

# BeamStar: An Edge-Based Approach to Routing in Wireless Sensor Networks

Shiwen Mao, *Member, IEEE*, and Y. Thomas Hou, *Senior Member, IEEE*

**Abstract**—Current expectations on sensor node in terms of size, cost, and energy efficiency have led to a severely limited design space on hardware and software. In this paper, we explore capabilities at the network edge for sensor networks, aiming to reduce the hardware and software complexity of a sensor node without sacrificing network performance. We present a novel edge-based routing protocol, nicknamed *BeamStar*, for wireless sensor networks. Under *BeamStar*, the base station exploits some nice properties associated with *directional antenna* and *power control* at the base station. We devise a simple protocol so that each sensor node can determine its location information passively with minimum control overhead. We also show how to design a robust routing protocol based on the location information at each sensor node. Under the proposed protocol, sensor nodes are relieved of the activities (or burdens) that are associated with control and routing, thus enabling much simpler hardware and software implementation at sensor nodes. Simulation results demonstrate that *BeamStar* achieves high reliability at comparable energy consumptions as compared with prior work. It is a viable approach to pursue size and cost reduction for future sensor node design.

**Index Terms**—Edge-based control, directional antenna, power control, localization, routing, wireless sensor networks.

## 1 INTRODUCTION

RESEARCH activities on wireless sensor networks have been nothing less than unprecedented in recent years. Numerous algorithms and protocols have been developed and some testbeds are now operational on a limited scale. However, what has been lagging behind in the progress is the size and cost of a sensor node. It has been envisioned that sensor nodes, with all the capabilities demonstrated today and new capabilities promised for tomorrow, will be on the scale of tiny dust or cubic-millimeter scale [1]. However, despite continued advances in microelectromechanical systems (MEMS), low-power VLSI, and computing, it remains a formidable task to implement many of the capabilities on a sensor node on such a tiny scale.

For example, based on the advanced 0.13- $\mu\text{m}$  complementary metal-oxide semiconductor (CMOS) technology, a memory block of 8 Kbytes (or, more precisely,  $512 \times 128$  bits) alone will occupy an area of  $0.22 \text{ mm} \times 1.4 \text{ mm}$  [2]. If we consider other components such as the processor, radio frequency (RF), mixed signal circuits, and so forth that will also be incorporated on a sensor node, the resulting size will become much bigger. For a sensor node packaged with all of these components on a cubic-millimeter scale, the storage, processing, and communication capabilities on the node will be seriously limited.

We observe that most of the current efforts on sensor network research have limited their design space solely to

the sensor nodes themselves. Under such an approach, the burden of achieving complex networking functions (for example, topology control, routing, localization, synchronization, and so forth) all rests upon the sensor nodes in the network core (that is, a *core-based paradigm*).<sup>1</sup> Such an approach relies heavily on future advances in silicon technology to reduce the size and cost of sensor nodes.

In this paper, we explore a new design space for sensor networks with the aim of reducing the complexity burden on a sensor node. We believe that, although it is important to continue pursuing novel algorithms and protocols to squeeze the most out of the existing design space (sensor nodes), it is equally important to explore other new design paradigms for future sensor networks. In this paper, we propose exploring capabilities at the network edge (that is, an *edge-based approach*). This approach taps into capabilities at the edge base station, which has not been fully exploited in prior efforts. This expanded design space has the potential to simplify various algorithms and protocols on a sensor node, thus offering new possibilities to drive down the size and cost of sensor nodes.

In particular, we present a novel edge-based routing protocol, called *BeamStar*, for wireless sensor networks. The motivation of *BeamStar* is based on the following observations: First, at a sensor node, it is highly desirable to utilize much of its limited resources (that is, hardware and energy) on sensing and data forwarding and to minimize the resource spent on the overhead due to control plane functions. Second, there is huge disparity in terms of capabilities and available resources in a sensor node and a base station. A sensor node is tiny and severely resource constrained, whereas a base station has few resource constraints. Further, the potential capabilities in a base station have not been properly exploited. It would be desirable to exploit such a disparity and shift the burden of

1. In this paper, we call the base station the “edge” mainly because it is the end point of data collection.

• S. Mao is with the Department of Electrical and Computer Engineering, Auburn University, 200 Broun Hall, Auburn, AL 36849. E-mail: smao@ieee.org.

• Y.T. Hou is with the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, 302 Whittemore Hall (0111), Blacksburg, VA 24061. E-mail: thou@vt.edu.

Manuscript received 25 Dec. 2005; revised 7 June 2006; accepted 5 Apr. 2007; published online 25 Apr. 2007.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0377-1205. Digital Object Identifier no. 10.1109/TMC.2007.1071.

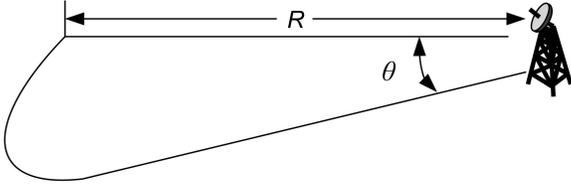


Fig. 1. A sector made by a power-controlled directional antenna transmission.

complex network control and management functions from a sensor node to a base station.

As an instance of this approach, in this paper, we show how some communication capabilities at the base station can be exploited. Specifically, we propose employing directional antennas and power control capabilities at a base station. By having the base station scan the network using a power-controlled directional antenna, each sensor node can uniquely obtain its location information. With such location information, sensor nodes in the network can route reports to the base station via *controlled broadcasting*. At each sensor node, forwarding decisions are made via a simple comparison of a received report with locally stored forwarding rules (also set by the base station). One assumption of BeamStar is that the sensors are within the transmission range of the base station so as to receive control messages from the base station. As a result, BeamStar works best for networks with line of sight (LOS). Extensions for highly cluttered environments and channel dynamics are discussed later in the paper.

We show that such an edge-based approach enables extremely simple hardware and software design on sensor nodes, since all the control functionalities are effectively shifted to the base station. There is no need for the sensor nodes to exchange control information for localization, synchronization, and routing, except for receiving several control messages *passively* from the base station to calibrate location information and configure forwarding rules during the initialization phase. Furthermore, data packets are relayed by sensor nodes within a constrained mesh (or sector) toward the base station. Each data packet may be copied and forwarded by multiple sensor nodes, yielding a better chance of successful delivery. The routing scheme is thus error resilient.

The rest of this paper is organized as follows: In Section 2, we present the basic idea of BeamStar. Practical considerations and various extensions are discussed in Section 3. In Section 4, we present simulation results to demonstrate the properties of BeamStar and compare it to a representative protocol following the core-based approach. Section 5 reviews related work, and Section 6 concludes this paper.

## 2 THE BASIC PROTOCOL

Consider a wireless sensor network consisting of a large number of sensors deployed over a wide region. We assume that sensors are randomly deployed in the network with a proper density to guarantee a connected network (for example, the “magic number” of six to eight neighbors as given in [3] and [4]). Should there be an event, sensor reports must be relayed hop by hop to the base station. In the following, we present the key components of BeamStar

Control Message:

BaseID	SN	RN	SeqNum	Other control information
--------	----	----	--------	---------------------------

Sensor Report Message:

BaseID	SourceID	LastRelayID	Timestamp	Data
--------	----------	-------------	-----------	------

Fig. 2. Packet formats under BeamStar.

and discuss its properties. We will first discuss BeamStar under LOS environments in this section and will consider the effects of obstacles later in Section 3.

### 2.1 Base-Station-Assisted Location Discovery and Synchronization

In many applications, sensor data must be associated with location and time (that is, *where* and *when* the event was first detected) to make it usable. This requires efficient localization and synchronization schemes, in addition to a routing protocol that will deliver data back to a base station. For large-scale sensor networks, fully distributed protocols, although technically plausible, may result in substantial control overhead (for example, message exchanges for localization, synchronization, or maintaining a routing table or gradients) or high requirements on a sensor node (for example, CPU, storage, or memory), which will not help in reducing the size and cost of a sensor node.

Our approach in this paper is to explore potential infrastructure support by an edge base station to perform intelligent location discovery and synchronization. In particular, we assume that the base station is not energy constrained and is equipped with a *directional antenna* with *power control* capability [5]. With such a directional antenna, the covered area of each base station transmission is a sector, as illustrated in Fig. 1. The radius of the sector  $R$  is determined by the transmit power, whereas the span of the sector  $\theta$  is determined by the beamwidth of the directional antenna. We assume that the base station can adjust its transmit power to reach all nodes in the network.

Since each sensor can directly receive control messages from the base station, it can easily synchronize its local clock with that of the base station. To determine the relative position of a node (or group of nodes) to the base station, the base station scans the entire region using different power levels. In other words, the base station sends control messages containing the current directionality information and the transmit power level (indicating the relative distance toward the base station) through successive scans of the network. A node’s location is thus determined by the directionality of the last base station transmission (called *Sector Number (SN)*), as well as the lowest power level that it can receive from the base station (called *Ring Number (RN)*). We therefore define the ID of a sensor to be the 2-tuple  $\{SN, RN\}$ .

Control messages broadcast by the base station can have a format as shown in Fig. 2. The *BaseID* field carries the identifier of the base station, which will be useful when multiple base stations are used (see Section 3.1). The *SN* field carries the index associated with the direction of the current transmission, and *RN* is the index associated with the current transmit power. The field *SeqNum* is the sequence number of the current scan. Additional control information can also be carried in the packet. When a sensor

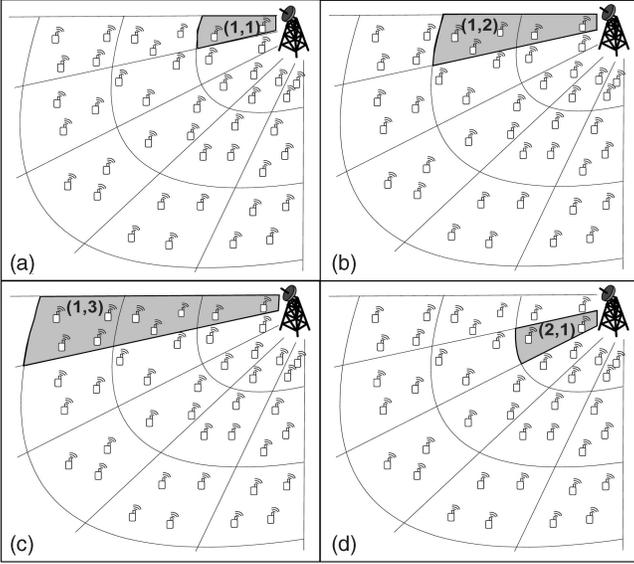


Fig. 3. An illustration of the base-station-assisted location discovery process: (a) the first transmission with  $\{SN = 1, RN = 1\}$ , (b) the second transmission with  $\{SN = 1, RN = 2\}$ , (c) the third transmission with  $\{SN = 1, RN = 3\}$ , and (d) the fourth transmission with  $\{SN = 2, RN = 1\}$ . All these transmissions have the same  $SeqNum$  value, which is increased by 1 after the base station completes scanning the entire network.

$k$  receives multiple control messages with the same  $SeqNum$  (see Fig. 3), its ID is chosen as

$$\begin{cases} SN_k = \max\{SN_i | i \in \{\text{rcvd ctrl msgs with } SeqNum\}\} \\ RN_k = \min\{RN_i | i \in \{\text{rcvd ctrl msgs with } SeqNum\}\} \end{cases} \quad (1)$$

Note that an ID with a higher  $SeqNum$ , that is, obtained from a subsequent base station scan, may overwrite a lower  $SeqNum$  ID, thus allowing reconfiguration of the network.

Such a base-station-assisted location discovery process is illustrated in Fig. 3, where one base station serves as the sink node for the entire network. In Fig. 3a, the base station transmits to the first region (shaded area) by adjusting its phase angle and transmit power. During this transmission, the directional antenna broadcasts a message with  $\{SN, RN\} = \{1, 1\}$  to all nodes within this region. Sensors located in this region will then obtain their ID as  $\{SN, RN\} = \{1, 1\}$ . Subsequently, in Fig. 3b, the base station adjusts its transmit power to the next higher level and broadcasts a message with location information  $\{SN, RN\} = \{1, 2\}$ . Those sensor nodes that have acquired ID  $\{1, 1\}$  from the earlier transmission will not change their ID according to (1). All other nodes covered by the second transmission will store this ID, that is,  $\{SN, RN\} = \{1, 2\}$ , to represent their geographical locations. Figs. 3c and 3d show the case for two subsequent transmissions with  $\{SN, RN\} = \{1, 3\}$  and  $\{SN, RN\} = \{2, 1\}$ , respectively. By repeating this procedure, all sensor nodes can derive their IDs (that is, location information) via receiving broadcast control messages from the base station. Throughout the process, there is no need for any sensor node to exchange any information with its neighbors. The complexity on the control plane has been effectively shifted from sensor nodes to the base station.

Note that BeamStar requires that the entire sensor network be within the maximum transmission range of

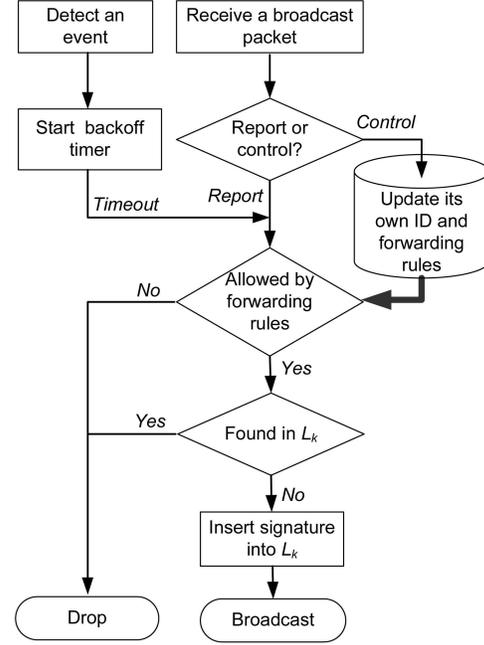


Fig. 4. Flowchart of BeamStar operation.

the base station, and it works best in LOS environments. For sensor networks that span vast areas, multiple base stations may be necessary to provide possibly overlapped coverage for the sensor nodes. We will discuss the case of multiple base stations and other network environments in Section 3.1.

## 2.2 Location-Aware Data Forwarding

We now discuss routing and forwarding under BeamStar. The flowchart in Fig. 4 illustrates this process. When an event occurs in the network, it may be detected simultaneously by multiple sensors. To avoid unnecessary redundancy, we only need to have one sensor node to generate a report for this event. Ideally, we would hope to have the sensor node that is closest to the event to generate this report. This requirement can be implemented by having each sensor set a timer upon detecting an event, where the time-out value is set to be inversely proportional to the detected signal strength. Therefore, among all the neighboring sensors that have detected the same event, the sensor with the highest signal strength will time out first and broadcast the corresponding report to its neighbors. Those sensors that have detected the same event (with a live timer ticking) will cancel the timer and the scheduled transmission (which carries the same timestamp).

The format of a sensor report message is shown in Fig. 2. The header fields are

- *BaseID*. This is the identifier of the destination base station (in the case of multiple base stations).
- *SourceID*. This is the 2-tuple identifier, that is,  $\{SN, RN\}$ , of the source sensor node.
- *LastRelayID*. This is the identifier of the last sensor node that forwarded this report.
- *Timestamp*. This is the time instance when the event was detected (set by the source node).

To avoid forwarding the same report multiple times, each sensor  $k$  maintains a short *signature list*  $L_k$

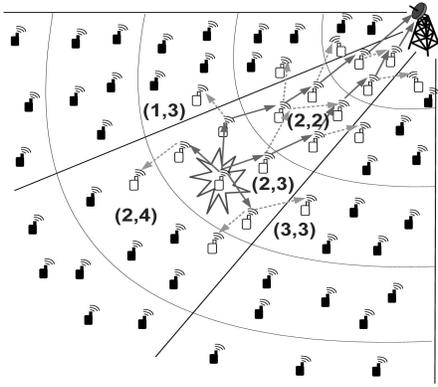


Fig. 5. An illustration of data forwarding under BeamStar upon detection of an event. Sensor nodes involved in the process are marked in white.

for recently forwarded reports. Note that the 3-tuple  $\{BaseID, SourceID, Timestamp\}$  uniquely identifies a sensor report because each sensor will generate at most one report at any given time instance. By using  $BaseID$ , a source node can send the same report to multiple base stations if it chooses to.

As shown in Fig. 4, upon receiving a report, the sensor makes forwarding decisions by making two comparisons. If the last relay of this report is from a different region, it will first compare the report's  $LastRelayID$  value with its locally stored *forwarding rules*, which are set by the base station (for example, during the initialization scanning phase and possibly piggybacked in control messages). The forwarding rules contain one or more IDs of its neighboring regions, for example,

$$\{\{SN_k^i, RN_k^i + 1\}, \{SN_k^i + 1, RN_k^i\}, \{SN_k^i - 1, RN_k^i\}\}.$$

In the general case, such forwarding rules may contain IDs for regions farther away from the base station or for regions that are not even a direct neighbor (for example, in obstacle-rich environments). If no match is found, this report will be dropped. If a match is found (or the last relay is from the same region), the sensor will make a second comparison by searching the signature of the received report in  $L_k$ . If no match is found, the new signature will be inserted into  $L_k$  (the oldest signature may be discarded from  $L_k$  if the list is full), and this report will be rebroadcast. Otherwise, this report has already been forwarded before and will thus be dropped.

We use the example in Fig. 5 to illustrate the routing process. Suppose that an event is detected by a sensor in a region with ID  $\{2, 3\}$ . This sensor will generate a report and broadcast the report to its neighbors. Upon receiving this report, the neighboring sensors will further broadcast the report to their neighbors until, eventually, all nodes within this region receive this report. Note that the use of  $L_k$  ensures that each sensor node will broadcast this report at most once. At the four boundaries of this region, the forwarding decision is made with a simple comparison of the locally stored *forwarding rules* and the ID of the last relay (that is,  $LastRelayID$ ) as discussed. With the default forwarding rule, only sensors in region  $\{2, 2\}$  will relay this report. Reports received by nodes in the other three neighboring regions, that is, regions with IDs  $\{2, 4\}$ ,  $\{1, 3\}$ , and  $\{3, 3\}$ , will be dropped. This is due to the fact that

relaying in these regions will not bring the packet closer to the base station. By following this process of broadcast, reception, and drop or rebroadcast, the report (or multiple copies of the report) will eventually be delivered to the base station.

Note that this approach is similar, in some sense, to location-aided routing (LAR) [6]. However, *precise* node location information is used in LAR for forwarding decisions, whereas only *coarse* region IDs are used in BeamStar. The region-based approach can greatly reduce the sensor node complexity.

### 2.3 BeamStar Properties

There are many advantages of using this simple routing scheme. In BeamStar, intermediate sensor nodes do not need to store a routing table or maintain flow state information. There is even no need for a sensor to discover its one-hop neighbor nodes. Each node operates in a purely stateless manner using only local information (that is, its ID, the forwarding rules, and the signature list, which come from received messages). The operations and, thus, the requirements on sensor node design are minimal, enabling size and cost reduction on hardware.

The computation complexity for forwarding each sensor report is low. Upon receiving a report, the sensor node  $k$  first compares the  $LastRelayID$  of the report with its forwarding rules. The signature of this received report may also be compared with those in the list  $L_k$ , with at most  $|L_k|$  comparisons. The maximum length of  $L_k$  is a small number and remains constant for various networks. Overall, the computational complexity is  $O(1)$  in forwarding a report.

Second, the storage requirement is also minimal. With BeamStar, each node only needs to store its own ID, a short signature list, and simple forwarding rules, which suffice for robust routing. The total storage requirement for a sensor node is also  $O(1)$ , regardless of network size or density.

Third, the control overhead of BeamStar consists of the initial base station scan that delivers IDs and forwarding rules to sensor nodes and periodic connectivity tests during operation. For the initial scan, each sensor node only needs to receive at most  $N_r$  control messages from the base station, where  $N_r$  is the total number of rings in the network (or the total number of power levels of the directional antenna). For the latter, the base station will direct a query to the sensors in the outermost region in each sector to trigger a report being forwarded hop by hop back to the base station (see Section 3.5). As a result, each sensor will receive the "test" report once and broadcast it once. Such connectivity tests could be performed from time to time to check the "health" of the network (for example, detecting connectivity holes). Such checks are similar in spirit to many existing protocols that periodically flood control messages to set up and refresh the gradient field in the network (see, for example, [7] and [8]). Note that, whenever there is a report received from a sector during the last time window, it implicitly indicates that all the sectors between the base station and that sector are operational.

## 3 EXTENSIONS

In the last section, we described the basic operations of BeamStar. We can view this basic scheme as one end of the

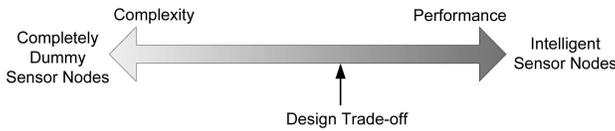


Fig. 6. Performance-complexity trade-off in sensor node design.

spectrum for sensor node design, as illustrated in Fig. 6. In practice, we do not need to limit the sensor nodes to be completely dumb. Some additional complexity in hardware and software, if allowed within the size and cost constraints, can be implemented on sensor nodes to increase their intelligence and capabilities. BeamStar is complementary to many existing core-based approaches from this perspective.

Fig. 6 illustrates the design space for sensor nodes. To extend the BeamStar idea under various circumstances, a balanced approach between complexity and performance can be developed. In this section, we discuss several possible extensions and advanced operations under this spirit.

### 3.1 Multiple Base Stations

BeamStar can be easily extended to support multiple base stations for larger sensor networks. The basic operation of the sensor nodes are similar to what was discussed before with only a slightly increased storage requirement.

A sensor node within the ranges of multiple base stations will choose a unique ID for itself based on messages from each of these base stations. When the sensor detects an event, it can simply choose the closest base station (by comparing the  $RN$ s of its multiple IDs) to send the report. An intermediate sensor node, after receiving the report, will first check the  $BaseID$  value in the report header (which was set by the source node). Then, the node makes a comparison and drop/forward decision using its ID and forwarding rules associated with the chosen base station.

A refined scheme can also be applied to balance the traffic load and energy consumption of the sensors. Suppose that a source sensor is under the coverage of  $M$  base stations. It can choose a base station  $i$  as its report destination based on a probability  $p_i$ , which is defined as follows:

$$p_i = \frac{\frac{1}{RN_i}}{\sum_{j=1}^M \frac{1}{RN_j}}. \quad (2)$$

Such a probabilistic decision for a base station will help distribute traffic evenly within the network. Note that a source node can also send the same report to multiple base stations (each with a different  $BaseID$  value) if it chooses to.

### 3.2 Granularity Control of Region Size

Under the single base station model, the grid size tends to be large for a region far away from the base station (see Fig. 3). Although multiple base stations could help improve region granularity in the network, it is still desirable to explore other approaches to control the region size and granularity.

In addition to manipulating the power increment of directional antennas, we could also manipulate beam orientation to achieve better granularity control. Note that,

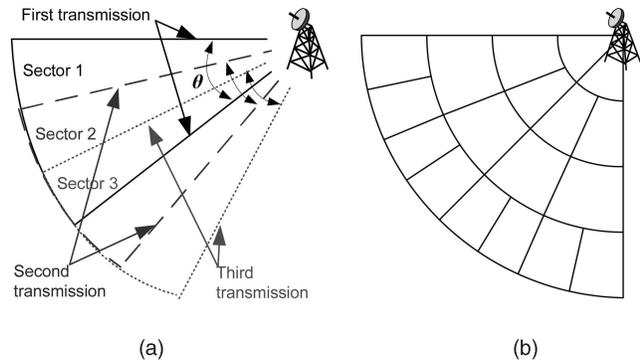


Fig. 7. Techniques to create smaller regions. (a) Creating sectors with a span one third of the minimum beamwidth. (b) Nonuniform region partitioning for different power levels.

although beamwidth can be adjusted, there is a limit due to the physical properties of directional antennas [5]. However, beam orientation can be exploited to refine region partition. Fig. 7 illustrates this concept. Let the antenna beamwidth be  $\theta$ . The base station makes three *overlapping* transmissions sequentially by shifting its directional angle by  $\theta/3$  each time. A sensor receiving multiple transmissions derives its ID according to (1). As a result, three sectors will be created, effectively reducing the span from  $\theta$  to  $\theta/3$ . In Fig. 7b, we show that it is even possible to partition an unequal number of regions that are at different distances from the base station.

### 3.3 Wake Up on Demand

Although we focus on developing robust routing protocols in this paper, it is worth noting that BeamStar can also work with sensor-specific media access control (MAC) protocols to further conserve energy. For example, since sensors may consume much power in the idle mode [9], [10], we could put sensor nodes to “sleep” after the initialization phase. When there is a need to receive or transmit data, the dual radio design in [9] can be used to wake up a sleeping sensor. In order to return to the sleep mode, each sensor starts a keep-alive timer when it wakes up and refreshes the timer each time when a new report is received or transmitted. When the timer expires, the sensor will go back to sleep. This wake up on-demand scheme is especially useful for sensor networks monitoring rare events.

As another example, the popular sensor-MAC (S-MAC) protocol could also be used with BeamStar [11]. The base station could assist in forming the virtual clusters for sleep scheduling or even directly schedule the duty cycles for the sensor nodes in a centralized manner. This is possible since all the sensors can receive such control messages from the base station and they are also synchronized with the base station.

### 3.4 Query for Events

For many applications, it is important for the base station to be able to query the sensor network (as a distributed database) for information, in addition to passively waiting for event reports. This function can be easily supported by BeamStar. To query a specific region, the base station can adjust its beam direction according to the target  $SN_i$ , set its transmit power level according to the target  $RN_i$ , and then transmit a query message carrying the ID of

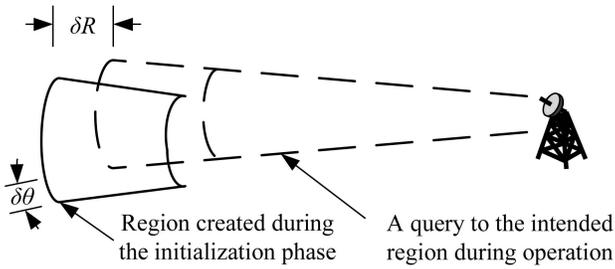


Fig. 8. Potential mismatch between a query transmission and the network partition.

the target region (or sensors)  $\{SN_i, RN_i\}$ . By a simple comparison of the IDs, sensors in other regions will ignore this query, whereas sensors in the target region will generate response reports and route them to the base station through multihop relays.

Due to channel dynamics or environment changes, the covered region by a query from a directional antenna with the specific transmit angle and power level may not completely overlap with the target region (created during an earlier initialization phase). This is illustrated in Fig. 8. We now show that such a mismatch problem is easy to handle for query purposes. Note that what is really needed is to convey the query message to the sensor nodes in the target area. This can be done by having the base station send multiple queries based on its estimation of directional error  $\delta\theta$  and distance error  $\delta R$ , that is, with transmit powers corresponding to  $R \pm \delta R$  and angles  $\theta \pm \delta\theta$ . In other words, there is really no need for the covered region by the directional antenna to be exactly identical with the target region. If a query report is needed from *every* sensor node in the target region, then we can adjust the directional antenna beam to be large enough to cover the intended target region so that every sensor node in that region can hear the query. On the other hand, if a query report from *any* sensor in the target region is sufficient, then a partial coverage by the directional antenna beam on the target region would do the job.

### 3.5 Connectivity Test and Reorganization

Even for a densely deployed sensor network, it is still possible to have a “hole” or “blind spot” where there is inadequate coverage or where a significant number of sensors have died out. It is important to detect these holes in a timely fashion so that necessary actions can be taken to maintain network connectivity and sensing coverage. Many existing routing protocols adopt periodic flooding of control messages to set up new paths [7] or refresh the gradient field in the network [8].

In BeamStar, a quick connectivity test can be made by having the base station send diagnostic queries to the outermost region in each sector. If a corresponding response is received by the base station within a time window, it infers that all regions within that sector are functioning properly (see Fig. 9a). Otherwise, we can have the base station send a diagnostic query to various spots in the sector (for example, using a binary search technique) to pinpoint the specific region(s) that have lost connectivity (see Fig. 9b).

Once a hole is detected in the network, it is necessary to take action to mitigate its impact. In addition to

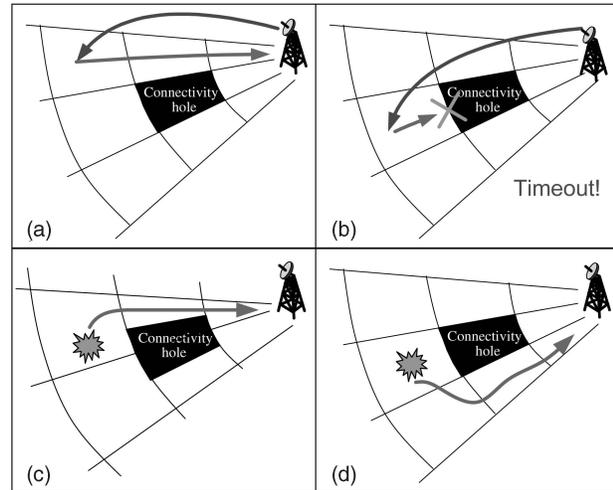


Fig. 9. Detecting connectivity holes in the network: (a) the sector functions properly and (b) a hole is detected with a time out. Mitigation of connectivity holes: (c) repartitioning the regions and (d) reprogram routing behavior to get around the hole.

replenishing new sensor nodes, BeamStar can reconfigure the network to restore connectivity with the following two simple approaches. One approach is to have the base station reconfigure the logical structure of the network so that the hole can be split and merged into the new grid structure (see Fig. 9c). Although the same physical hole remains in the network, it will no longer block packet routing for other (outer) regions in the network as it would have before the reconfiguration.

A second approach is to have the base station change the routing behavior of the neighboring regions. For example, the base station could send out new forwarding rules to sensor nodes in the neighboring regions of the hole, instructing them to relay reports that are otherwise blocked by the hole (see Fig. 9d). Note that, under this approach, a base station only needs to instruct a small number of cells to change their routing configuration; the routing behavior of the remaining cells is intact.

### 3.6 Issues Related to Directional Antennas

#### 3.6.1 Side-Lobe Suppression

A practical consideration associated with directional antennas is the potential presence of side lobes. In general, the radiation pattern of a directional antenna has several side lobes in addition to the main lobe. If untreated, such side lobes could have an undesirable effect on the cells under their coverage.

In practice, adaptive side-lobe cancellers are widely used to eliminate the spatially coherent interference received in the side lobes of a primary antenna. This effect could be achieved by dynamically adjusting the weights for a small set of antenna elements in order to minimize the residual output from the side lobes. In directional antenna design, such side lobes can be effectively suppressed by increasing the number of elements in the antenna array, by adjusting the spacing of and the type of antennas used in the array, and by advanced signal processing [5].

#### 3.6.2 Propagation Considerations

Another practical consideration is the nonideal (that is, nonfree space) characteristics of the wireless propagation

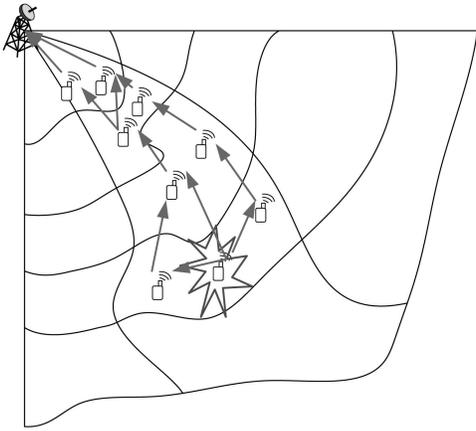


Fig. 10. Irregular regions due to obstacles, heterogeneous path losses, and multipath propagation.

environment. Generally, a directional channel is more reliable than an omnidirectional antenna channel [12] due to more focused transmissions and the absence of mobility. If the environment is close to free space [13], the basic approach will provide accurate location information to the sensor nodes. However, if the environment is heavily cluttered, sensor locations might be less accurate due to 1) obstacles and various propagation conditions within different sectors and 2) multiple receptions caused by multipath reflections.<sup>2</sup>

When the base station scans the network, the contour of a ring will be indented behind an obstacle. Likewise, if path loss at different directions is different, the contour of the rings will also be irregular. For multipath reflections, a reflected control message from a lower numbered sector (or in the exactly opposite LOS direction) will be ignored due to (1). If a sensor receives a control message reflected from an obstacle in a higher numbered sector, those sensors lying in between this sensor and the reflecting obstacle will also get this control message, under the assumption that signal propagation is continuous. As a result, these sensors will set their *Sector Number* to the higher one received from this reflection. Thus, multipath reflections from a higher numbered sector will make the sector partitions irregular along the reflection trajectory.

The joint effect of these two nonideal characteristics is that the regions partitioned by the directional antenna scan will be irregular, as shown in Fig. 10. However, BeamStar does not require regular partitioning of the network. This is quite different from those efforts on localization [13], [16], where the goal is to provide accurate location information for each sensor node. Rather, BeamStar partitions the network into coarse grain regions, which allows sensor reports to be forwarded to the base station, region by region, as illustrated in Fig. 10.

We envision three approaches to further improve the BeamStar performance under cluttered environments. The first approach is to adopt multiple base stations in obstacle-rich environments. In a recent work [17], Niculescu and Nath show that, using several base stations with revolving directional antennas, the localization error can be

2. Note that, in sparse networks or when obstacles are relatively large, the so-called “dead-end” problem may occur, which is general in geographical routing [14]. Many effective solutions proposed in the literature can also be applied here [14], [15].

driven down to a couple of meters, even in an indoor environment. It is expected that the accuracy could be even better for outdoor environments when multiple base stations are used.

The second approach is to conduct propagation measurements in the field. There has been considerable existing and ongoing work on measuring radio path loss for indoor [18] and outdoor environments [18], [19]. Based on the measured path loss, the base station can associate its transmission power levels with the corresponding distance in a more accurate manner.

The third approach (if the sensing area is inaccessible) is to use accurate geospatial information to predict path loss and channel propagation. Such highly accurate geospatial information can be obtained from some public database libraries [20]. There are several existing software tools for radio propagation prediction and network planning (for example, SitePlanner from Wireless Valley Communications [21] and WinProp from AWE Communications [22]). Equipped with such geospatial information, the base station can make an accurate calibration for the distortion caused by multipath reflections and obstacles and to adjust its location, beamwidth, and transmit power levels accordingly to mitigate these effects.

## 4 SIMULATION STUDIES

In this section, we present simulation studies for the proposed BeamStar protocol. The objective of this simulation study is twofold. First, we will show how the various design parameters affect the performance of BeamStar. Second and perhaps more importantly, we will compare BeamStar with a representative core-based approach—Directed Diffusion (DD) [7]. We will show that BeamStar achieves a comparable or better performance than a core-based approach but with greatly reduced hardware and software requirements on a sensor node. As a result, the size and cost of a sensor node can be substantially reduced.

### 4.1 Simulation Models and Performance Metrics

#### 4.1.1 Simulation Models

We implement the BeamStar protocol and perform simulations using OPNET Modeler [23]. For the results reported in this paper, two networks are simulated: 1) a 256-node network and 2) a 400-node network. Both networks are uniformly deployed over a 500 m × 500 m field. As in [8], we let one base station stay at a corner of the field and one source node be located at the diagonal corner.

Our sensor node implementation is illustrated in Fig. 11, which has a four-layer protocol structure. The sensor application module consists of a constant-bit-rate source, which generates a report every 100 ms (64 bytes each). For the routing layer, we implement the basic BeamStar protocol as described in Section 2 and study the routing protocol performance in the simulations.

For comparison purposes, we also implement DD [7]. Specifically, the original DD (or the two-phase pull diffusion) as described in [7] is used in our simulations. The main reason for choosing DD is that it is one of the most popular core-based routing protocols for wireless sensor networks. Under DD, a sink first broadcasts *exploratory interests* to find sources. Sources that match the interest reply with low-rate reports to establish paths toward the

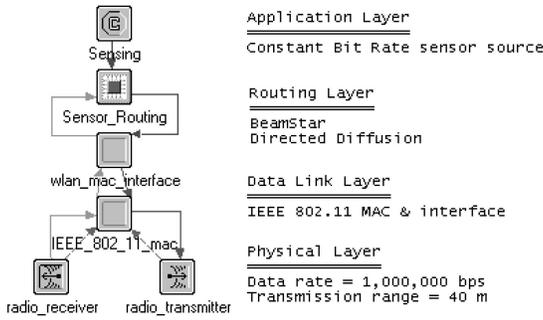


Fig. 11. Sensor node implementations in the simulation study.

sink, and the sink uses positive and negative reinforcements to select or prune parts of the path. All messages are 64 bytes in length.

BeamStar works with a very simple delayed broadcast MAC scheme. However, in order to be consistent with prior work [7], [8], we use IEEE 802.11 in the MAC layer. Specifically, delayed broadcast is used for BeamStar communications, which is less reliable than the 802.11 distributed coordination function (DCF) MAC used by DD. The data rate of the wireless channel is 1 megabit per second (frequency hopping). The radio transmission range is set to 40 m [7]. For consistency, we use the same energy consumption model as in [7] and [8]. The transmit, receive, and idle power consumptions are 0.66 W, 0.395 W, and 0.035 W, respectively. We count all types of energy consumption in the simulations, including transmission, receiving, idling, overhearing, collisions, and other unsuccessful transmissions, MAC layer headers, retransmissions, and request to send/clear to send/acknowledgments (RTS/CTS/ACKs).

Network dynamics such as link failure and node failure are also considered in our simulation model. Link failures are modeled as an on/off process with exponentially distributed on/off periods. When the link is in the “on” state, there is no packet loss; when the link is in the “off” state, a received packet is dropped with probability  $p_l$  ranging from 0 to 40 percent. The average “on” and “off” times are all set to 10 sec. Node failures are also modeled using similar on/off processes, but when the node is in the “off” state, it will stop sending or receiving packets (and thus does not consume any energy). The average “on” time is 10 sec, whereas the average “off” time varies from 0 to 1.111 sec, giving an average node failure probability  $p_n$  from 0 to 10 percent. These failure models allow us to examine the performance of the protocols under random and bursty packet losses.

#### 4.1.2 Performance Metrics

In our simulations, the following performance metrics are used [7], [8], [24]:

- *Successful delivery ratio.* This is the fraction of unique reports that are successfully received by the base station.
- *Average communication energy.* This is the total communication energy consumption, including transmitting, receiving, retransmissions, overhearing, collision, and other failed transmissions of both report and control messages, over the total number of distinct reports received at the base station.

- *Average energy.* This is the ratio of total energy consumption, including communication energy consumption and *idle energy consumption*, over the total number of distinct reports received at the base station.
- *Average report delay.* This is the average end-to-end delay over all distinct reports received at the base station.
- *Control message overhead.* This is the ratio between the total number of control messages received and transmitted and the total number of distinct reports received at the base station.

To make a fair comparison with DD in [8], we normalize the energy and control message overhead over the number of successfully delivered reports, since DD generates fewer reports than BeamStar during its exploratory phase.

In all the figures presented in this section, each data point is the average of 10 simulations with different random seeds. Each simulation run is at least 1,200 simulated sec, and some of the simulations run for a much longer time in order to get stable results. In all figures, we plot the 95 percent confidence interval for each data point.

## 4.2 BeamStar Design Parameters

In this section, we examine the impact of the system parameters, including the maximum broadcast delay  $T_{max}$ , the total number of rings  $N_r$ , and the total number of sectors  $N_s$  (or the beamwidth  $\theta$ ) on the BeamStar performance. BeamStar is largely a broadcast-based routing protocol. Although the MAC protocol complexity is greatly reduced, its performance may be more susceptible to node density, as compared with the unicast IEEE 802.11 MAC protocol. When multiple nodes sense channel idle, they will start transmission simultaneously and thus cause collision. In our protocol, we let a sensor delay transmitting a packet for a random period of time in order to avoid such collisions. The back-off delay is uniformly generated in the interval  $[0, T_{max}]$ .

As a result,  $T_{max}$  is a parameter that affects the reliability, end-to-end delay, and jitter of the proposed protocol. In Fig. 12, we plot the successful delivery ratio versus  $T_{max}$  and the average report delay versus  $T_{max}$  for the 256-node and 400-node networks without link/node failures.<sup>3</sup> In Fig. 12a, when  $T_{max}$  is very small, there are some packet losses due to broadcast collisions. However, the successful delivery ratio quickly approaches 1 for increased  $T_{max}$  in both cases. In Fig. 12b, the average delay slightly increases with  $T_{max}$  due to larger back-off delays in every hop along the path. We also find that the average delay is relatively higher when  $T_{max} = 0$  ms. This is because, in the presence of collisions, the report forwarded along the shortest path may be lost due to collision; the first received copy of a unique report may have been forwarded along a longer path. In all the following simulations, we set  $T_{max}$  to 2 ms.

In Fig. 13, we examine the successful delivery ratio and the average communication energy versus the total number of rings (or power levels) for the 400-node network. In the simulations, we set the total number of sectors ( $N_s$ ) to 12 and increase the total number of rings  $N_r$  from 1 to 11, thus changing the height of each region from 530.3 m to 48.2 m.

3. In the case of link/node failures, the chance of collision will be smaller.

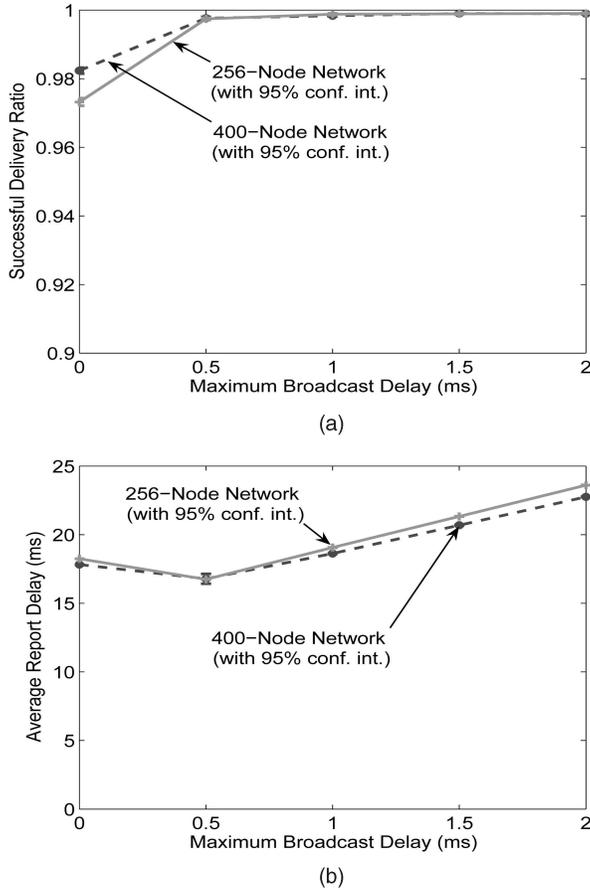
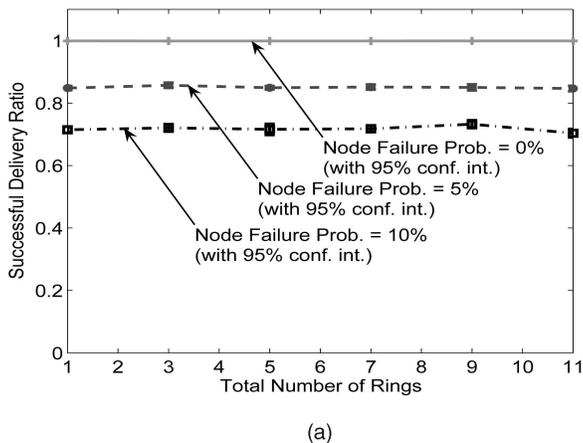


Fig. 12. (a) Successful delivery ratio versus maximum broadcast delay  $T_{max}$  and (b) average report delay versus maximum broadcast delay  $T_{max}$ .

The three curves in each figure correspond to node failure rates  $p_n = 0$  percent,  $p_n = 5$  percent, and  $p_n = 10$  percent, respectively. In Fig. 13a, we observe that various  $N_r$  values give the same successful delivery ratio. This is because, according to the default forwarding rule, all the sensors on the inner side of the source node will forward its reports no matter how the regions are partitioned. However, we find



in Fig. 13b that the average communication energy generally decreases with larger  $N_r$ s. This is because, generally with finer partitions, sensors on the outer side of the source may be partitioned to a different region and will not forward reports for the source anymore.

An interesting observation in Fig. 13b to make is that the average communication energy becomes slightly larger when  $N_r$  increases from seven to nine. When  $N_r$  is seven, the source node is located closer to the outer boundary of its region. When  $N_r$  is nine, although each region is smaller, the source node is located very close to the inner boundary of its region. As a result, more sensor nodes on the outer side of the source node are involved in forwarding its reports, thus resulting in a higher communication energy cost.

In Fig. 14, we examine the impact of the total number of sectors on the BeamStar performance. We set the total number of rings to  $N_r = 7$  and increase the beamwidth from 10 to 60 degrees, thus changing  $N_s$  from 36 to 6. The three curves in each figure correspond to node failure rates of  $p_n = 0$  percent,  $p_n = 5$  percent, and  $p_n = 10$  percent, respectively. In Fig. 14a, we find that, as the sectors get larger, higher successful delivery ratios are achieved, since more copies of the same report will be forwarded by the sensor nodes in the same sector. However, Fig. 14b clearly shows that reliability improvement comes at the cost of higher energy consumption for each received report. Therefore,  $N_s$  (or  $\theta$ ) provides a trade-off between reliability and energy cost. In practice, we can explore this fact to dynamically adjust  $N_s$  according to the monitored successful delivery ratio during network operation.

### 4.3 Comparison with DD

In this section, we compare BeamStar with DD under the same link failure or node failure conditions. For BeamStar, the parameters are  $N_r = 7$ ,  $N_s = 12$ , and a constant report rate of 10 reports/sec. For DD, the exploratory data rate is set to 1 report/sec. Once the path is reinforced, the data rate is increased to 10 reports/sec. The sink broadcasts an exploratory interest to refresh the gradients in sensor nodes every 100 sec, which is found to be effective for the failure models used in the simulations.

Fig. 15 presents simulation results for various benchmark performance metrics as a function of the node failure

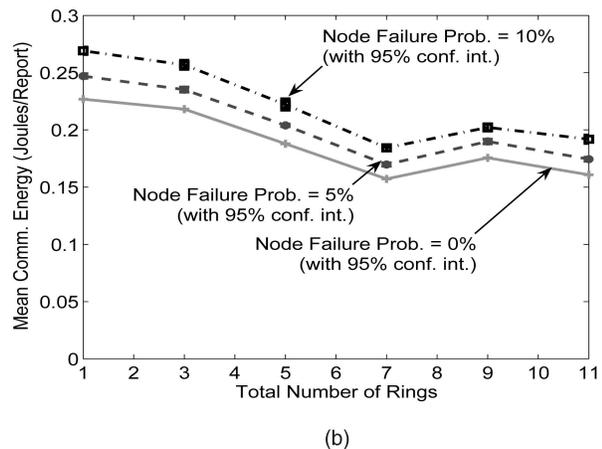


Fig. 13. (a) Successful delivery ratio versus the total number of rings  $N_r$  and (b) average communication energy versus the total number of rings  $N_r$  for the 400-node network.

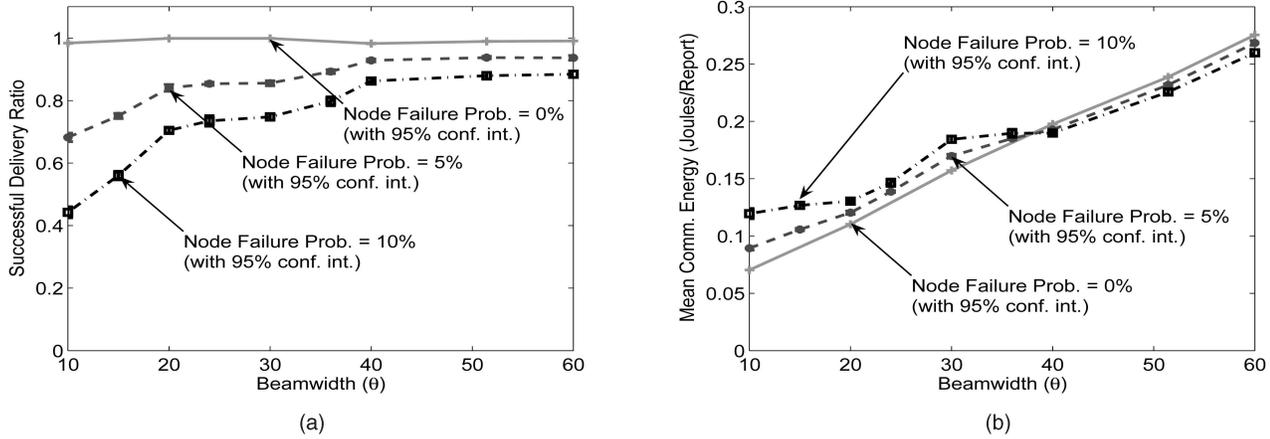


Fig. 14. (a) Successful delivery ratio versus beamwidth  $\theta$  (in degrees) and (b) average communication energy versus beamwidth  $\theta$  (in degrees) for the 400-node network.

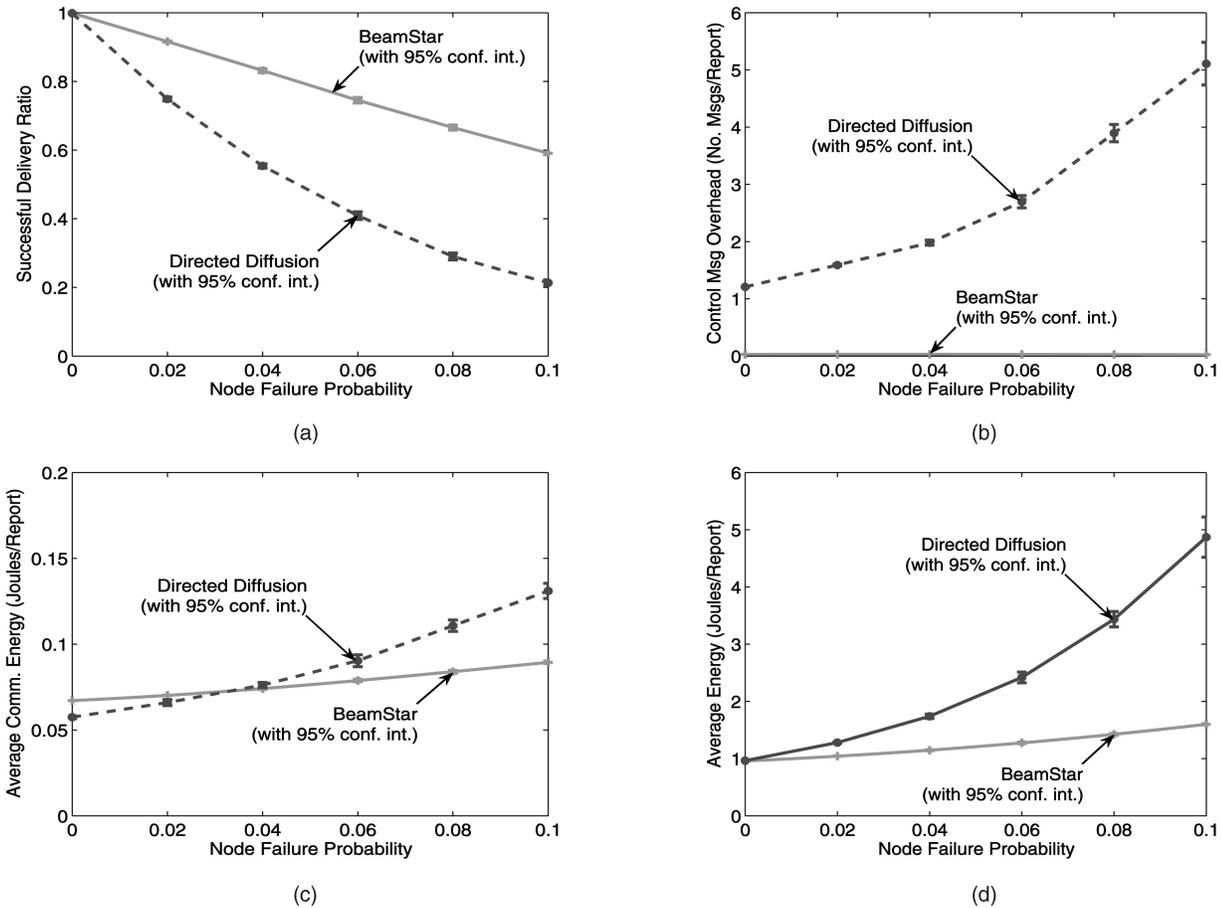


Fig. 15. Comparison of BeamStar with DD for the 256-node network. (a) Successful delivery ratio. (b) Control message overhead. (c) Average communication energy. (d) Average total energy.

probability for the 256-node network. The successful delivery ratios are plotted in Fig. 15a. For the loss-free case, both protocols achieve close to 100 percent delivery. As  $p_n$  increases, however, the BeamStar curve becomes consistently higher than the DD curve, indicating that forwarding packets via a broadcast mesh (BeamStar) is much more effective than unicast routing (DD). In addition, the control message overhead of BeamStar is relatively constant and negligible when compared to DD, as can be

seen from Fig. 15b. For a higher node failure probability, fewer reports will be received, resulting in increased DD control message overhead since it is normalized over number of received reports.

In Fig. 15c, we show the average communication energy consumption for the two protocols. We find that the average communication energy is comparable for the two protocols for low node failure probabilities. As  $p_n$  increases, the DD average communication energy becomes larger, whereas

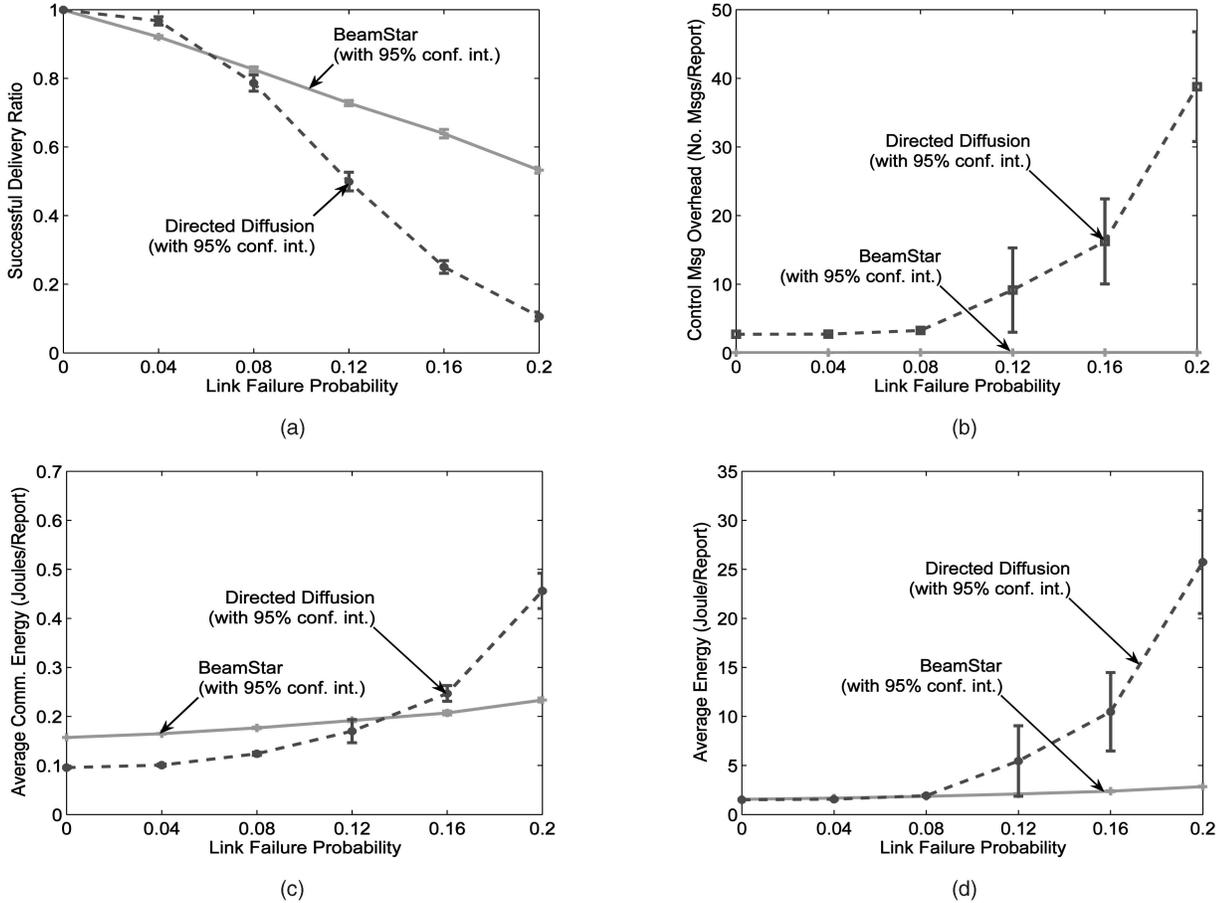


Fig. 16. Comparison of BeamStar with DD for the 400-node network. (a) Successful delivery ratio. (b) Control message overhead. (c) Average communication energy. (d) Average total energy.

the BeamStar average communication energy increases at a slower rate. This is, in fact, due to the different routing mechanisms used in these protocols. With unicast routing, DD achieves reliability by using RTS/CTS to reserve the channel and uses the absence of ACK for retransmissions, which all consume additional energy. On the other hand, BeamStar achieves reliability by a constrained broadcast. Although each broadcast transmission is less reliable, forwarding reports using a mesh is more effective in combating high node failure rates. More reports are received when BeamStar is used and, thus, the average communication energy value is reduced.

We also plot the average total energy consumption (including the idle energy consumption) in Fig. 15d. We find that the DD curve is above the BeamStar curve and increases quickly with  $p_n$ . This is because, in DD, the entire network is involved in the gradient establishment phase, whereas in BeamStar, routing is largely constrained within the sector of the source node. In addition, fewer reports are received in the DD case, resulting in increased average energy consumption. This effect is more obvious for higher node failure rates. Using more intelligent sensor MAC protocols that can schedule idle sensors to sleep (for example, S-MAC [11]), it is expected that the idle energy consumption (and, thus, the total energy consumption) could be effectively reduced for both protocols, especially for low event rates.

In Fig. 16, we perform a simulation study for the 400-node network. We observe similar trends from these figures. An interesting observation is that, for low link failure rates, DD achieves a higher successful delivery ratio than BeamStar (see Fig. 16a), which is different from what we have seen in Fig. 15a. This is due to the nature of failure models used in the simulations: The node failure model is used in Fig. 15a to generate *bursty* losses, whereas the link failure model is used in Fig. 16a to generate *random* losses. For infrequent random losses, DD's link layer retransmissions are quite effective, thus achieving a highly successful delivery ratio. In high loss environments, both reports and control messages can be lost. As a result, the DD curve quickly falls to low values as  $p_l$  increases.

Finally, we compare the delay performance of the two protocols for the 400-node network. Fig. 17a shows the average delay curves for various link loss rates, whereas Fig. 17b shows the average delay curves for various node failure rates. In both figures, the average delay of BeamStar stays around 22 ms due to its simple broadcast-based routing. The DD average delays are larger than the BeamStar average delays in both cases and the DD average delay increases with both the link failure rate and the node failure rate due to increased number of hop-by-hop retransmissions. Similar observations are made from simulation results for the 256-node network. We omit those delay figures for brevity.

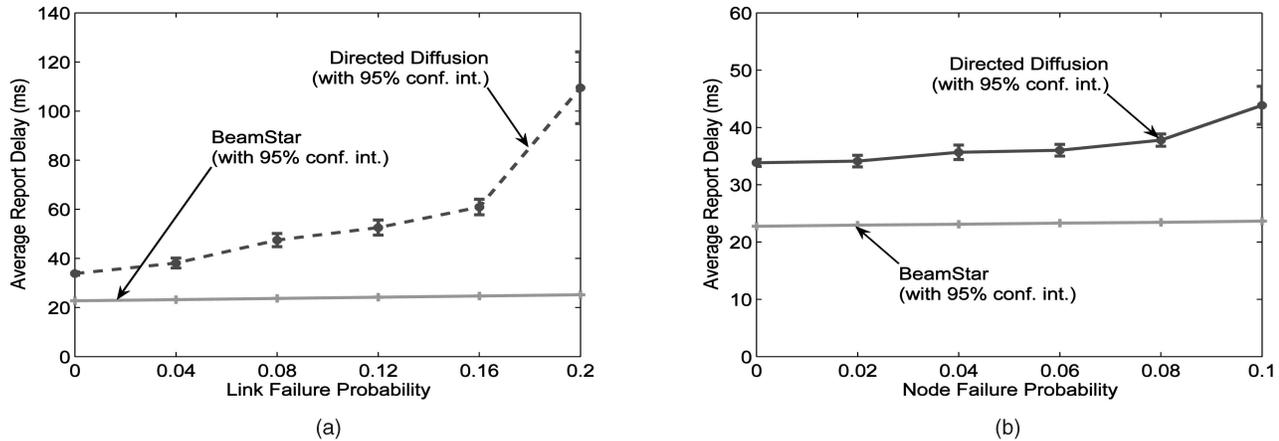


Fig. 17. Average end-to-end delay for the 400-node network. (a) Average report delay versus link failure. (b) Average report delay versus node failure.

## 5 RELATED WORK

Directional antennas have been explored in the context of ad hoc networks. Different problems pertaining to directional antennas have been studied, such as directional MAC [25], [26], directional routing [27], [28], capacity [29], [30], and energy efficiency [31], [32]. Note that all of these efforts consider using a directional antenna at a mobile node in the network core. However, it is not feasible to use directional antennas at sensor nodes due to stringent space and cost constraints. Instead, under our approach, directional antennas are only used at the *network edge*, whereas sensor nodes in the network core still use compact and inexpensive omnidirectional antennas.

A directional antenna has been used for localization in wireless networks [16], [17]. In [16], Nasipuri and Li present a localization scheme where three or more directional antennas are used to scan the network in a synchronized manner, and each sensor node computes its location using the angle-of-arrival estimation technique. In [17], Niculescu and Nath present a very high frequency (VHF) Omnidirectional Ranging (VOR) scheme for indoor positioning. This scheme employs three base stations with rotating directional antennas, and the angles and ranges from the three base stations are used to determine the location of a mobile node. These papers have a similar objective of reducing the hardware/software complexity of sensor nodes. However, there are two important differences. First, the focus in [16] and [17] is on accurate localization for an individual sensor node, whereas the focus of BeamStar is to partition the network into coarse grain regions so as to facilitate simple and robust data routing. Second, in these two schemes, each sensor measures the times, or angles and ranges, when it receives different beacon signals to calculate its location. In BeamStar, a sensor simply receives a control message from the base station, which carries its location information.

In a recent work [13], Romer proposes using lasers at a base station (called “Lighthouse”) to scan the network so that smart dust nodes can estimate their physical location information. Such an approach is limited by the LOS requirement of laser beams (discussed in [13, Section 4.5.5]). In addition, the focus of Lighthouse is to compute accurate location for sensors, and routing is not explicitly considered. Further, in Lighthouse, each sensor measures the time difference between laser beams and computes its own location. In BeamStar, no such measurements and computa-

tions are needed. Nevertheless, the lighthouse scheme motivates us to explore edge capabilities.

## 6 CONCLUSIONS

In this paper, we presented BeamStar, a novel edge-based routing protocol for wireless sensor networks. The proposed protocol was motivated by reducing hardware and software complexity at a sensor node so as to achieve size and cost reduction. The main idea is to shift some of the communication and processing intensive functions from sensor nodes to the base station. Specifically, we exploited the capabilities of directional antennas and power control at the base station to assist localization, synchronization, routing, and (potentially) many other complex tasks. As a result, the functions of each sensor node can be made much simpler than existing approaches, enabling considerable cost and size reduction on sensor nodes.

We present extensive simulation results to demonstrate the performance of the proposed BeamStar protocol in the context of routing. We found that BeamStar achieves higher reliability at comparable energy cost as compared to a representative core-based routing approach. Therefore, BeamStar represents a viable approach to deliver similar or better performance while enabling a much simpler hardware and software design (and thus size and cost reduction) at sensor nodes. Although our focus in this paper is on the routing performance, it is conceivable that such an edge-based paradigm can be exploited to simplify other complex tasks in sensor networks, such as code distribution, programmability, and tasking and query. These are interesting problems worth further investigation.

## ACKNOWLEDGMENTS

The authors wish to thank Professor Samir Das, the *TMC* editor of this paper, and the anonymous reviewers for their constructive comments. They also thank Dr. Min Chen of the University of British Columbia and Dr. Fan Ye of the IBM T.J. Watson Research Center for their help with the simulations. This research was supported in part by the US National Science Foundation (NSF) under Grant CNS-0347390 and the US Office of Naval Research (ONR) under Grant N00014-05-1-0481. This work was done while Shiwen Mao was with Virginia Polytechnic Institute and State University, Blacksburg.

## REFERENCES

- [1] B. Warneke, M. Last, B. Leibowita, and K. Pister, "Smart Dust, Communicating with a Cubic-Millimeter Computer," *Computer*, vol. 34, no. 1, pp. 44-51, Jan. 2001.
- [2] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, third ed. Addison-Wesley, 2005.
- [3] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Trans. Comm.*, vol. 32, no. 3, pp. 246-257, Mar. 1984.
- [4] T. Hou and V. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Trans. Comm.*, vol. 34, no. 1, pp. 38-44, Jan. 1986.
- [5] T. Rappaport, *Smart Antennas: Adaptive Arrays, Algorithms, and Wireless Position Location*. IEEE Press, 1998.
- [6] Y. Ko and N. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad-Hoc Networks," *Wireless Networks*, vol. 6, no. 4, pp. 307-321, 2000.
- [7] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Trans. Networking*, vol. 11, no. 1, pp. 2-16, Feb. 2003.
- [8] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," *ACM/Springer Wireless Networks*, vol. 11, no. 2, pp. 285-298, Mar. 2005.
- [9] C. Shurgers, V. Tsiatsis, S. Generiwal, and M. Srivastava, "Topology Management for Sensor Networks: Exploiting Latency and Density," *Proc. ACM MobiHoc*, pp. 135-145, June 2002.
- [10] A. Savvides, C.-C. Han, and M. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," *Proc. ACM MobiHoc*, pp. 166-179, July 2001.
- [11] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 3, pp. 493-506, June 2004.
- [12] J. Polastre, R. Szewczyk, A. Mainwaring, and D. Culler, "Lessons from a Sensor Network Expedition," *Proc. First European Workshop Wireless Sensor Networks*, pp. 307-322, Jan. 2004.
- [13] K. Romer, "The Lighthouse Location System for Smart Dust," *Proc. ACM First Int'l Conf. Mobile Systems, Applications, and Services (MobiSys '03)*, pp. 15-30, May 2003.
- [14] B. Karp and H. Kung, "Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. ACM MobiCom*, pp. 243-254, Aug. 2000.
- [15] X. Ma, M. Sun, G. Zhao, and X. Liu, "Improving Geographical Routing for Wireless Networks with an Efficient Path Pruning Algorithm," *Proc. IEEE Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON '06)*, pp. 246-255, Sept. 2006.
- [16] A. Nasipuri and K. Li, "A Directionality Based Location Discovery Scheme for Wireless Sensor Networks," *Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA '02)*, pp. 105-111, Sept. 2002.
- [17] D. Niculescu and B. Nath, "VOR Base Stations for Indoor 802.11 Positioning," *Proc. ACM MobiCom*, pp. 58-69, Sept./Oct. 2004.
- [18] G. Durgin, T. Rappaport, and H. Xu, "Measurements and Models for Radio Path Loss and Penetration Loss in and Around Homes and Trees at 5.85 GHz," *IEEE Trans. Comm.*, vol. 46, no. 11, pp. 1484-1496, Nov. 1998.
- [19] H. Xu, R. Boyle, T. Rappaport, and J. Schaffner, "Measurements and Models for 38 GHz Point-to-Multipoint Radiowave Propagation," *IEEE J. Selected Areas in Comm.*, vol. 18, no. 3, pp. 310-321, Mar. 2000.
- [20] Virginia Polytechnic Inst. and State Univ. (Virginia Tech) Center for Geospatial Information Technology (CGIT), "Online Geospatial Maps," <http://www.cgit.vt.edu/resources.htm#freedata>, 2007.
- [21] Wireless Valley Communications, "SitePlanner," <http://www.wirelessvalley.com/Products/SitePlanner/SitePlanner.asp>, 2007.
- [22] AWE Communications, "WinProp," <http://www.awe-communications.com/>, 2007.
- [23] OPNET Technologies, "OPNET Modeler," <http://www.opnet.com/products/modeler/home.html>, 2007.
- [24] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A Taxonomy of Wireless Micro-Sensor Network Models," *ACM Mobile Computing and Comm. Rev.*, vol. 6, no. 2, pp. 28-36, Apr. 2002.
- [25] X. Yang, N. Vaidya, and R. Ramanathan, "Using Directional Antennas for Medium Access Control in Ad Hoc Networks," *Proc. ACM MobiCom*, pp. 59-70, Sept. 2002.
- [26] L. Bao and J. Garcia-Luna-Aceves, "Transmission Scheduling in Ad Hoc Networks with Directional Antennas," *Proc. ACM MobiCom*, pp. 23-28, Sept. 2002.
- [27] R. Choudhury and N. Vaidya, "Ad Hoc Routing Using Directional Antennas," technical report, Univ. of Illinois, Urbana-Champaign, Aug. 2002.
- [28] R. Choudhury and N. Vaidya, "Impact of Directional Antennas on Ad Hoc Routing," *Proc. Eighth Int'l Conf. Personal Wireless Comm. (PWC '03)*, pp. 590-600, Sept. 2003.
- [29] R. Ramanathan, "On the Performance of Ad Hoc Networks with Beamforming Antennas," *Proc. ACM MobiHoc*, pp. 95-105, 2001.
- [30] S. Yi, Y. Pei, and S. Kalynaraman, "On the Capacity Improvement of Ad Hoc Wireless Networks Using Directional Antennas," *Proc. ACM MobiHoc*, pp. 108-116, June 2003.
- [31] J. Wieselthier, G. Nguyen, and A. Ephremides, "Energy-Limited Wireless Networking with Directional Antennas: The Case of Session-Based Multicasting," *Proc. IEEE INFOCOM*, pp. 190-199, June 2002.
- [32] A. Spyropoulos and C. Raghavendra, "Energy Efficient Communications in Ad Hoc Networks Using Directional Antennas," *Proc. IEEE INFOCOM*, pp. 23-27, June 2002.



**Shiwen Mao** received the BS and MS degrees in electrical engineering from Tsinghua University, Beijing, in 1994 and 1997, respectively. He received the MS degree in system engineering and the PhD degree in electrical and computer engineering from Polytechnic University, Brooklyn, New York, in 2000 and 2004, respectively. He was a research member at IBM China Research Lab, Beijing, from 1997 to 1998 and a research intern at Avaya Labs-Research, Holmdel, New Jersey, in the summer of 2001. He was a research scientist in the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, from December 2003 to April 2006. Currently, he is an assistant professor in the Department of Electrical and Computer Engineering, Auburn University, Alabama. His research interests include cross-layer design and optimization in multihop wireless networks, as well as multimedia communications. He was a corecipient of the 2004 IEEE CS Leonard G. Abraham Prize in the field of communications systems. He is currently serving as the editor of the *Advances in Multimedia Journal*. He is the coauthor of *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004). He is a member of the IEEE.



**Y. Thomas Hou** received the BE degree from the City College of New York in 1991, the MS degree from Columbia University in 1993, and the PhD degree from Polytechnic University, Brooklyn, New York, in 1998, all in electrical engineering. Since Fall 2002, he has been with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, where he is now an associate professor. His current research interests are in radio resource (spectrum) management and networking for software-defined radio wireless networks, optimization and algorithm design for wireless ad hoc and sensor networks, and video communications over dynamic ad hoc networks. From 1997 to 2002, he was a researcher at Fujitsu Laboratories of America, Sunnyvale, California. He was a recipient of an Office of Naval Research (ONR) Young Investigator Award (2003) and a US National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award (2004). He is active in professional services and is currently serving as an editor of the *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Vehicular Technology*, *ACM/Springer Wireless Networks (WINET)*, and *Elsevier Ad Hoc Networks Journal*. He is TPC cochair of the Second International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM '07). He was the chair of the First IEEE Workshop on Networking Technologies for Software Defined Radio Networks (2006). He will serve as cochair of the TPC of IEEE INFOCOM '09. He holds two US patents and has three more pending. He is a member of the ACM and a senior member of the IEEE.