

# SAND: Sectored-Antenna Neighbor Discovery Protocol for Wireless Networks

Emad Felemban  
Computer Engineering Department  
Umm Al Qura University  
Makkah, Saudi Arabia  
eafelemban@uqu.edu.sa

Robert Murawski,  
Eylem Ekici  
Department of Electrical  
and Computer Engineering  
Columbus, OH  
(murawskr, ekici)@ece.osu.edu

Sangjoon Park, Kangwoo Lee,  
Juderk Park, Zeeshan Hameed  
ETRI  
Daejeon, South Korea  
(sangjoon, stn, zdpark, zhmir)@etri.re.kr

**Abstract**—Directional antennas offer many potential advantages for wireless networks such as increased network capacity, extended transmission range and reduced energy consumption. Exploiting these advantages, however, requires new protocols and mechanisms at various communication layers to intelligently control the directional antenna system. With directional antennas, many trivial mechanisms, such as neighbor discovery, become more challenging since communicating parties must agree on where and when to point their directional beams to enable communication.

In this paper, we propose a fully directional neighbor discovery protocol called Sectored-Antenna Neighbor Discovery (SAND) protocol. SAND is designed for sectored-antennas, a low-cost and simple realization of directional antennas, that utilize multiple limited beamwidth antennas. Unlike many proposed directional neighbor discovery protocols, SAND depends neither on omnidirectional antennas nor on time synchronization. In addition, SAND performs neighbor discovery in a serialized fashion allowing individual nodes to discover all potential neighbors within a predetermined time. Moreover, SAND guarantees the discovery of the best sector combination at both ends of a link, resulting in more robust and higher quality links between nodes. Finally, SAND gathers the neighborhood information in a centralized location, if needed, to be used by centralized networking protocols. The effectiveness of SAND has been assessed via simulation studies and real hardware implementation.

**Index Terms**—wireless ad hoc networks, sectored antennas, directional antennas, neighbor discovery.

## I. INTRODUCTION

Performance improvement through directional antennas in wireless networks has been discussed extensively in the literature [1], [2], [3], [4]. Directional antennas, as opposed to omnidirectional antennas, concentrate the transmission power towards a certain direction with limited beamwidth around this direction. As a result, directional antennas lend many promising features to wireless networks. First, interference between neighboring nodes is greatly reduced which increases the simultaneous transmissions in the neighborhood and network capacity [5]. Second, signal to interference and noise ratio (SINR) is increased due to higher directional gain allowing transmissions a higher data rates. Finally, for a given transmission power, the communication range is greatly extended.

This work has been performed while Emad Felemban was at the Ohio State University

Alternatively, lower transmission power is needed to cover the same transmission range covered by an omnidirectional antenna.

One of the simplest realizations of directional antennas is sectored antennas where a fixed number of fixed beamwidth antenna elements are mounted to cover the whole azimuth. One major advantage of sectored antennas compared to other types of directional antennas is its low-cost and implementation simplicity. Switching sectors is done by simply selecting an antenna element as shown in Figure 1.

Protocol design for wireless networks with directional antennas is a challenging problem due to the problems related to directional antennas such as “Directional Hidden Terminal” problem [6] and the “Deafness” problem [7]. In addition to these problems, basic network operations such as neighbor discovery become more complicated, as well.

Neighborhood information plays an important role in multi-hop wireless networks for routing, clustering and MAC operation. Neighbor discovery is a relatively simpler problem when omnidirectional antennas are used since a simple broadcast can reach all nodes within the transmission range. The problem, however, becomes more challenging when directional antennas are used due to the following reasons: i) The limited radial range of the beamwidth of the directional antenna that covers only a fraction of the azimuth. This limitation requires the neighbor discovery scheme to be repeated in different directions to cover the whole azimuth. ii) Neighboring nodes must know when and where to point their directional beams to discover each other. iii) Due to non-ideal realizations of directional antennas (i.e., the existence of side lobes), two neighboring nodes might find multiple links between themselves through different Sector-to-Sector (S2S) links.

Many neighbor discovery protocols have been proposed for wireless networks that use directional antennas. The main objective of these protocols is to discover the neighbors around a node and store the neighborhood information locally. In the literature, three main approaches were used to perform the neighbor discovery. A set of proposed protocols utilize an omnidirectional antenna to bootstrap the neighbor discovery process [8], [9], [10]. Other protocols require time synchronization to perform neighbor discovery [8], [11], [12].

Finally, random schemes [13] perform the neighbor discovery by sending Hello packets through different random directions and then listening to another random direction. As discussed in Section II, all these approaches are associated with significant shortcomings.

In this paper, we introduce the Sectorized Antenna Neighbor Discovery (SAND) protocol, an integrated neighbor discovery mechanism that allows individual nodes to discover their neighbors and report their neighborhood information back to a centralized location for later use. The key idea of our proposed protocol is to serialize the neighbor discovery process among all nodes. This allows each node to perform neighbor discovery in a bounded time and reliably relay the neighborhood information. Unlike many other directional neighbor discovery protocols, SAND does not require an omnidirectional antenna or time synchronization. Moreover, SAND discovers all S2S links between nodes that they can communicate over using a systematic approach. SAND can be utilized with any distributed or centralized routing or MAC protocol that uses directional antennas since neighborhood information is available both centrally and locally.

## II. RELATED WORK

The performance improvements through directional antennas are well explored in the literature [1], [2], [3], [4] for ad hoc wireless networks, mesh networks [14] and sensor networks [15], [10]. To exploit these advantages many protocols have been proposed for MAC layer [3], [16], [17], [18], routing [19], [20] and topology control [21], [22]. Several neighbor discovery schemes have been proposed in the literature, as well. Some of the schemes utilize an omnidirectional antenna to bootstrap the neighbor discovery process like in [8], [9], [10]. Utilizing a secondary omnidirectional antenna for neighbor discovery has several disadvantages. First, differences in the omnidirectional and sector antenna gain patterns could result in different sets of discovered neighbors. Second, when discovering neighbors in SAND, we are not only interested in the identity of potential neighbors, but in measuring the link quality for the various Sector-to-Sector (S2S) links. This additional consideration cannot be realized using omnidirectional antennas. Finally, the cost of a secondary antenna system, along with the associated hardware required to support it, is undesirable in cost-sensitive networks. These schemes also assume that the discovering node knows its geographical location and can include this information in the advertisement packets which may be an expensive option for some wireless network implementations.

To eliminate the need for omnidirectional antennas, other proposed neighbor discovery schemes assume that nodes are time synchronized to guarantee that all nodes switch their sectors synchronously such as in [11], [13], [23]. While time synchronized neighbor discovery schemes eliminate the usage of omnidirectional antennas, they require time synchronization for all nodes which may increase the hardware and control overhead burden on the wireless network. Furthermore, achieving time synchronization in a distributed manner is also very

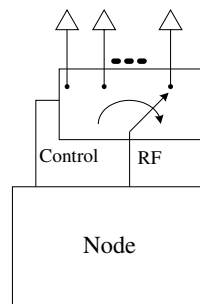


Fig. 1. Node Architecture with Sectorized Antennas

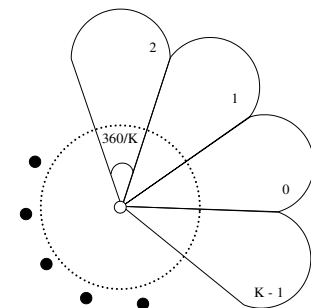


Fig. 2. Idealized  $K$  Sectorized-Antenna

unlikely without an initial communication infrastructure and solely based on directional/sectorized antennas.

Asynchronous and fully directional neighbor schemes were proposed in [2], [13] where each node listens in a random direction for a random time and then transmits a hello message in a random direction and goes back to listening in another random direction. Although this approach is simple and straightforward, it does not bind the time for neighbor discovery since it is a probabilistic approach. Moreover, it does not guarantee that discovered links are bidirectional.

With SAND protocol, directional neighbor discovery is done solely using directional antennas in both receiver and transmitter, allowing the discovery of more nodes than when using omnidirectional antennas. Moreover, SAND protocol does not require each node to know its geographical location or assume time synchronization between nodes. The serialized approach of SAND allow nodes to discover all potential Sector-to-Sector (S2S) links between a given pair of nodes which has been completely disregarded in the literature. In addition to discovering neighbor nodes and storing the neighborhood information locally, SAND gathers the neighborhood information from all nodes in a centralized location to be used, if needed, by centralized networking protocols. To the best of our knowledge, SAND is the first neighbor discovery protocol that has all these features.

## III. SECTORED ANTENNA NEIGHBOR DISCOVERY

SAND is designed for a  $K$  sectorized antenna that covers the entire omnidirectional  $360^\circ$  range, as shown in Figure 2. SAND can handle non-ideal, overlapping sectorized-antennas coverage. Switching between sectors is done by simply selecting a different antenna element, as shown in Figure 1, allowing this type of directional antenna to be applicable for a wider range of wireless network types. Each node can only activate one sector at a time for both transmission and reception. In this system, SAND does not require an additional omnidirectional antenna.

The key idea of SAND is the serialization of the neighbor discovery process among all nodes. While serialization results in a linear increase in discovery time with network size, it is important to note that neighbor discovery is performed at network initialization and for infrequent network topology updates. The concepts of SAND can be extended to more

frequent, incremental updates as well. However, this is beyond the scope of this paper.

Serialized neighbor discovery allows each node to perform neighbor discovery within a bounded time. Moreover, each node can relay its neighborhood information to a centralized location reliably. Serialization is achieved using a token passing approach where only the node that holds the token (Token-Holder node) is allowed to perform the neighbor discovery. Once the Token-Holder node finishes its neighbor discovery, it passes the token to one of its neighbor nodes that did not perform the neighbor discovery yet. This token passing mechanism starts a temporary parent-child relationship between the node that passes the token and the node that receives the token. This parent-child relationship creates a temporary routing tree that is also used to relay the neighborhood relationships to a centralized node if needed [24].

Once a node holds the token, the neighbor discovery process starts which can be summarized in three steps:

- 1) Holding the attention of its neighbors (Honing In mechanism)
- 2) Discovering neighbors (Hello-Reply Mechanism)
- 3) Passing the token to a neighbor or Releasing the token to the parent (Token Passing and Releasing Mechanisms)

#### A. Initialization

Initially, nodes start in *Fast Scan* mode where they switch their sectors sequentially looking for any channel activity in the active sector. During this mode, each node waits for  $t_{switch}$  before switching to another sector. Note that, such switching is not synchronized among nodes. The token is assigned, initially, to the node that will gather all the neighborhood information later on such as a cluster head or a sink node. The node that holds the token is called the Token-Holder node.

#### B. Hone In Mechanism

Before the Token-Holder can discover its neighbors, the Token-Holder must hold the 'attention' of its neighbors. Recall that, initially, all nodes are in Fast Scan mode switching their sectors looking for any channel activity. The Token-Holder begins the hone-in process by broadcasting a series of *HoneIn* beacons to all sectors in series. Each *HoneIn* message contains a sequence number  $x$  that indicates the remaining *HoneIn* beacons before the Token-Holder finishes its process. Upon the reception of a *HoneIn* message, neighboring nodes stop scanning and set a timer to expire when the Hone-In process is complete after  $x \cdot t_{HoneIn}$  where  $t_{HoneIn}$  is the predefined interval between the transmission of two successive *HoneIn* beacons. An example of the hone in process in one sector is shown in Figure 3. This process is repeated for all  $K$  sectors of the Token-Holder. Note that since nodes are not synchronized, the frequency of switching between sectors should be slower than the frequency of sending *HoneIn* messages allowing switching nodes to catch at least one *HoneIn* message (e.g., in our simulation and hardware implementation  $t_{switch}$  is set to be  $2 \cdot t_{HoneIn}$ ).

An example of the timing associated with the Hone-In process is shown in Figure 4. In this example, two neighbors receive the beacon message from the Token-Holder node. Sector 2 of Neighbor A is facing sector 2 of the Token-Holder while sector 1 of Neighbor B is facing sector 1 of the Token-Holder. When the sectors are aligned properly, the neighbor node ceases switching of sectors and starts a timer which will expire when the Token-Holder node has complete its Hone-In process. Note that, the *HoneIn* mechanism initiates local time synchronization for the remaining of the neighbor discovery of the Token-Holder node. The time synchronization is not needed once token is passed.

The time  $T_{HoneIn}$  required for the hone is

$$T_{HoneIn} = h * K * t_{HoneIn}, \quad (1)$$

where  $h$  is the number of HoneIn beacons that is transmitted by the Token-Holder in each sector and  $K$  is the number of sectors. To minimize  $T_{HoneIn}$ ,  $h$  should be selected as  $K+1$  to guarantee that a neighboring node receives at least one beacon. After the Hone-In process is complete, the Hello-Reply timer will expire on all neighboring nodes, indicating that the Hello-Reply process should start.

#### C. Hello-Reply Mechanism

After the hone in process completes, all neighbor nodes start the Hello-Reply process by switching to the first sector. The Token-Holder node then begins discovering its neighbors by polling the neighbors and waiting for their replies. This two-way handshake guarantees the discovery of bidirectional links to all neighbor nodes. During the Hello-Reply process, both the Token-Holder and neighbor nodes loops through their  $K$  sectors in a pre-determined time sequence, ensuring that the communication over all S2S links are tested for later comparison. The process of sequentially testing all sector combinations is illustrated in Figure 7.

For all  $K^2$  S2S links, the Token-Holder broadcasts  $N_{rounds}$  *Hello* packets, waiting for *Reply* packets from neighbors after each transmission. Neighbor nodes, after receiving the *Hello* packet, select a random slot for their response. A maximum of  $N_{slots}$  are available after each *Hello* message. The slotted reply mechanism is intended to reduce the probability of collision between neighbor nodes. If two nodes do select the same slot, their *Reply* packets will collide and thus will not be discovered by the Token-Holder in this Hello-Reply round. Each *Reply* packet contains the ID and sector of the originating node. Also, the *Reply* packet contains a bit flag to indicate whether or not the neighbor node has performed its own neighbor discovery yet. After receiving the *Reply* packet, the Token-Holder adds the neighbor's information into its locally collected neighbor table.

The Token-Holder repeats *Hello-Reply* packet exchange for  $N_{rounds}$  times for each sector-pair to discover new neighbors which their *Reply* packets have collided in previous exchanges. To maximize the probability of discovering all nodes within a bounded time, the Token-Holder node transmits the IDs of all neighboring nodes that have been discovered using a Long

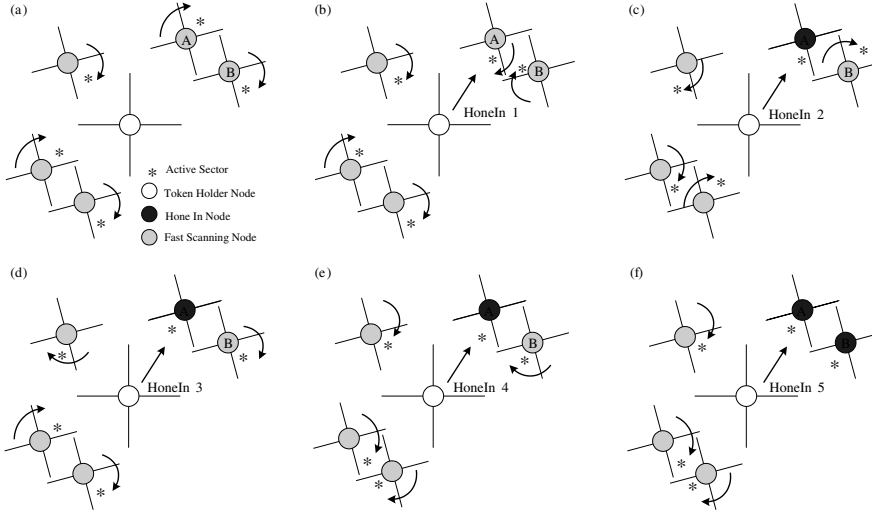


Fig. 3. Hone In process with 4 Sectored Antenna: (a) All Nodes except the Token-Holder node are in Fast Scan mode, (b) Token-Holder broadcasts the first HoneIn beacon to its first sector while node *A* activates the sector pointing towards the Token-Holder node, (c) Node *A* receives the HoneIn beacon and locks its sector while node *B* continues switching its sectors, (d) Token-Holder broadcasts its third HoneIn beacon, (e) Node *B* switches to the sector pointing toward the Token-Holder node and receives the 4<sup>th</sup> HoneIn beacon, (f) Node *B* locks its sector towards the Token-Holder node and stops switching its sectors.

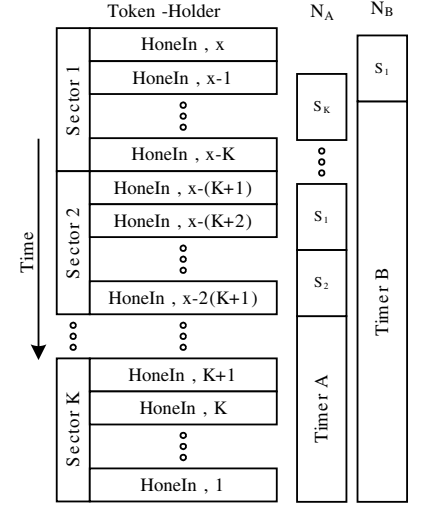


Fig. 4. Hone-In Process for Two Neighbor Nodes

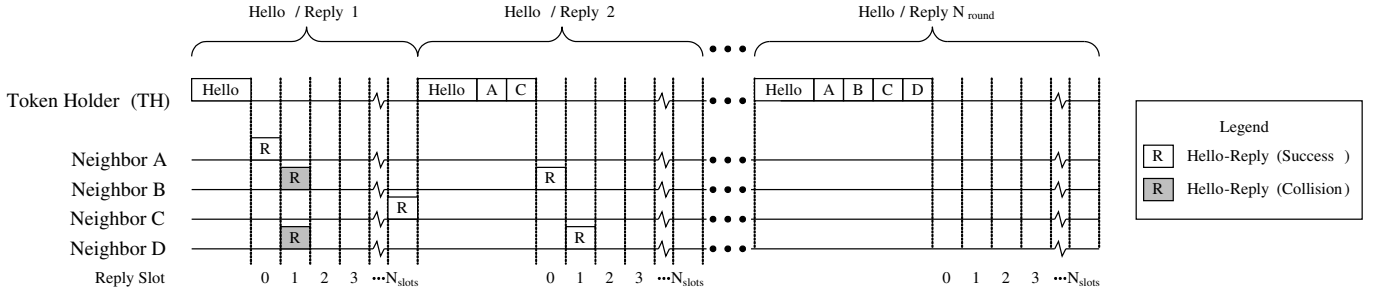


Fig. 5. Hello-Reply Messages Exchange

Hello packet in successive *Hello-Reply* rounds. Only nodes that do not read their IDs in this last packet are allowed to reply. This mechanism allows unlucky nodes that have not been discovered yet due to *Reply* packet collisions to send a successful *Reply* packet with a higher probability.

An example of the *Hello-Reply* packet exchange for one sector pair is given in Figure 5. After neighbor nodes receive the first *Hello* packet, they start a slotted time and select a random slot to transmit their *Reply* packets. Both node *B* and *D* select the slot number 1 to send their replies. As a result their reply packets collide, while nodes *A* and *C* select the 0<sup>th</sup> and last slots, respectively. After the first *Hello-Reply* round, the Token-Holder discovers both nodes *A* and *C* but not *B* and *D*. In the second *Hello-Reply* round, the Token-Holder includes the IDs of the already discovered nodes, nodes *A* and *C*, in the Hello packet. Allowing more chance for both nodes *B* and *D* to be discovered.

This *Hello-Reply* packet exchange is repeated in the remaining  $K - 1$  sectors. Obviously, the time  $T_{HelloReply}$  needed to finish this process depends on the number of slots  $N_{slots}$  and the number of *Hello-Reply* packets rounds  $N_{rounds}$  and can

be calculated as:

$$T_{HelloReply} = K * [(\tau_{Hello} + N_{slots}) \cdot (N_{rounds} - 1) + \tau_{HelloLong} + N_{slots}],$$

where  $\tau_{Hello}$  and  $\tau_{HelloLong}$  are the transmission time for the Hello and long Hello packets.

#### D. Token Passing Mechanism

After the Token-Holder finishes with its directional neighbor discovery mechanism described in Section III-C, it passes the token to one of its neighbors that has not performed the neighbor discovery process yet. Recall that, during *Hello-Reply* exchanges each replying node sets a bit in the reply packet to indicate whether the replying node has performed its neighbor discovery or not. The current Token-Holder searches its neighbor table for the first node that has not perform the neighbor discovery yet. Once such a neighbor is found, the Token-Holder sends a *GoToFastScan* packet with the ID of the new Token-Holder node in the destination field over all  $K$  sectors. Upon the reception of this packet, the new Token-

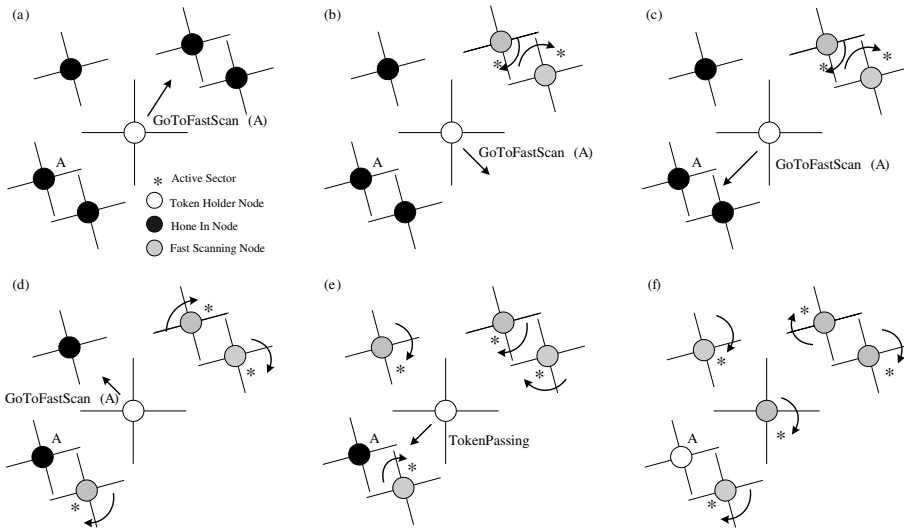


Fig. 6. Token Passing Mechanism: (a) Token-Holder sends GoToFastScan packet that includes the node ID  $A$  of the next Token-Holder to its first sector, (b) Neighbor nodes in the first sector starts their Fast Scan mode, the Token-Holder transmits the GoToFastScan packet to the second sector, (c) The Token-Holder sends the GoToFastScan packet to the third sector which contains the new Token-Holder node  $A$ , (d) Node  $A$  stays honed in while other nodes in that sector start their Fast Scan mode. (e) After sending the GoToFastScan packet to all sectors, the Token-Holder passes the token to node  $A$ , (f) After the acknowledgment of the reception of the token, node  $A$  becomes the new Token-Holder and the old Token-Holder starts its Fast Scan mode.

Holder waits for the token, other neighbor nodes go to Fast Scan mode and start switching their sectors.

After sending the *GoToFastScan* packet over all its  $K$  sectors, the Token-Holder sends the Token to the new Token-Holder and waits for an acknowledgment to ensure a reliable transmission of the token. Once the old Token-Holder receives the acknowledgement, it goes into Fast Scan mode and starts switching its sectors. The new Token-Holder tags the node that passes the token as a parent to release the token back as will be described in Section III-E. An example of Token passing is shown in Figure 6. If the Token-Holder node cannot find any node that needs the token, it releases the token to its parent as will be shown in the following section.

### E. Token Releasing Mechanism

When all neighbors of a Token-Holder complete their neighbor discovery, the Token-Holder hands the token back to its parent. However, before it releases the token back, it should hold the attention of its parent using a limited version of the hone in process described in Section III-B. Instead of performing the hone in process in all sectors, the Token-Holder node performs the hone in process only on the sector that contains its parent node. Once the hone in process ends, the Token-Holder inserts its neighborhood information into the token release message and sends it to the parent node who will become the new Token-Holder node. Once all nodes perform their neighbor discovery, eventually, the token will be released to the first node that started the neighbor discovery along with the whole neighborhood information.

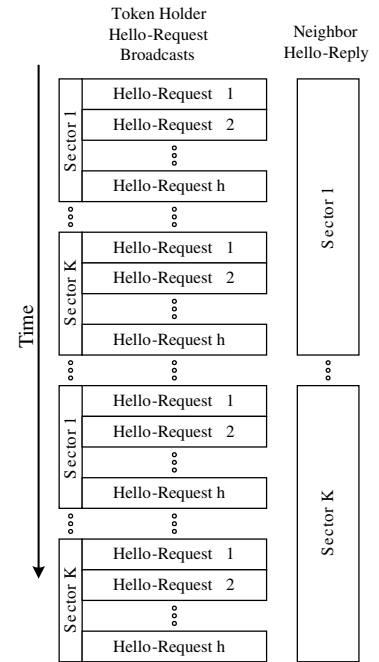


Fig. 7. Sector Switching for Hello-Reply Mechanism

## IV. ANALYSIS OF HELLO-REPLY MECHANISM

In this section, we present an analytical model to analyze the Hello-Reply mechanism of SAND. Furthermore, an optimization algorithm is proposed to select optimal parameter configuration of the Hello-Reply mechanism (i.e. the number of time slots  $N_{slots}$  and the number of rounds  $N_{rounds}$ ). For convenient notation, we use  $s$  to refer to the number of slots  $N_{slots}$  and  $r$  to refer to the number of rounds  $N_{rounds}$ .

### A. Analytical Model for Slot Selection Mechanism

To discover a node during an arbitrary slot time 'a' out of  $s$  slots, exactly one node out of  $n$  neighboring nodes should select that slot and transmit its reply packet<sup>1</sup>. Let us define slot 'a' as a **Discovery Slot**. Then, the probability  $p(n, s)$  that an arbitrary slot is a discovery slot (i.e. exactly one node transmits during that slot) can be given as

$$p(n, s) = \binom{n}{1} \left(\frac{1}{s}\right) \left(1 - \frac{1}{s}\right)^{n-1}, \quad (2)$$

where  $\frac{1}{s}$  is the probability that a node selects a random slot out of  $s$  slots. Let  $m(n, s)$  be the maximum number of nodes that can be discovered during a single Hello-Reply round given that there are  $n$  nodes competing for  $s$  slots. Formally,  $m(n, s)$  can be defined as

$$m(n, s) = \begin{cases} s - 1 & \text{if } n > s \\ n & \text{if } n \leq s. \end{cases} \quad (3)$$

<sup>1</sup>An approximate value of  $n$  can be calculated using node density

Then, the probability  $q_1(n, s)$  of discovering exactly 1 node out of  $n$  nodes during a Hello-Reply round that consists of  $s$  slots is given as

$$q_1(n, s) = \binom{s}{1} p(n, s) l(n-1, s-1), \quad (4)$$

where  $l(n-1, s-1)$  is the probability that none of the remaining  $(s-1)$  slots is a discovery slot. Alternatively,  $l(n-1, s-1)$  can be defined also as the probability that none of the remaining  $(n-1)$  nodes is discovered during the remaining  $(s-1)$  slots. The term  $p(n, s)$  is the probability of having an arbitrary discovery slot and  $\binom{s}{1}$  is the number of ways this arbitrary slot can be selected. The probability  $q_i(n, s)$  of discovering  $i$  nodes ( $1 \leq i \leq m(n, s)$ ) in one Hello-Reply round can be calculated as

$$q_i(n, s) = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{if } n \leq s \ \& \ i = n - 1 \\ \binom{s}{i} \left[ \prod_{j=0}^{i-1} p(n-j, s-j) \right] & \\ \times l(n-i, s-i) & \text{Otherwise.} \end{cases} \quad (5)$$

which can be used to calculate the probability mass function of the number of neighbor nodes that can be discovered in one Hello-Reply round.

Finally, the probability  $l(n-i, s-i)$  is defined as the probability that none of the remaining  $(s-i)$  slots are discovery slots, or alternatively, it is the probability that no discovery slot exist among  $(s-i)$  slots and is calculated as

$$\begin{aligned} l(n-i, s-i) &= q_0(n-i, s-i) \\ &= 1 - \sum_{k=1}^{m(n-i, s-i)} q_k(n-i, s-i). \end{aligned} \quad (6)$$

### B. Analytical Model for Hello-Reply Rounds

Recall that from Equation 3, the discovering node can only discover up to  $m(n, s)$  nodes in one Hello-Reply round. To estimate the probability that  $k$  nodes are discovered when the neighbor discovery process ends (i.e., after  $r$  Hello-Reply rounds), we must keep track of different possibilities that lead to discovering the  $k$  nodes in  $r$  rounds.

To account for such possibilities, multiple Hello-Reply rounds can be described as a two dimensional probability map with  $(n+1) \times r$  states as shown in Figure 8. Each state  $(x, y)$  represents the probability that  $y$  nodes are discovered up to and including the  $x^{th}$  round. Any state  $(x, y)$  can only access states  $(x+1, y), (x+1, y+1), \dots$  or  $(x+1, \min(y+m(y, s), n))$ . The transition probabilities from states  $(x, a)$  to  $(x+1, b)$  are given by the probability mass of the number of discovered nodes in round  $x$ .

Note that the states' probabilities are greatly dependent on the number of nodes  $n$  and the number of slots  $s$ . In fact, the probability  $b_s^n(x, y)$  of the state  $(x, y)$  given that there are  $n$

---

**FindOptimalConfiguration**( $n, \epsilon, maxSlots, t_r, t_s$ ): For a given  $n$ , find the minimum discovery time  $r * (t_r + t_s * s)$  such that  $P_{Miss} \leq \epsilon$

---

**Input:**  $n, \epsilon, maxSlots, t_r, t_s$

**Output:**  $s_{optimal}, r_{optimal}$

**begin procedure**

1.  $DiscoveryTime = \infty$
  2. **for**  $s = 2$  **to**  $maxSlots$  **do** /\* Slots For Loop\*/
  3.  $notFound = TRUE$
  4.  $r = 1$  /\* Start with one round\*/
  5. **while**  $notFound$  **do** /\* Rounds While Loop\*/
  6.  $P_{Miss} = 1 - b_s^n(r, n)$  /\* using Equation 7 \*/
  7.  $newDiscoveryTime = r * (t_r + t_s * s)$
  8. **if**  $(P_{Miss} \leq \epsilon)$  **and**  $(newDiscoveryTime \leq DiscoveryTime)$
  9.  $DiscoveryTime = newDiscoveryTime$
  10.  $s_{optimal} = s$
  11.  $r_{optimal} = r$
  12.  $notFound = FALSE$
  13. **else**
  14.  $r = r + 1$  /\* Increase the number of rounds\*/
  15. **end if**
  16. **end while** /\* Rounds While Loop\*/
  17. **end for** /\* Slots For Loop\*/
- end procedure**
- 

nodes and  $s$  slots per round can be given as

$$b_s^n(x, y) = \begin{cases} q_y(n, s) & \text{if } x = 1 \\ \sum_{k=0}^n [q_{(k)}(n - (y - k), s) \times b_s^n(x-1, y-k)] & \text{if } 1 < x \leq r. \end{cases} \quad (7)$$

Therefore, the pdf of the number of neighbors discovered after  $r$  rounds is given by calculating the probabilities  $b_s^n(r, 0), b_s^n(r, 1), \dots, b_s^n(r, n)$ .

### C. Model Validation

To validate the accuracy of our analytical models, we have implemented the slot selection mechanism in MATLAB and compared its result with the result obtained from the models presented earlier. Simulation results are averaged over 10000 runs with different seeds. Figure 9 shows a comparison of the cumulative distribution function of the number of discovered nodes out of 10 nodes using different number of slots. As shown in the Figure, the model matches perfectly with the simulation results. Figure 10 shows the cumulative distribution function of the number of discovered nodes using 4 slots in different number of rounds with similar accuracy.

### D. Parameter Optimization

The aforementioned Hello-Reply model can be used to optimize the parameters of the neighbor discovery mechanism. We are interested in finding the optimal system configuration

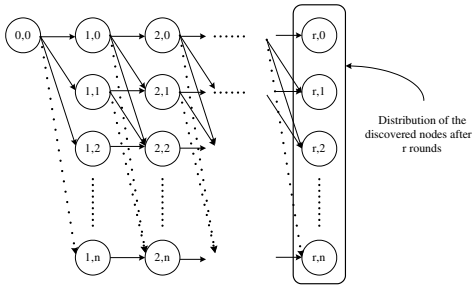


Fig. 8. Two Dimensional Map for  $r$  rounds and  $n$  nodes

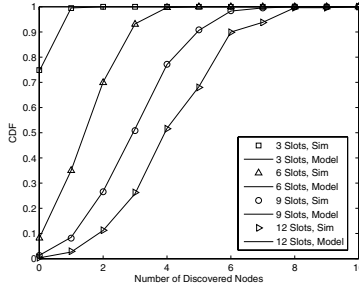


Fig. 9. CDF of the number of discovered nodes with 10 nodes and different number of slots in one round

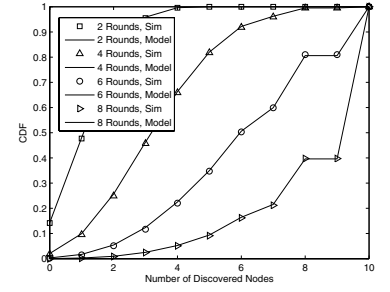


Fig. 10. CDF of the number of discovered nodes with 10 nodes, 4 slots and different number of rounds

that minimizes the discovery time such that probability that a node misses any of its  $n$  neighbors is less than a certain threshold  $\epsilon$ . Each round  $r$  takes  $t_r + t_s * s$  seconds to finish where  $t_r$  is the time to send the Hello packet and  $t_s$  is the slot duration. Then, the discovery time can be calculated as

$$t_d = r * (t_r + t_s * s). \quad (8)$$

We can represent this optimization problem as the following linear program

$$\begin{aligned} & \text{Minimize} && r * (t_r + t_s * s) \\ & \text{Subject To} && P_{Miss} \leq \epsilon, \end{aligned} \quad (9)$$

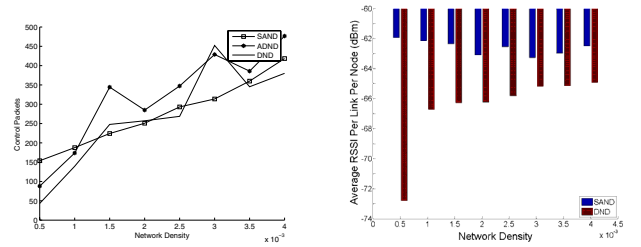
where  $P_{Miss}$  is the probability of missing at least one node and can be computed using Equation 7 as follows

$$\begin{aligned} P_{Miss} &= 1 - P[\text{n nodes are discovered in r rounds}] \\ &= 1 - b_s^n(r, n). \end{aligned} \quad (10)$$

To solve this optimization problem, we propose a heuristic approach to find the optimal number of slots and rounds. The approach sequentially iterates through all slot values starting from 2 slots to  $maxSlots$  slots. Then, for each number of slots, the approach finds the minimum number of rounds that satisfies the condition  $P_{Miss} \leq \epsilon$  and minimizes the discovery time. The algorithm **FindOptimalConfiguration**( $n, \epsilon, maxSlots, t_r, t_s$ ) describes our approach.

## V. SIMULATION EVALUATION

To verify the concepts of the proposed SAND protocol, we have implemented SAND in Qualnet simulator [25] using the directional antenna module emulating a 4 sectored antennas pattern shown in Figure 12(d). For a comparative study, we have implemented the asynchronous direct discovery mechanism proposed in [13], abbreviated throughout this section as Directional Neighbor Discovery DND, which does not depend either on omnidirectional antennas nor on time synchronization. With SAND protocol, the neighbor discovery stops once the neighbor discovery token is received by the collecting node. The Hello-Reply parameters (i.e.  $N_{rounds}$  and  $N_{slots}$ ) are optimized using the algorithm *FindOptimalConfiguration* where the number of nodes  $n$



(a) Per Node Control Packets (b) Per Node Average Link RSSI

Fig. 11. Performance Comparison

are increased from 5 to 40 nodes and  $\epsilon = 0.95$ . Nodes are distributed randomly on a  $100m \times 100m$  field with transmitting power of  $10dBm$  making a multihop network of two hops.

Since DND is a distributed protocol, we have used two conditions to stop the operation of DND. The first condition is to stop DND as soon as all nodes discover all their neighbors over any S2S links. With the second condition, DND stops as soon as the best Sector-to-Sector Links between all neighbors are discovered. Throughout the simulation studies, we call the second implementation of DND as Advanced DND (ADND). DND performance greatly depends on the Hello packets' sending rate which is optimized according to Equation (11) in [13].

Figures 11 show the performance comparison between the three discovery protocols. Figure 11(a) shows the number of control packets needed to perform the neighbor discovery. Note that for SAND, this number includes the control packets for the exchanging the tokens and neighborhood information. As shown in the Figure, ADND requires higher number of control packets compared to DND and SAND. DND requires comparable amount of overhead as SAND but its requirements highly varies. Figure 11(b) shows the average Received Signal Strength RSSI per link per node as the network density increases. The results for ADND has been omitted since it discovers, like SAND, the best S2S links between neighbor nodes. On the other hand, DND protocol stops once it discovers all neighbors without considering the RSSI of the links. The increase in per-link RSSI translates to either more reliable

point-to-point communication, or potentially, if the underlying MAC protocol allows variable data rate communication, higher bit rate transmissions adaptive modulation and coding.

In our simulations, the discovery time for single node, with a network density of  $3 * 10^{-3}$  nodes per square meter, took approximately 1 second. As SAND is a serialized protocol, neighbor discovery time for the entire network increases linearly with the number of nodes in the network. SAND can be used in small-scale static mesh or ad-hoc networks. For large-scale WSNs, SAND can be run inside multiple clusters in parallel if the WSN is pre-partitioned into clusters. However, the advantages of SAND over DND and ADND are clear: with comparable control overhead, SAND discovers optimal S2S link pairs for neighboring nodes and collects this information at a centralized location. Moreover, DND and ADND's practical implementations still suffer from uncertainty of discovery durations since there is no centralized oversight.

## VI. TESTBED IMPLEMENTATION

To validate SAND, we have implemented our protocol on a custom hardware platform developed by ETRI shown in Figure 12(c). All nodes use Texas Instruments (TI) MSP430 series processor for computation, TI CC2420 transceiver for communications, and a 6-sectored antenna with overlapping  $120^\circ$  azimuth. Figures 12(a) and 12(b) show the top and bottom view of the sectored antenna, respectively. In the experiments, we only use 3 sectors to cover the entire azimuth. SAND was programmed using a proprietary operating system Nano-Q-Plus developed by ETRI. The testbed implementation of SAND was configured with the following parameters: transmission power is 0 dBm (1mW),  $t_{switch}$  is 10ms,  $t_{HoneIn}$  is 5ms,  $N_{slots}$  is 5, and  $N_{rounds}$  is 5.

SAND was tested in an outdoor environment using a topology where five nodes (Nodes 1-5) are placed at random distances between 50m and 85m from a sink node (Node 0). Nodes 1-5 are oriented such that their sectors with Sector ID 1 point towards the sink node. Neighbor information was collected at each node and reported to the sink node. Figure 13 shows the neighbor relations discovered at Node 0 for its neighbor Node 4 which were 52m apart. The first two columns represent the sectors of Node 0 and 4, respectively. RSSI refers to the signal strength observed. Figure 14 shows the S2S links that were discovered for all neighbor relations not including the sink node (node 0). Due to the spatial separation of the nodes, not all nodes are neighbors, as shown by the relatively brief list of neighbor relations.

Due to the proximity of Nodes 0 and 4, we can immediately see the need for careful selection of sectors for communication. Though there are multiple sector pairs that can support some level of communication, by testing and selecting the best S2S link between the neighboring nodes, as is done with SAND, we can see in this example an increase of 18 dB in RSSI. These additional neighbor relations, which are discovered through SAND, can be useful to TDMA scheduling schemes such as [24] to avoid unexpected interference due to improper sector reuse. For example, without this information,

Node 0 Sector	Node 4 Sector	RSSI
2	1	-69
3	1	-74
2	2	-77
2	3	-77
1	1	-78
3	2	-81
3	3	-81
1	2	-84
1	3	-87

Fig. 13. All S2S Links between Nodes 0 and 4

(Node,Sec)	(Node,Sec)	RSSI
(1,3)	(4,1)	-82
(2,1)	(4,3)	-84
(2,1)	(3,1)	-84
(2,1)	(1,1)	-85
(4,2)	(5,3)	-83

Fig. 14. Best S2S Links between Remaining Node Pairs

the scheduler could assign communication on Sector 2 of Node 4 and Sector 1 of Node 0 simultaneously, oblivious to the fact that, due to the presence of side-lobes in the sector antennas, these sectors can interfere.

## VII. PRACTICAL CONSIDERATIONS

While implementing SAND on hardware, one must consider many practical issues that go beyond the theoretical development of the protocol. Some of the more significant considerations include i) imperfect synchronization, ii) hardware and operating system related delays, memory and bandwidth limitations, iii) realistic sector antenna patterns that include more than just the main antenna lobe, and finally iv) token recovery due to lost packets during neighbor discovery.

Imperfect synchronization is not only a problem between nodes, but within the operating system of the node itself. For example, the hardware this research was implemented on utilizes a 32.768 KHz crystal to synchronize its internal timers. The internal interrupts therefore have a granularity of approximately  $32.768 \mu s$ . When utilizing hardware timers, the protocol can be fairly accurate in its timing, assuming that the processor is not busy when the hardware timer expires, though the number of available hardware interrupt timers is limited.

Memory and bandwidth limitations can have a significant impact on the performance of a neighbor discovery protocol, especially when the protocol, like SAND, is designed to collect link quality information on all discovered sector pairs. In this case, there can be a maximum of  $K^2$  neighbor table entries for each neighbor. Assuming there is sufficient memory in each node for such neighbor tables, as the information is transferred to a central location for processing, the burden, in terms of memory and bandwidth requirements, will increase dramatically in multi-hop topologies. If only best S2S links are needed, it is possible to filter the neighbor tables locally, eliminating entries that correspond to weak links and thus reducing the bandwidth and memory burden for all nodes in the network significantly.

One must keep in mind when developing any neighbor discovery protocol with sector antennas that realistic sector antenna patterns have side lobes. For a neighbor discovery protocol, this translates to non-optimal sector combinations being discovered for neighboring nodes. SAND thoroughly tests all sector combinations for each neighbor to ensure that



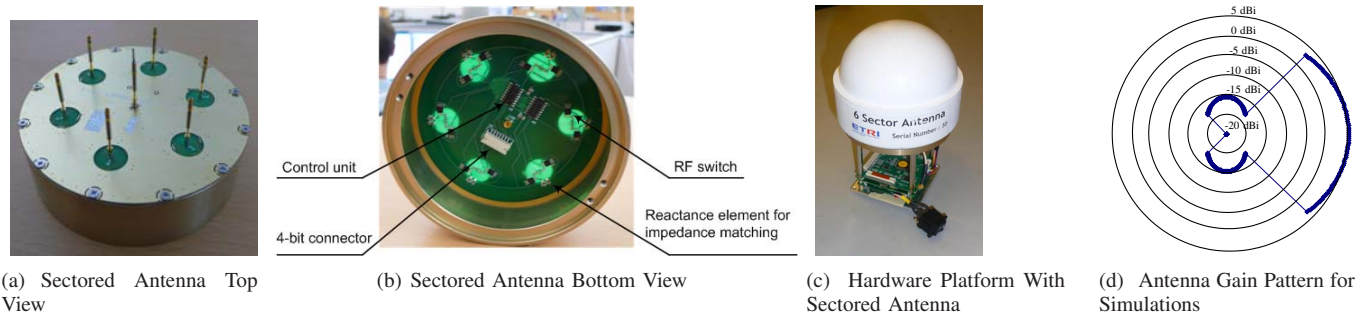


Fig. 12. Testbed / Simulation Antenna Implementations

the optimal sector is discovered. While this may lead to increased burden on the network due to enlarged neighbor table, as described above, in the end this leads to a more reliable network as we ensure that all neighbors communicate through the sectors with the most favorable link conditions.

Finally, in token based protocols, such as SAND, one must always account for the possibility of a lost token. Unreliable wireless links or failed wireless nodes can easily cause breaks in the temporary depth-first routing tree. Such losses can be dealt with through timeouts. When passing the neighbor discovery token to a neighbor node, a timer can be set that expires after the expected token return time. After the timer expires, the token holder can resume the token-passing scheme, ignoring the bad link.

### VIII. CONCLUSIONS

In this paper, we presented a fully directional neighbor discovery protocol called Sectored-Antenna Neighbor Discovery (SAND) protocol. Unlike many proposed directional neighbor discovery protocols, SAND depends neither on omnidirectional antennas to bootstrap the discovery process nor on time synchronization. In addition, SAND performs neighbor discovery in a serialized fashion allowing individual nodes to discover all potential neighbors within a predetermined time. Moreover, SAND systematically discovers all S2S links between neighboring nodes and gathers the neighborhood information in a centralized location, if needed, to be used by centralized networking protocols. Effectiveness of SAND has been assessed via simulation studies and a real hardware implementation.

### IX. ACKNOWLEDGMENT

This research was supported by the Dual Use Technology Program, South Korea (06-II-LC-01, Surveillance and Reconnaissance Sensor Network Development).

### REFERENCES

- [1] Y. Ko, V. Shankarkumar, and N.H.Vaidya, "Medium access control protocols using directional antennas in adhoc networks," *IEEE INFOCOM 2000*, pp. 13–21, 2000.
- [2] R. Ramanathan, "On the performance of ad hoc networks with beam-forming antennas," in *MobiHoc '01*.
- [3] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, "Using directional antennas for medium access control in ad hoc networks," in *MobiCom '02*.
- [4] S. Yi, Y. Pei, and S. Kalyanaraman, "On the capacity improvement of ad hoc wireless networks using directional antennas," in *MobiHoc '03*.

- [5] Y. Wang and J. Garcia-Luna-Aceves, "Spatial reuse and collision avoidance in ad hoc networks with directional antennas," in *GLOBECOM '02*.
- [6] M. Sekido, M. Takata, M. Bandai, and T. Watanabe, "A directional hidden terminal problem in ad hoc network mac protocols with smart antennas and its solutions," in *GLOBECOM '05*.
- [7] H. Gossain, C. Cordeiro, D. Cavalcanti, and D. Agrawal, "The deafness problems and solutions in wireless ad hoc networks using directional antennas," in *GLOBECOM '04*.
- [8] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, "Ad hoc networking with directional antennas: A complete system solution," in *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 496–506, March 2005.
- [9] R. Santosa, B. Lee, C. Yeo, and T. Lim, "Distributed Neighbor Discovery in Ad Hoc Networks Using Directional Antennas," in *IEEE CIT '06*.
- [10] S. Zhang and A. Datta, "A directional antenna based MAC protocol for wireless sensor networks," *ICCSA 2005*, pp. 686–695, 2005.
- [11] G. Jakllari, W. Lup, and S. Kirshnamurthy, "An Integrated Neighbor Discovery and MAC protocol for Ad Hoc Networks Using Directional Antennas," in *IEEE Transaction on Wireless Communication*, vol. 6, pp. 1114–1124, March 2007.
- [12] Z. Zhang and B. Li, "Neighbor discovery in mobile ad hoc self-configuring networks with directional antennas: algorithms and comparisons," *IEEE Trans on Wireless Comm.*
- [13] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *IEEE INFOCOM 2005*, vol. 4, pp. 2502–2512, March 2005.
- [14] U. Kumar, H. Gupta, and S. Das, "A topology control approach to using directional antennas in wireless mesh networks," in *ICC 2006*.
- [15] T. Dimitriou and A. Kalis, *Efficient Delivery of Information in Sensor Networks Using Smart Antennas*, pp. 109–122. 2004.
- [16] Y.-B. Ko, V. Shankarkumar, and N. Vaidya, "Medium access control protocols using directional antennas in ad hoc networks," in *INFOCOM '00*.
- [17] T. Korakis, G. Jakllari, and L. Tassioulas, "A MAC protocol for full exploitation of directional antennas in ad-hoc wireless networks," (*ACM MobiHoc '03*), pp. 98 – 107, June 2003.
- [18] A. Nasipuri, S. Ye, J. You, and R. Hiromoto, "A mac protocol for mobile ad hoc networks using directional antennas," in *IEEE WCNC 2000*.
- [19] A. Saha and D. Johnson, "Routing improvement using directional antennas in mobile ad hoc networks," in *GLOBECOM '04*.
- [20] S. Roy, D. Saha, S. Bandyopadhyay, T. Ueda, and S. Tanaka, "A network-aware mac and routing protocol for effective load balancing in ad hoc wireless networks with directional antenna," in *MobiHoc '03*.
- [21] V. Nambodiri, L. Gao, and R. Janaswamy, "Power efficient topology control for wireless networks with switched beam directional antennas," in *IEEE MASS '05*.
- [22] Z. Huang, C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo, "Topology control for ad hoc networks with directional antennas," in *IEEE ICCCN '02*.
- [23] G. Pei, A. Kim, J. Nast, D. Norris, and P. Norris, "A neighbor discovery protocol for directional antenna networks," in *IEEE Military Communications Conference MILCOM*, vol. 1, pp. 487–492, Oct 2005.
- [24] E. Felemban, S. Vural, R. Murawski, E. Ekici, S. Park, K. Lee, and Y. Moon, "SAMAC: A Cross-Layer Communication Protocol for Sensor Networks with Sectored Antennas," *To Appear in Trans. on Mobile Computing*.
- [25] "<http://www.scalable-networks.com/>,"