategy chosen by a best possible routing algorithm.

We assume that every node v in the system has a buffer $Q_{v,d}$ for each destination d. If a packet reaches its destination buffer $Q_{d,d}$, it is absorbed, and we count it as a successful delivery. The number of deliveries that is achieved by an algorithm is called its *throughput*. Since the adversary is allowed to inject an unbounded number of packets, we will allow routing algorithms to drop packets so that a high throughput can be achieved with a buffer size and a cost that is as small as possible.

In order to compare the performance of an optimal algorithm with our online algorithm, we will use competitive analysis. Given any sequence of topology changes and packet injections σ, let $\mathrm{OPT}_{B,C}(\sigma)$ be the maximum possible throughput (i.e. the maximum number of deliveries) achievable when using a buffer size of B and allowing an average cost of C per delivery (where the average is taken by dividing the total cost spent on all packets by the number of successful deliveries). Let $A_{B',C'}(\sigma)$ be the throughput achieved by some given online algorithm A with buffer size B' and an asymptotic average cost of at most C' per delivery. (Asymptotic means here that as the number of successful deliveries goes to infinity, the average cost goes to at most C'.) We call an online algorithm A (t, s, c)-*competitive* if for all σ and all B and C, A can guarantee that

$$A_{s \cdot B, c \cdot C}(\sigma) \ge t \cdot \mathrm{OPT}_{B,C}(\sigma) - r$$

for some value $r \ge 0$ that is independent of σ (but may depend on s, B and n). Note that $t \in [0, 1]$. For the case that we do not consider the cost of transmissions, we simply say that A is (t, s)-competitive.

3.2 MAC-based routing

We begin with the scenario in which protocols for topology and medium access control are given and it remains to provide a routing protocol. Recall that in this case edges are provided in each step that do not interfere with each other.

The (T, γ)-balancing algorithm

Let $h_{(v,d),t}$ denote the amount of packets in buffer $Q_{v,d}$ at the beginning of time step t. For any destination buffer $Q_{d,d}$, $h_{(d,d),t} = 0$ at any time. $h_{(v,d),t}$ will also be called the *height* of buffer $Q_{v,d}$ at step t. The maximum height a buffer can have is denoted by H.

We now present a simple balancing strategy. Several variants of it have been used in previous papers (e.g. [4, 9, 10]), but without considering edge costs. In every time step $t \geq 1$ the (T, γ)-balancing algorithm performs the following operations.

1. For every edge $e = (v, w)$, determine the destination d with maximum $h_{(v,d),t} - h_{(w,d),t} - c(e) \cdot \gamma$ and check whether $h_{(v,d),t} - h_{(w,d),t} - c(e) \cdot \gamma > T$. If so, send a packet from $Q_{v,d}$ to $Q_{w,d}$ (otherwise do nothing).
2. Receive incoming packets and absorb all packets that reached the destination. Afterwards, receive all newly injected packets. If a packet cannot be stored in a buffer because its height is already H, then delete the new packet.

In the above algorithm, we assume that nodes continuously exchange the buffer height values. In a practical implementation, we can reduce the amount of control information exchange for this purpose. This aspect is discussed in detail in the full paper [28].

Note that if T is set large enough to ensure that packets can only move downwards in their buffer position, then only newly injected packets will ever get deleted. In this case, the admission control problem for the sources has a simple solution: only admit those packets for which there is still buffer space available. We show that this solution is surprisingly effective.

Let δ denote the maximum number of edges incident to a node that can be used concurrently, i.e. the maximum number of available frequencies. Let $\bar{C}$ denote the average cost allowed for an optimal algorithm to deliver a packet and $\bar{L}$ denote the (best possible) average path length used by successful packets in an optimal algorithm under this assumption. We assume that $\bar{C}$ is known to the online algorithm whereas for $\bar{L}$ just an upper bound may be known. The following result demonstrates that the (T, γ)-balancing algorithm can reach a $(1-\epsilon)$-fraction of the optimal throughput at the cost of increasing the buffer size by a factor of essentially $O(\bar{L}/\epsilon)$ and the average cost per packet by a factor of $O(1/\epsilon)$. The proof is presented in the full paper [28].

THEOREM 3.1. *For any $\epsilon > 0$ and any $T \geq B + 2(\delta - 1)$ and $\gamma \geq (T + B + \delta)\bar{L}/\bar{C}$, the (T, γ)-balancing algorithm is $(1 - \epsilon, 1 + 2(1 + (T + \delta)/B)\bar{L}/\epsilon, 1 + 2/\epsilon)$-competitive.*

3.3 Topology-based routing

Next we show that it is possible to compete with an optimal algorithm even when medium access control is not provided. Recall the definition of the interference number in Section 2.4. Suppose that we use a topology control algorithm such as ΘALG of Section 2, and suppose that every node v knows for every edge $e = (v, w)$ of the resultant topology an upper bound I_e on the maximum current interference number of any edge e interferes with. (In the ideal, 2-dimensional Euclidean space it would actually suffice just to have an upper bound on the own interference number, but in a space with obstacles, for example, this would not suffice.) Then we use the following simple symmetry-breaking technique to provide medium access control: Each edge e provided by the topology control scheme chooses to become active with probability $1/(2I_e)$. All active edges are passed on to the (T, γ)-balancing algorithm. We refer to the combined medium access and routing protocol as a (T, γ, I)-*balancing* algorithm, where I is the maximum of I_{e} over all edges every offered by the topology control protocol.

We note that if the algorithm decides to send packets along two active edges that interfere with each other, then neither of the transmissions is successful. Fortunately, the following lemma demonstrates that there is a high probability of successfully using an active edge.

LEMMA 3.2. *Every active edge has a probability of at most 1/2 to interfere with other active edges.*

Now we are ready to compare our (T, γ, I)-balancing algorithm with the performance of an optimal algorithm. We assume here that an optimal algorithm still has to restrict itself to the edges provided by the topology control scheme, but apart from that is free to use any of these edges for communication as it likes. We even allow it to use edges successfully at the same time that would normally interfere with each other. Let δ, the maximum degree of a node in a step, be now equal to 1 (i.e. only one frequency is available).

THEOREM 3.3. *For any $\epsilon > 0$ and any $T \geq 2B + 1$ and $\gamma \geq (T + B)\bar{L}/\bar{C}$, the (T, γ, I)-balancing algorithm is $((1 - \epsilon)/(8I), 1 + 2(1 + T/B)\bar{L}/\epsilon, 1 + 2/\epsilon)$-competitive w.r.t. an optimal algorithm that is based on the same topology control scheme.*

The proof is presented in the full paper [28]. Theorem 3.3 can be combined with an analysis along the lines of Theorem 2.8 to yield the following claim for ΘALG and the (T, γ, I)-balancing algorithm, when compared with an optimal algorithm that is unrestricted in what edges it can use.

COROLLARY 3.4. *Suppose the nodes in the ad hoc network are stationary and the adversary only controls packet injections. For suitable values of T and γ, the (T, γ, I)-balancing algorithm, in conjunction with ΘALG, is $(O(1/I), O(\bar{L}))$-competitive w.r.t. an optimal algorithm that may use any edges of G^*.*

For the special case of having a random distribution of nodes in the unit square, Corollary 3.4 and Lemma 2.10 imply the following:

COROLLARY 3.5. *Suppose the nodes in the ad hoc network are randomly distributed in a unit square and the adversary only controls packet injections. For suitable values of T and γ, the (T, γ, I)-balancing algorithm, in conjunction with ΘALG, is $(O(1/\log n), O(\bar{L}))$-competitive w.r.t an optimal algorithm that may use any edges of G^*.*

3.4 Communication with fixed transmission strength

Finally, we demonstrate that an even better competitive ratio than the one given in Corollary 3.5 can be shown if the nodes are distributed in an arbitrary way in a 2-dimensional Euclidean space but all nodes have the same, fixed transmission strength. That is, we assume that every node transmits at the same fixed power level so that it will be successfully received by all nodes within distance 1, if there were no interference. Now recall the interference model in Section 2.4. For a transmission from a sender s to a receiver t to be successful, two properties have to be kept: (i) the distance between s and t is at most 1, and (ii) every node in every other sender-receiver pair must have a distance of more than $1+\Delta$ from s and t. If (ii) holds for two sender-receiver pairs, they are said to be *independent*.

Consider now the 2-dimensional space to be partitioned into a honeycomb-like hexagonal pattern as shown in Figure 5, with hexagons of side length $3 + 2\Delta$ (and therefore diameter $2(3 + 2\Delta)$). Each sender-receiver pair (s, t) is assigned to that hexagon that contains s. Our strategy for selecting independent sender-receiver pairs is rather simple: Suppose that every sender-receiver pair has a benefit associated with it, which equals the maximum difference in buffer heights, over all destination buffers. Within each hexagon, we first determine the sender-receiver pair of maximum benefit (breaking ties in an arbitrary way). If this sender-receiver pair has a benefit of more than some threshold $T > 0$, it is called a *contestant*. For each contestant, we decide with probability p_t to transmit a packet along its connection, where p_t is chosen so that the probability of a successful transmission is at least $1/2$. Two important lemmas can be shown for this strategy.

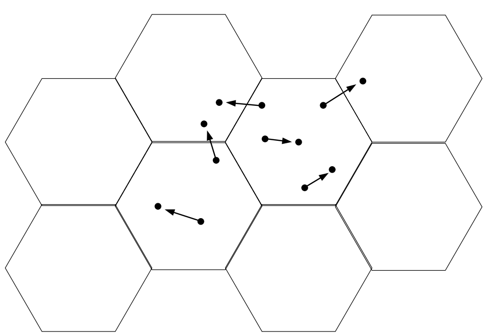

Figure 5: Subdivision of the Euclidean space into hexagons.

LEMMA 3.6. *The sum of the benefits of all contestants is by at most some constant factor c_b smaller than the maximum total benefit that can be achieved by any independent set of sender-receiver pairs with benefit beyond T.*

LEMMA 3.7. *If $p_t \leq 1/6$, then for each contestant (s, t), with probability at least $1/2$ no other contestant is selected that interferes with (s, t).*

The proofs of the two lemmas are presented in the full paper [28]. Let the *honeycomb algorithm* be a combination of the contestant selection strategy and the $(T, \gamma, 3)$-balancing algorithm applied to the contestants. The two lemmas above and Theorem 3.3 then yield the following result.

THEOREM 3.8. *For any $\epsilon > 0$ and any $T \geq 2B + 1$, the honeycomb algorithm is $((1-\epsilon)/(24\, c_b), 1+(1+T/B)L/\epsilon, 1+2/\epsilon)$-competitive.*

4. REFERENCES

[1] M. Adler and C. Scheideler. Efficient communication strategies for ad-hoc wireless networks. *Theory of Computing Systems*, 33:337–391, 2000.

[2] Y. Afek and E. Gafni. End-to-end communication in unreliable networks. In *Proc. of the 7th IEEE Symp. on Principles of Distributed Computing (PODC)*, pages 131–148, 1988.

[3] W. Aiello, B. Awerbuch, B. Maggs, and S. Rao. Approximate load balancing on dynamic and synchronous networks. In *Proc. of the 25th ACM Symp. on Theory of Computing (STOC)*, pages 632–641, 1993.

[4] W. Aiello, E. Kushilevitz, R. Ostrovsky, and A. Rosén. Adaptive packet routing for bursty adversarial traffic. In *Proc. of the 30th ACM Symp. on Theory of Computing (STOC)*, pages 359–368, 1998.

[5] M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention-resolution protocols. In *Proc. of the 37th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 380–389, 1996.

[6] M. Andrews, A. Fernández, A. Goel, and L. Zhang. Source routing and scheduling in packet networks. In *Proc. of 42nd IEEE Symp. on Foundations of Computer Science (FOCS)*, October 2001.

[7] S. Arya and M. Smid. Efficient construction of a bounded degree spanner with low weight. In *Proceedings of 2nd European Symposium on Algorithms*, pages 48–59, 1994.

[8] F. M. auf der Heide, C. Schindelhauer, K. Volbert, and M. Grunewald. Energy, congestion and dilation in radio networks. In *Proceedings of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 230–237, August 2002.

[9] B. Awerbuch, P. Berenbrink, A. Brinkmann, and C. Scheideler. Simple routing strategies for adversarial systems. In *Proc. of the 42nd IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 158–167, 2001.

[10] B. Awerbuch, A. Brinkmann, and C. Scheideler. Anycasting in adversarial systems:routing and admission control. In *30th International Colloquium on Automata, Languages,and Programming (ICALP)*, 2003.

[11] B. Awerbuch and F. Leighton. A simple local-control approximation algorithm for multicommodity flow. In *Proc. of the 34th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 459–468, 1993.

[12] B. Awerbuch and F. Leighton. Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks. In *Proc. of the 26th ACM Symp. on Theory of Computing (STOC)*, pages 487–496, 1994.

[13] B. Awerbuch, Y. Mansour, and N. Shavit. End-to-end communication with polynomial overhead. In *Proc. of the 30th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 358–363, 1989.

[14] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LANs. In *Proc. of the 1994 SIGCOMM Conference*, pages 212–225, 1994.

[15] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson. Adversarial queueing theory. In *Proc. of the 28th ACM Symp. on Theory of Computing (STOC)*, pages 376–385, 1996.

[16] A. Colvin. CSMA with collision avoidance. *Computer Communications*, 6(5):227–235, 1983.

[17] D. Eppstein. Spanning trees and spanners. In J. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. Elsevier, Amsterdam, 2000.

[18] J. Freebersyser and B. Leiner. A DoD perspective on mobile ad hoc networks. In C. Perkins, editor, *Ad Hoc Networking*, pages 29–51. Addison-Wesley, 2001.

[19] D. Gamarnik. Stability of adversarial queues via fluid models. In *Proc. of the 29th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 60–70, 1998.

[20] D. Gamarnik. Stability of adaptive and non-adaptive packet routing policies in adversarial queueing networks. In *Proc. of the 31st ACM Symp. on Theory of Computing (STOC)*, pages 206–214, 1999.

[21] J. Gao, L. Guibas, J. Hershberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking & Computing*, October 2001.

[22] A. Goel. Stability of networks and protocols in the adversarial queueing model for packet routing. In *Proc. of the 10th ACM Symp. on Discrete Algorithms (SODA)*, pages 911–912, 1999.

[23] M. Grünewald, T. Lukovszki, C. Schindelhauer, and K. Volbert. Distributed maintenance of resource efficient wireless network topologies. In *8th International Euro-Par Conference*, August 2002.

[24] P. Gupta and P. Kumar. Capacity of wireless networks. *IEEE Transactions on Information Theory*, IT-46:388–404, 2000.

[25] Y. Hassin and D. Peleg. Sparse communication networks and efficient routing in the plane. In *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing*, pages 41–59, July 2000.

[26] IEEE. Medium access control (mac) and physical specifications. *IEEE P802.11/D10*, January 1999.

[27] H. H. III, M. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkrantz, and R. S. tearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Journal of Algorithms*, 26:238–274, 1999.

[28] L. Jia, R. Rajaraman, and C. Scheideler. On local algorithms for topology control and routing in ad hoc networks. Technical report NU-CCS-02-07, Northeastern University, September 2002. An electronic copy is available from http://www.ccs.neu.edu/home/lujunjia/toporouting.ps.

[29] P. Karn. MACA – A new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134–140, 1990.

[30] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of ACM Symposium on Mobile Computing and Networking*, pages 243–254, August 2000.

[31] L. Li, J. Halpern, V. Bahl, Y.-M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks. In *Proceedings of ACM Symposium on Principles of Distributed Computing*, pages 264–273, August 2001.

[32] X. Li, P. Wan, Y. Wang, and O. Frieder. Sparse power efficient topology for wireless networks. *Journal of Parallel and Distributed Computing*, 2002.

[33] C. Perkins. Ad hoc networking: An introduction. In C. Perkins, editor, *Ad Hoc Networking*, pages 1–28. Addison-Wesley, 2001.

[34] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of IEEE Infocom 2000*, March 2000.

[35] T. Rappaport. *Wireless Communications: Principles and Practices*. Prentice Hall, 1996.

[36] J. Salowe. Euclidean spanner graphs with degree four. *Discrete Applied Mathematics*, 54:55–66, 1994.

[37] C. Scheideler and B. Vöcking. From static to dynamic routing: Efficient transformations of store-and-forward protocols. In *Proc. of the 31st ACM Symp. on Theory of Computing (STOC)*, pages 215–224, 1999.

[38] M. Steenstrup. Cluster-based networks. In C. Perkins, editor, *Ad Hoc Networking*, pages 75–138. Addison-Wesley, 2001.

[39] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12:261–268, 1980.

[40] P. Tsaparas. Stability in adversarial queueing theory. Master's thesis, Dept. of Computer Science, University of Toronto, 1997.

[41] P. Varaiya and J. Walrand. *High-Performance Communication Networks*. Morgan Kaufmann, 2000. Chapter 7.

[42] Y. Wang and X. Li. Distributed spanner with bounded degree for wireless ad hoc networks. In *Parallel and Distributed Computing Issues in Wireless networks and Mobile Computing*, April 2002.

[43] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proceedings of IEEE Infocom*, 2001.

[44] A. Yao. On constructing minimum spanning trees in k-dimensional spaces and related problems. *SIAM Journal on Computing*, 11(4):721–736, 1982.