A Novel Topology Discovery Service for Self-Organized WSNs

Safwan Al-Omari and Weisong Shi
Wayne State University
{somari, weisong}@wayne.edu

Abstract

We propose a topology discovery service, which is a network service that promotes the concept of cross-layer design. The topology service, which is the fundamental of any self-organized and self-healing WSN, maintains several topology parameters that describe the sensor field and how the nodes are distributed in the field. To demonstrate the benefits of our service, we design a density-aware GPSR routing protocol (DA-GPSR) to take advantage of the parameters provided by our service and improve the performance of the original GPSR protocol. DA-GPSR uses node density to route around areas with high node density (i.e., crowded areas) and improves end-to-end performance of GPSR. We use TinyOS and TOSSIM for the implementation and simulation, respectively.

1 Introduction

Researchers believe that cross-layer design is the key to optimizing WSN communication stack protocol layers, and so overcome energy and computing limitations of wireless sensor nodes [3, 7]. In cross-layer design, some pieces of information at one layer are used to improve the performance of another layer in the communication stack. For example, a routing protocol can consider link quality provided by the MAC layer when selecting a path from a source to a destination to improve delivery rate. Likewise, a topology management protocol can take advantage of the node duty schedule maintained by the application to put the node into a full sleep mode –when idle– and save energy.

Parameters related to coverage and network topologies can also be used by protocol components to improve their performance. Sensor field dimensions, total number of nodes, node degree (i.e., number of nodes within the sensing/communication range), and node density in a specific sensor field area are typical examples of topology parameters. A routing layer, for example, may choose to route around areas with high node density to avoid high levels of collisions.

We propose a topology discovery service that maintains several topology parameters to support cross-layer design. To the best of our knowledge, we are the first to propose such a service that explicitly aims to support cross-layer design. The topology parameters are accessible by other protocol components through Sensor core (Score) [1], which is a cross-layer library that supports arbitrary interface between any protocol components without the need for explicit pair-wise APIs. The lack of pair-wise APIs allows protocol components running in the context of Score to maintain their modular design.

In addition to our topology discovery service contribution, we use the topology discovery service to design and implement the **D**ensity-**A**ware GPSR routing protocol (denoted as DA-GPSR), which uses node degree to route around areas with high node densities to improve end-to-end (E2E) performance of the original GPSR [4, 9] routing protocol.

The rest of the paper is organized as follows: Section 2 explains the topology discovery service including the parameters it provides in detail; Section 3 and Section 4 present the design and implementation of DA-GPSR and evaluation respectively; conclusion and future work are presented in Section 5.

2 Topology Discovery Service

On its own, the topology discovery service does not map into any of the OSI reference model layers and it represents a typical network service that solely supports cross-layer design. Other traditional network layers, such as MAC and routing layers, use the topology parameters maintained by the topology discovery service in order to improve their performance. In order to build and maintain these parameters, the topology discovery service actively sends and receives protocol messages, for example, neighboring nodes exchange their neighbor lists to find *communication redundancy* and *freshness*.

After calculating the topology parameters, the topology discovery service publishes them into *Score* so that other network components can access at will. It is vital to note

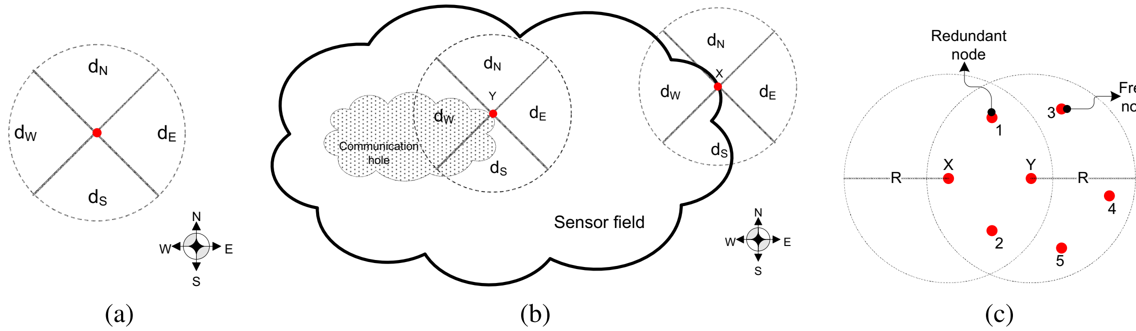


Figure 1. Illustrating topology parameters: (a) shows a sensor node and its directional density; (b) depicts a sensor field with a communication hole (shaded area), node X located on the boundary exhibits zero d_N and d_E. Node Y also exhibits very low d_W which suggests a communication hole in that direction; (c) depicts several nodes with their respective communication range R. X has a communication redundancy and freshness values of 2 and 3 with node Y respectively. Nodes 1 and 2 are redundant, while nodes 3 , 4, and 5 are fresh.

that the introduction of the topology discovery service does not impose sever restructuring of the communication stack. Thanks for *Score* that allows the network components to access the topology parameters without the need for direct interface with the topology discovery service.

It is important to differentiate topology discovery from neighbor discovery. Neighbor discovery mainly focuses on discovering a node's neighbor list, usually by simple probing messages. The topology discovery service builds on the neighbor list information to provide higher-level information with semantics, for example, what is a node's *communication redundancy* with each node in the node neighbor list.

2.1 Topology parameters

Topology parameters describe in general the sensor field including its physical dimensions (i.e., size) and the number of nodes deployed in that field. This deployment implies several other topology parameters including connectivity and coverage topologies, and the average node density in general as well as node density in different locations in particular. This deployment also dictates how many neighbors each node has in its neighbor list (i.e., node degree) on average and identifies nodes with above and below average node degree. The link quality value for a node with each one of its neighbors can also be considered as a topology parameter. Collectively, the topology parameters describe a node's topological (connectivity and coverage) relationship with its immediate neighbors as well as its topological position in the sensor field (i.e., on the boundary of the connectivity graph or on a boundary of a communication hole or high density area). Following is a list of these parameters:

Physical dimensions of the sensor field: describes the shape and dimensions of the sensor field. This information along with an estimation of the nominal communication range, can be used for example by a routing protocol to get an estimation of the longest (hop count) possible path between any two nodes in the sensor field. Such an estimation could be useful for a routing layer to recover from routing loops (i.e., when the path exceeds the maximum hop count).

Total number of nodes: specifies the total number of nodes the network has in total. It can be used, along with sensor field dimensions, to calculate deployment node density. A topology management protocol can use this as an indication of how aggressive the protocol should be when putting nodes into sleep.

Node density: could be either communication or sensing density, which is simply the number of neighbors in a nodes communication/sensing range divided by communication/sensing area. A node density can be used by a topology management protocol to maintain a specific number of active nodes at anytime.

Directional density: represents a node density in a particular direction (i.e., NORTH, SOUTH, EAST, WEST). Figure 1(a) explains the directional density. A large difference in a node's directional densities may suggest that the node is located on the topological boundary of the network, some communication/sensing hole, or high density area.

Boundary: a flag that indicates whether a node is located on the boundary of a sensor field area with irregular topological properties. An area is considered topologically irregular if the area has above average node density (high node density), or below average node density (communication hole). The boundary flag may also indicate that the node is located on the topological boundary of the network. Boundary nodes can collaborate to seize irregular areas. A routing protocol becomes aware of such areas and avoid routing through them, for example, routing around communication holes. Figure 1(b) depicts a sensor field with a communication hole area (i.e., shaded area), we can notice that a node on the boundary of the communication hole and a node on the sensor field boundary exhibit a variation in their directional densities.

Communication/sensing redundancy: a node's redundancy with each one of its neighbors describes the number of nodes that both of them share in their neighbor lists.

Communication/sensing freshness: a node's freshness with each neighbor is the number of nodes that exist in the neighbor's neighbor list and not in the node's neighbor list. Fig. 1(c) illustrates communication redundancy and freshness.

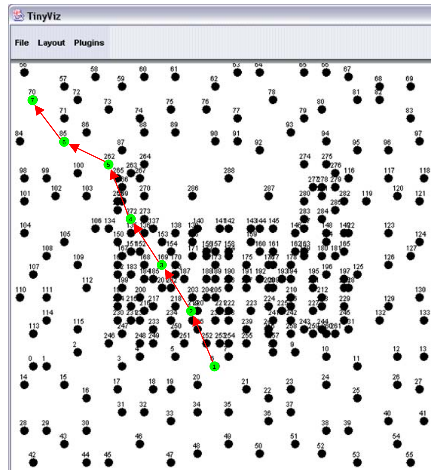
Both *communication redundancy* and *freshness* can be used for example in a controlled flooding protocol, where the total number of transmissions required to disseminate a message is minimized by selecting particular wireless links when forwarding a message [2].

3 Routing In Irregular Topologies

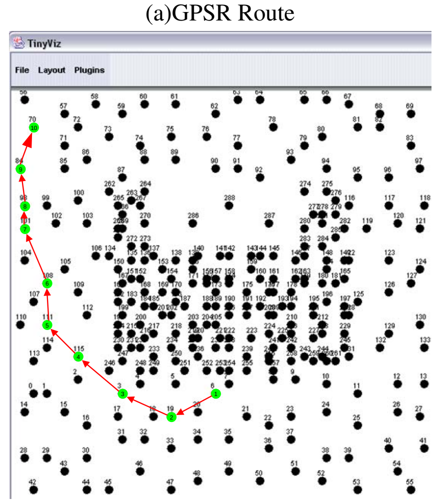
In this case study, we use the topology discovery service to develop a modified version of GPSR routing protocol to improve end-to-end (E2E) performance in irregular topologies. By irregular topology we mean a topology where wireless links are not uniformly distributed in the sensor field (i.e., some sensor field areas have high connectivity and so high interference levels, and areas with low connectivity and so lower interference levels). We argue that irregular topologies are commonplace in real life scenarios due to several reasons such as non-uniform node deployment, environmental conditions that make the nominal communication range variable at different locations in the sensor field, and node failures.

While routing from a source to a destination, DA-GPSR avoids areas with high node density in contrast to GPSR, which strictly routes directly toward the destination. The rational behind avoiding areas with high node density is that these areas are more susceptible to collisions and have higher levels of interference, which results in degradation in the one-hop success rate and ultimately degradation in E2E performance. As a result of avoiding areas with high node density, routes chosen by DA-GPSR tend to have higher hop count compared to routes chosen by GPSR. Nonetheless, in irregular topologies, using shortest hop-count routes obviously will not lead to the best E2E delivery rate possible as these routes may have to pass through areas with high interference levels. However, it may be not intuitive whether routing around those areas will result in better the E2E delay.

The basic idea of DA-GPSR is to augment the link quality of the next hop sensor node on the route to the destination in addition to the distance to the destination used by



(a)GPSR Route



(b) DA-GPSR Route

Figure 2. TOSSIM snapshot of routes chosen by GPSR and DA-GPSR.

GPSR. Two factors affect the link quality of a wireless link, SNR ratio and interference level. The former is controlled by environmental factors and we don't have much power to improve, while the latter is mainly controlled by the number of potential transmitters on a given wireless link. As presented in [8], interference has the larger effect on link quality and total network capacity. We employ interference definition of a wireless link introduced in [5], and further, define route interference level as the maximum and average interference level over wireless links on the route.

3.1 Network model

We assume a 2-dimensional sensor field with N sensor nodes distributed over the sensor field. All sensor nodes have the same transmission range r. If a node transmits a packet, all the nodes within a distance r of the transmitting node, which form the node's neighbor set, will receive the message. Since in this case study, we are interested in interference only, we assume a perfect pair-wise wireless channel (i.e., no packet losses due to signal fading). Finally, we assume that the relative physical location of the nodes is available to them by means of some localization protocol.

3.2 MAC protocol

In this subsection we describe the underlying MAC protocol used with GPSR and DA-GPSR. Our MAC protocol employs a TDMA channel access with packet acknowledgement and retransmissions. As discussed in [6], in a network with S transmission slots and d potential forwarding nodes, the probability that node i forwards a message successfully is defined as follows:

$$Pr(suc.) = \left(\frac{S-1}{S}\right)^d \quad (1)$$

Upon receiving a data message, the receiver node sends a short acknowledgement message back to the sender. If the sender node does not receive the acknowledgement within a round trip time, it assumes that the data message has been lost, and so retransmits the data message. The sender node continue on retransmitting until the message is successfully received or it reaches the maximum number retransmissions (**MAX_RET**), and so, give up. To avoid congestion control problem and keep the MAC protocol simple, each data message is allowed enough time to propagate from the source to the destination nodes. The single-hop success rate with retransmissions, therefore, is defined as follows:

$$Pr(suc.\ with\ ret.) = 1 - \left((1 - Pr(suc.))^{MAX\_RET}\right) \quad (2)$$

From the above equations, we can see that S, d, and **MAX_RET** are the parameters that control the single-hop success rate, and therefore, E2E performance. S and **MAX_RET** are MAC protocol configuration parameters, while d is determined by the route that the routing protocol (i.e., GPSR and DA-GPSR) chooses from a source to a destination nodes.

3.3 DA-GPSR

Let S_i be the neighbor set of node i, $|x, j|$ the Euclidean distance between node x and node j, and $|S_x|$ be the node degree of node x. At node i, we introduce two ranking values for each node $x \in S_i$ given a final destination node j. First, $dRank_{x,j}$ as the distance rank of neighbor x. Second, $degRank_x$ as the degree rank of neighbor x. $dRank_{x,j}$ ranks a node (i.e., x) based on its physical proximity to the destination (i.e., j), while $degRank_x$ ranks a node (i.e., x) based on its link quality with the current node (i.e., i). Interference of wireless link (i, x) equals to the summation of $|S_i|$ and $|S_x|$. Thus, for node i, the link quality with any of its neighbors boils down to the node degree of that neighbor (i.e., node degree of node i is the same for all neighbors). To bring distance rank and the degree rank into the same domain values, we normalize both values as follows:

$$degRank_x = \frac{\max_{m \in S_i}(|S_m|) - |S_x|}{\max_{m \in S_i}(|S_m|)} \quad (3)$$

$$dRank_{x,j} = \frac{\max_{m \in S_i}(|m, j|) - |x, j|}{\max_{m \in S_i}(|m, j|)} \quad (4)$$

To keep DA-GPSR a localized algorithm, we use local maximums over the neighbor set of the node to normalize $dRank_{x,j}$ and $degRank_x$.

$$Rank_x = \alpha \cdot dRank_{x,j} + (1 - \alpha) \cdot degRank_x \quad (5)$$

Under DA-GPSR forwarding rules, node i ranks the nodes in its neighbor set (i.e., S_i) according to formula 5 when forwarding to destination node j. Then, node i forwards the data message to the neighbor with the highest $Rank$ value. We use an α value of 0.7 in our simulation. Using an α value of 1.0 brings DA-GPSR to the original GPSR algorithm.

Using the interference level (IR) of a wireless link introduced in [5], we further define IR_{avg} and IR_{max} to describe a route quality. Let $route_{a,b}$ passes through nodes $\{x1, x2, ..., xn\}$. Then,

$$IR_{avg} = \frac{\sum_{i=1}^{n} |S_{xi}|}{n - 1} \quad (6)$$

$$IR_{max} = \max_{i=1}^{n} |S_{xi}| \quad (7)$$

Fig. 2 compares routes chosen by GPSR to routes chosen by DA-GPSR, GPSR tends to choose routes with less hop count, however, DA-GPSR chooses routes with lower IR_{avg} and IR_{max}. In Fig. 2(a) and Fig. 2(b), the GPSR route has a hop count value of 6, an IR_{avg} value of 29.8, and a IR_{max} value of 46, while the DA-GPSR route has a hop count value of 9, an IR_{avg} value of 10.9, and a IR_{max} value of 19. As we show in the next Section, the impact of interference level of a route outweighs the impact of hop count on the ultimate E2E performance.

4 Evaluation

We implement and evaluate DA-GPSR using TinyOS and TOSSIM. To avoid excessive long running time of

TOSSIM bit-level simulation, we use packet-level simulation with simulated collisions. Next, we present our performance metrics followed by simulation results in Section 4.1 and Section 4.2 respectively.

4.1 Performance metrics

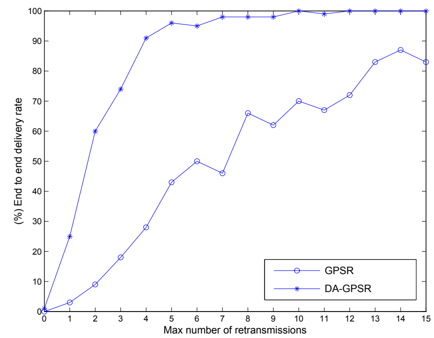
We use two E2E performance metrics to compare GPSR to DA-GPSR, which are defined as follows. First, **E2E delivery rate**, which is the ratio of the number of messages sent at the data source to the number of data messages received successfully at the destination node. Second, **E2E delay**, which is the time in milliseconds required to deliver a data message from the source to the destination.

4.2 Simulation results

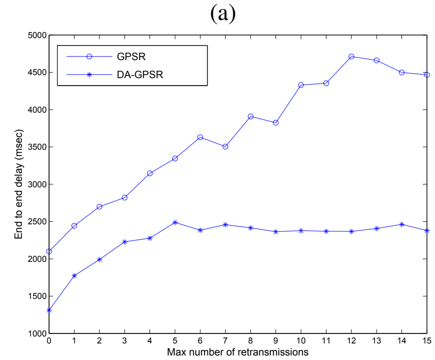
We perform low-level experiments, in which we study the effect of MAC protocol parameters (i.e., **S** and **MAX_RET**) configuration on the E2E performance of GPSR and DA-GPSR. These low-level experiments give us an insight on how DA-GPSR performs in contrast to GPSR given the same underlying MAC protocol settings. From the application point of view, the specific MAC level settings are not the primary concern. Instead, E2E delivery rate is what actually matters. Therefore, we further perform a high-level experiment that compares the performance of GPSR and DA-GPSR in terms of the the E2E delay given the same achievable E2E delivery rate.

In Fig. 3, we fix **S** to 24 and vary **MAX_RET** from 0 to 15. Then, for the same source/destination nodes, we send 100 data messages using GPSR and DA-GPSR and calculate the E2E delivery rate and delay in Fig. 3(a) and Fig. 3(b) respectively. From Fig. 3(a), we notice that E2E delivery rate equals to zero when **MAX_RET** equals to zero. This means that not a single data message arrived successfully at the destination, hence, the E2E delay for **MAX_RET** value of zero is not defined. Therefore, **MAX_RET** value of zero is removed from the domain in Fig. 3(b).

We can observe from Fig. 3(a) that E2E delivery rate improves very fast under DA-GPSR compared to GPSR. The Figure shows that for a **MAX_RET** value of 4, DA-GPSR already achieved over 90% E2E delivery rate, while GPSR is lagging behind with an E2E delivery rate value of 40%. Also, DA-GPSR shows more reliable E2E delivery rate performance compared to GPSR, in which, E2E delivery rate keeps fluctuating even for high **MAX_RET** values. Fig. 3(b) studies the effect of **MAX_RET** on the E2E delay in milliseconds. Intuitively, E2E delay is driven by two factors: the number of hops, including retransmissions, a message travels from source to destination (i.e., each retransmission is counted as one hop), and time delay of each hop, which is the same in GPSR and DA-GPSR (i.e., same



(a)



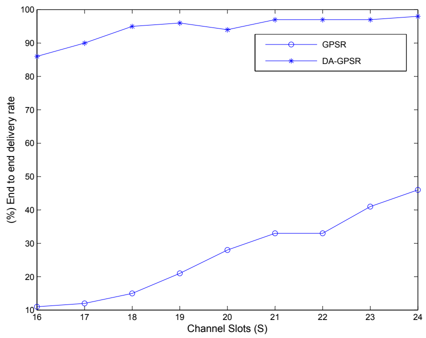
(b)

Figure 3. Effects of MAX_RET on GPSR and DA-GPSR. (S) is fixed to 24.

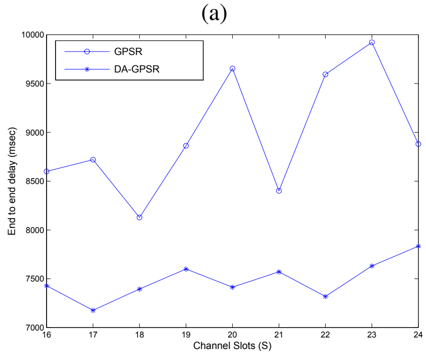
S and back off time). The figure suggests that route quality outweighs the advantage of having shorter physical hop count as each message loss and retransmission due to high interference results in an extra single-hop delay. As a result, choosing routes with less interference levels pays off and DA-GPSR outperforms GPSR even in terms of E2E delay.

In Fig. 4, we fix **MAX_RET** value to 7 and change **S** from 16 to 24 and run the same experiments as in Fig. 3 in order to study the relationship between **S** and E2E delivery rate and E2E delay, in Fig. 4(a) and Fig. 4(b) respectively. As evident in the figures, the effect of **S** on the performance of GPSR and DA-GPSR is not as big and consistent as that of **MAX_RET**. This is due to the fact that increasing **S** by one causes a very small increase in single-hop success rate (See formula 1). In Fig. 4(a), we observe that E2E delivery rate improves slightly as S increases. Also, it is very clear that DA-GPSR achieves much better E2E delivery rate for the same value of **S**.

Like Fig. 3(b), the relationship of **S** and the E2E delay is double-sided. Increasing **S** improves single-hop success rate, which decreases the need for retransmissions, and so decreases single-hop delay and hence, E2E delay. On the other hand, increasing **S**, increases single-hop delay as nodes need to back off for longer period of time on average before forwarding a message. However, since the effect of



(a)

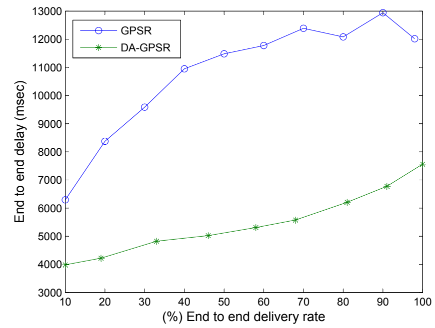


(b)

Figure 4. Effects of (S) on GPSR and DA-GPSR. MAX_RET is fixed to 7.

S on single-hop success rate is not that big, the second factor dominates. Therefore, increasing **S** has a net effect of increasing E2E delay.

In Fig. 5, We abstract **S** and **MAX_RET** into E2E delivery rate and compare the performance of GPSR and DA-GPSR in terms of E2E delay. As we mentioned, this is important to contrast the performance of GPSR to that of DA-GPSR given the same achievable E2E delivery rate. As we have no control to vary E2E delivery rate and observe the other performance metrics, we have to make several runs, in which we vary **S** from 16 to 24 and vary **MAX_RET** from 0 to 16 in order to get the full E2E delivery rate domain values (i.e., from 0% to 100%). For each run, we record E2E delay, then, we sort and group the runs based on E2E delivery rate. In each group, we calculate average value of E2E delay, and finally present the data in Fig. 5. The figure show that DA-GPSR outperforms GPSR, we observe that in order to achieve an acceptable E2E delivery rate value of 90% or above, GPSR needs around 12 seconds to deliver a single data message from source to destination, while DA-GPSR can deliver the same message in almost half the time.



(a)

Figure 5. Combining MAX_RET and S into E2E delivery rate and comparing GPSR to DA-GPSR.

5 Conclusion and Future Work

We present a novel network service that aims primarily at supporting cross-layer design. Using this service, DA-GPSR improves E2E delivery rate and reduces E2E delay by half. In our future work, we would like to propose more topology parameters and work on more case studies that use these parameters to enhance their performance.

References

[1] S. Al-Omari, J. Du, and W. Shi. Score: A sensor core framework for cross-layer design [extended abstract]. In *Proc. of QShine'06*, Aug. 2006.

[2] S. Al-Omari and W. Shi. A novel topology discovery serive for self-organized wsns. Technical Report MIST-TR-2006-007, Wayne State University, Apr. 2006.

[3] R. Bhatia and M. Kodialam. On power efficient communication over multi-hop wireless networks: joint routing, scheduling, and power control. In *Proc. of INFOCOM'04*, Mar. 2004.

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

[5] M. Burkhart, P. Rickenbach, R. Wattenhofer, and A. Zollinger. Does topology control reduce interference? In *Proc. of the MobiHoc'04*, May 2004.

[6] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. *IEEE Trans. on mobile computing*, July 2004.

[7] M. Chiang. To layer or not to layer: Balancing transport and physical layers in wireless multihop networks. In *Proc. of INFOCOM'04*, Mar. 2004.

[8] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 19(2):73–85, Feb. 2000.

[9] B. Karp and H. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proc. of MobiCom'00*, Aug. 2000.