Energy-Efficient Clustering/Routing for Cooperative MIMO Operation in Sensor Networks

Mohammad Z. Siam and Marwan Krunz
Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ 85721
{siam, krunz}@ece.arizona.edu

Ossama Younis
Applied Research, Telcordia Technologies, Inc.
Piscataway, NJ 08854
oyounis@research.telcordia.com

***Abstract*— Employing multi-input multi-output (MIMO) links can improve energy efficiency in wireless sensor networks (WSNs). Although a sensor node is likely to be equipped with only one antenna, it is possible to group several sensors to form a virtual MIMO link. Such grouping can be formed by means of clustering. In this paper, we propose a distributed MIMO-adaptive energy-efficient clustering/routing scheme, coined cooperative MIMO (CMIMO), which aims at reducing energy consumption in multi-hop WSNs. In CMIMO, each cluster has two cluster heads (CHs), which are responsible for routing traffic between clusters (i.e., inter-cluster communications). CMIMO has the ability to adapt the transmission mode and transmission power on a per-packet basis. The transmission mode can be one of four transmit/receive configurations: 1×1 (SISO), 2×1 (MISO), 1×2 (SIMO), and 2×2 (MIMO). We study the performance of CMIMO via simulations. Results indicate that our proposed scheme achieves a significant reduction in energy consumption, compared to non-adaptive clustered WSNs.**

I. INTRODUCTION

Nodes in wireless sensor networks (WSNs) are typically powered by small batteries. Replacement or recharging of these sensors is typically difficult due to two reasons: (1) sensors are deployed in large numbers, making the process of collecting them back for recharging expensive and time consuming, and (2) in some environments, such as disaster areas, it can be infeasible to reach the sensors once they are deployed. Consequently, improving the energy efficiency in WSNs has always been a primary objective in recent research.

Multi-input multi-output (MIMO) technology has the potential to increase channel capacity and reduce transmission energy consumption in fading channels. This is done by exploiting three types of gains: array, multiplexing, and diversity [1]. Array gain is achieved either at the transmitter through directional alignment of the transmitted signal or at the receiver through coherent combining of multiple copies of the signal that are received over independent fading paths. Multiplexing gain is obtained when different signals are transmitted over several antennas for the purpose of increasing the total transmission capacity of a link. Diversity gain is interpreted as the slope of the average bit error rate (BER) curve versus signal-to-noise ratio (SNR). In this work, we focus on the diversity gain, leaving the exploitation of the other types for future research.

Typically, a MIMO system requires a node to possess more than one antenna. However, it is possible to group two or more sensor nodes to form a cooperative (virtual) MIMO node. To form such a virtual node, which we refer to as a *micro-cluster*, sensors must exchange information to decide on the data that need to be transmitted. To ensure that the energy overhead involved in information exchange is manageable, only those nodes that are geographically close to each other should be grouped to form a cooperative MIMO node. Most proposed cooperative MIMO systems have not considered multi-hop communications, although multi-hop communications are essential for large-scale networks.

Cooperative MIMO techniques are designed to eliminate data redundancy by aggregating data as early as possible, prior to forwarding it. Data aggregation is done by exploiting node clustering, which organizes the network into a connected hierarchy. In the context of WSNs, clustering involves grouping nodes and electing a cluster head (CH) such that the non-CH nodes of a cluster can directly communicate with their CH. CHs forward aggregated data to the sink directly or via other CHs. Thus, the collection of CHs in the network forms a connected dominating set.

In this paper, we propose a distributed MIMO-adaptive energy-efficient clustering/routing scheme, coined cooperative MIMO (CMIMO), for multi-hop WSNs. In this scheme, each cluster has two CHs, which are responsible for inter-cluster communications. Clustering is done based on the remaining energy of nodes, neighbor proximity, and the size of neighbor lists. The rationale for selecting these criteria is to build cooperative MIMO links whose effect is as close as possible to actual MIMO systems (with two antennas per node) and that have manageable overhead. Diversity gain of such a cooperative MIMO node is maximized by adapting the "transmission mode" and the transmission power of inter-cluster communications on a per-packet basis. By "transmission mode" we mean one of four transmit/receive configurations: 1×1 (single-input single-output / SISO), 2×1 (multi-input single-output / MISO), 1×2 (single-input multi-output / SIMO), and 2×2 (multi-input multi-output / MIMO)[1]. The total energy consumption that our scheme minimizes consists of transmission and circuit energies. For a given target BER, a multi-antenna transmission requires less transmission power than a SISO system. However, it also requires more circuit power at both ends of the link. As a result, a distance-dependent tradeoff emerges between transmission and circuit powers [3]: For relatively small distances, circuit power is dominant, and hence a SISO mode is more energy-efficient than a multi-antenna mode. As the transmitter-receiver distance increases, the tradeoff shifts in favor of multi-antenna modes (SIMO, MISO, MIMO).

The rest of the paper is organized as follows. We describe the CMIMO scheme in section II. The system model along with the overall energy consumption of CMIMO are analyzed in section III. In section IV we discuss some issues related to the design of CMIMO, including connectivity, listening cost, synchronization, reclustering, and medium access control (MAC). The performance of the proposed scheme is evaluated via simulations in section V. Section VI briefly describes related work. Section VII discusses the main conclusions of this paper as well as some generalizations and extensions of the proposed CMIMO scheme.

[1]The feasibility of adapting the transmission mode on a per-packet basis was demonstrated in several experimental MIMO platforms [2].

II. The CMIMO Scheme

In this section, we first give an overview of the proposed CMIMO scheme, followed by more operational details.

A. Overview

CMIMO is a distributed MIMO-adaptive energy-efficient scheme that aims at minimizing the total energy consumption (transmission plus circuit energies) in multi-hop WSNs. It employs a dynamic clustering approach with up to two CHs per cluster: a *master* CH (MCH) and a *slave* CH (SCH). The two CHs operate as a cooperative MIMO node for inter-cluster communications (see Figure 1). The operation of CMIMO has three main phases: cluster formation, intra-cluster communications, and inter-cluster communications with cooperative MIMO capabilities. We discuss these phases in detail in the following subsection. Cluster formation is done in a distributed way, and results in at most two CHs per cluster (the MCH is mandatory whereas the SCH may or may not be present). During intra-cluster communications, the selected MCH is responsible for aggregating data sent by other nodes in the cluster and exchanging this data with the SCH, so that the two CHs operate as a cooperative MIMO node (if needed). When data is available at the CHs of the source micro-cluster, inter-cluster communications are carried out by forwarding/exchanging data with other micro-clusters or directly with the sink. An energy-efficient routing algorithm is executed over the topology of virtual MIMO nodes to determine the end-to-end path that minimizes the total energy consumption among all possible paths between the CHs and the sink. The MCH (or both MCH and SCH) of the receiving cluster selects the optimal transmission mode and transmission power for communication with other CHs. The mode and power can be different for different hops, depending on the distances between CHs in different clusters.

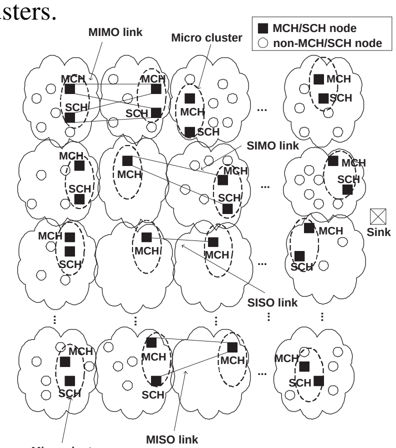

Fig. 1. Example topology of a clustered WSN with cooperative MIMO.

Our design is applicable to micro-clusters with more than two CHs. To simplify the exposition, we limit the number of CHs per cluster in this work to two. Note, however, that a higher number of CHs may result in more energy overhead, especially for intra-cluster communications. Therefore, we cannot assert that having more CHs in a micro-cluster improves the overall performance, compared to that of a 2×2 MIMO system.

B. Operational Details

In this section, we describe the operational details of the CMIMO scheme, including cluster formation, intra-cluster communications, and inter-cluster communications with cooperative MIMO capabilities.

TABLE I

First Iteration of Hello-Message Exchanges

Node ID	Heard hello messages
N_1	–
N_5	–
N_2	N_1
N_4	N_1, N_2, N_5
N_3	N_2, N_4

1) Cluster formation: The cluster formation process consists of the following steps:

Step 1: Neighborhood discovery: In this step, each node uses the classic carrier sense multiple access with collision avoidance (CSMA/CA) scheme to contend for the channel. Once a node v succeeds in accessing the channel, it sends a "hello" message at a fixed power (P_{intra}) to discover its 1-hop neighbors. This hello message carries the following: node ID, its remaining energy, and a list of v's neighbors (nodes that v has received hello messages from). What triggers node v to send a hello message is: (1) if v receives a hello message from a node, say u, that is not included in v's neighbor list, v adds u to its neighbor list and broadcasts the updated list to its neighbors. (2) If v receives a hello message from an already known neighbor u and u's neighbor list does not include v, v broadcasts a hello message to inform u about itself.

We now explain the neighborhood discovery process through an example. Consider the network in Figure 2. The first number at each node indicates the node ID, whereas the second number indicates the remaining energy. Dashed lines between any two nodes in this figure show that these nodes are neighbors. Assume that according to the CSMA/CA scheme, the sequence of transmissions is as follows: nodes 1 and 5 send first (they are not neighbors, i.e., they cannot hear each other), followed by node 2, then node 4, and finally node 3. Node 1 sends its hello message, informing others about its ID (N_1), its remaining energy (E_1), and that it has not heard from any neighbors yet. Node 5 sends its hello message, including its ID (N_5), its remaining energy (E_5), and its neighbors list, which is still empty. Node 2 sends its hello message, which contains its ID (N_2), its remaining energy (E_2), and its neighbors list, which has one neighbor (N_1). Node 4 sends its hello message, informing the other nodes about its ID (N_4), its remaining energy (E_4), and its neighbors list, which has three neighbors (N_1, N_2, and N_5). Finally, node 3 sends a hello message that includes its ID (N_3), its remaining energy (E_3), and its neighbors list. Suppose that the hello messages from N_1 and N_5 collided at node 3, so node 3 is aware of only two neighbors (N_2 and N_4). A summary of the hello-message exchanges in the first iteration is given in Table I.

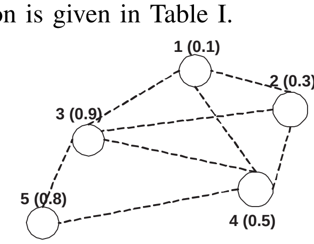

Fig. 2. Example that illustrates neighborhood discovery.

After each node sends its hello message in the first iteration, some nodes may still need to send additional hello messages if some inconsistency is detected in the neighborhood information. In our example, node 1 receives hello messages from new neighbors (N_2 and N_4) that have not been previously included in the neighbors list of node 1. Furthermore, node 1 has heard

TABLE II
SECOND ITERATION OF HELLO-MESSAGE EXCHANGES

Node ID	Heard hello messages
N_1	N_2, N_3^*, N_4
N_5	N_3^*, N_4
N_2	N_1, N_3, N_4
N_4	N_1, N_2, N_3, N_5
N_3	N_1, N_2, N_4, N_5

a hello message from node 3 that did not include N_1 in its neighbors list. Therefore, node 1 sends a second hello message, notifying N_2 and N_4 that it has received their hello messages, and informing N_3 (by adding a flag, which is represented in Table II by "$*$") that it will be expecting an updated hello message from N_3 that includes N_1. Node 5 has also to send a second hello message to notify N_4 that it has received its hello message, and inform N_3 that it is still waiting for N_3's updated neighbors list to include N_5. Nodes 2, 3, and 4 need also to send additional hello messages, as they have received hello messages from new neighbors that have not been previously included in their neighbor lists. A summary of the hello-message exchanges in the second iteration is given in Table II.

Note that N_1 and N_5 wait for a fixed duration of time (Δ) after sending their 2nd hello messages. After this Δ, if N_3 has not sent an updated list that includes N_1 and N_5, nodes N_1 and N_5 will send additional hello messages (only once), following the same procedure explained above. If N_1 and N_5 do not receive a response from N_3 after sending their third hello messages, they assume that N_3 has already heard their hello messages, but it may have not responded due to collision, etc.

Our neighborhood discovery approach has the following features. First, the algorithm is progressive, i.e., hello messages contain updated neighbors list in each consequent iteration. Second, we do not use explicit acknowledgments (ACKs) for the hello messages, which serve as an announcement of a node's information and as an implicit ACK to other nodes' information. The overhead of this approach is less than that of using explicit ACK for each received hello message, as one hello message may replace several ACKs.

Note that each node distributively decides to move into the next step (selecting MCHs) once no event triggers it to send more hello messages. As a consequence, the neighborhood discovery process will terminate at all nodes after a finite amount of time.

Step 2: Selecting MCHs. After neighborhood discovery is completed, the MCHs are to be selected. Since MCHs do more work than any typical node (for collecting, aggregating, and forwarding data), the selection criterion of MCHs is the node's remaining energy.

The proposed scheme for selecting MCHs can be summarized as follows. Each node maintains a table of remaining energy values of all its 1-hop neighbors (obtained from the neighborhood discovery step). All the nodes start the clustering process in the "undecided" state. Every node compares its remaining energy to those of its one-hop neighbors. If the node has the highest remaining energy in its neighborhood, it declares itself as an MCH and announces that to its neighbors. Any node that receives such a message stops competing for the role of an MCH[2]. A node that is either selected as an MCH or hears an MCH message switches to the "decided" state. This state means that the node is either an MCH or cannot be an MCH because it has heard an MCH message. The remaining undecided nodes repeat the above process until all the nodes are "decided."

The MCH selection process is illustrated in the example in Figure 3, where node 4 is an "undecided" node that is selected as an MCH based on its remaining energy. This is because nodes 3 and 6 become "decided" after hearing MCHs 2 and 7, respectively. On the other hand, node 5 has to wait for node 4 to decide because N_4 has more remaining energy than N_5. Since every "undecided" node continues the process until it is selected as MCH or until it hears from an MCH, it is clear that the clustering process converges.

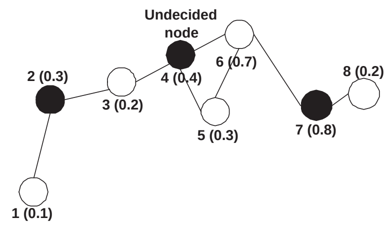

Fig. 3. An "undecided" node that is selected as an MCH (MCHs are indicated by dark circles).

The practicality of using remaining energy as a metric for selecting CHs was demonstrated in [4]. The authors formulated a simple energy model to keep track of the battery consumption of CHs and non-CH nodes. According to [4], a node can calculate the amount of energy used during its active time, and can find the percentage of its remaining energy.

Step 3: Selecting SCHs. The next step is to associate an SCH with each MCH, if possible. The purpose of having SCHs is to achieve diversity gain in inter-cluster communications through constructing cooperative MIMO nodes. To select SCHs, each MCH sends an "SCH invitation message" to the neighboring node whose neighbor list overlaps the most with that MCH's neighbor list. The rationale behind this criterion is to maximize the number of nodes that will eventually be served by both the MCH and SCH. At the same time, this criterion for selecting the SCH often forces the MCHs and SCHs to be close to each other, hence reducing the energy needed for coordinating the cooperative MIMO operation.

Upon receiving the first invitation message from an MCH, the invited node waits for a fixed duration of time (ζ) before making its decision. If the invited node receives more than one invitation within ζ, it chooses the closest inviting MCH (based on the strength of the received signal). The invited node announces its decision via an "SCH acceptance message." Upon receiving an "SCH acceptance message," the intended MCH confirms this association via an "SCH confirmation message." The purpose of this message is to inform the neighbors of this MCH that they should not expect subsequent "SCH invitation messages" from that MCH, so that they can move into the next step (cluster membership). The "SCH confirmation message" is sent at a fixed power (P_{inter}) that achieves network connectivity (we explain later how CMIMO results in a connected graph). This P_{inter} ensures that the "SCH confirmation message" reaches at least the closest MCH to the sending MCH node. This message plays a significant role in MCH-neighborhood

[2]For simplicity, we assume that no "ties" occur because remaining energy has floating-point representation and thus it is highly unlikely that two nodes will have the same remaining energy value. However, if integers are used to represent energy, then a tie can occur. Ties can be solved by any heuristic way, such as favoring nodes with smaller IDs.

discovery. Therefore, it includes the following fields: MCH ID, its SCH ID, and a list of MCHs that the MCH has already received SCH confirmation messages from. The other steps of this discovery approach are the same as in the neighborhood discovery explained previously.

The outcome of steps 2 and 3 over the topology of Figure 4 is shown in Figure 5. It should be noted that some MCHs may not have SCHs, especially if the topology is sparse. In this case, such an MCH cannot function as a virtual MIMO node.

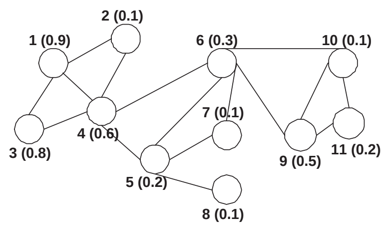

Fig. 4. Example topology that is used to illustrate cluster-formation process.

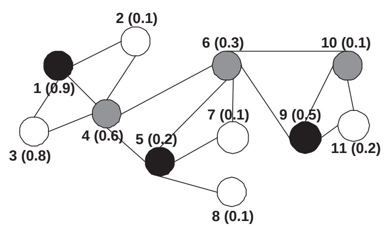

Fig. 5. WSN topology after step 3 (MCHs are indicated by dark circles, whereas SCHs are indicated by grey circles).

A node v that is neither an MCH nor an SCH autonomously decides to proceed to the next step (cluster membership) if: (1) v receives "SCH acceptance/confirmation messages" that are sent to/from all MCH neighbors, or (2) a fixed duration of time (θ) passes. This θ is used to avoid deadlock problems that may occur if the node does not receive "SCH acceptance/confirmation messages."

Step 4: Cluster membership. The final step in this phase is to have non-CH nodes decide on which cluster to join. Since any non-MCH node is a neighbor of one or more MCHs, such a node attempts to associate itself with its closest MCH by sending a "membership request message."

Upon receiving a "membership request message," the selected MCH waits for a fixed duration of time (ϕ), allowing other non-CH nodes to send their "membership request messages." After that, the MCH sends a "membership list message," announcing the IDs of the non-CH nodes that this MCH has accepted to be in its cluster. A node that does not find its ID in the "membership list message" resends (up to a maximum number of attempts) its "membership request message." It should be noted that there is some chance that the selected MCH will not include a given node in its updated "membership list message." This can be attributed to several reasons, e.g., the MCH does not intend to increase the number of its non-CH nodes above a specific threshold, etc. In such a case, the non-CH node tries to associate itself with the next closest MCH. Each MCH periodically announces its list of non-MCH nodes.

When an MCH announces a "membership list message," it includes the time division multiple access (TDMA) schedule that its SCH and non-CH nodes should follow in sending their data during intra-cluster communications. Once a new non-CH node joins a given MCH, the TDMA schedule is updated to include its ID, and is announced by the MCH via a "membership list message." As a convention, the first node to transmit according to the announced TDMA is the SCH followed by the non-CH nodes. Figure 6 shows a timing diagram that illustrates the above procedure over the topology of Figure 5. In that diagram, the first "membership list message" of node 5 contains one node, which is its SCH (node 6). A new node (node 7) sends a "membership request message" within the following ϕ duration, asking node 5 to join its cluster. An updated "membership list message" is then sent, announcing the new list, which now contains nodes 6 and 7. According to the updated TDMA schedule, the first node to transmit is the SCH (node 6), followed by node 7. During the following ϕ period, node 8 asks node 5 to join its cluster. The updated TDMA schedule includes node 8 as the third node to transmit its data, after nodes 6 and 7. The result of the "cluster membership" step over the topology of Figure 4 is shown in Figure 7.

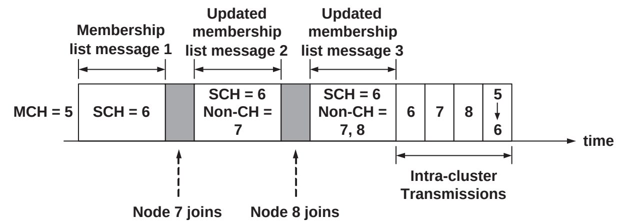

Fig. 6. Timing diagram that shows an announced TDMA schedule, updated forms of it, and intra-cluster transmissions.

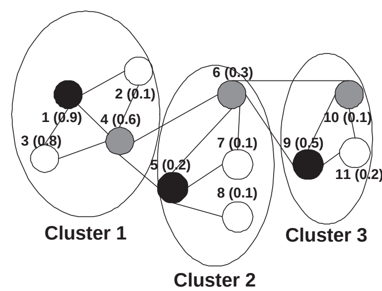

Fig. 7. WSN topology after step 4 (MCHs are indicated by dark circles, SCHs are indicated by grey circles, and non-CH nodes are indicated by white circles).

2) Intra-cluster communications: Each non-MCH node transmits its data to the MCH at a fixed power (P_{intra}) according to the TDMA schedule. After a non-CH node transmits its data, it goes to sleep until its next transmission turn. However, SCHs and MCHs do not go to sleep, as they may still receive data from other clusters (inter-cluster communications). Upon receiving data from the SCH and non-CH nodes, an MCH aggregates the received data, and sends the aggregated data to its SCH during the assigned TDMA slot. As a result, both the MCH and SCH will be ready for inter-cluster communications phase. Figure 6 shows a timing diagram for intra-cluster transmissions over the topology of Figure 5. In that diagram, the first node to transmit its data to the MCH (node 5) is its SCH (node 6), followed by node 7, and finally node 8. The last slot in that diagram is for node 5 to send the aggregated data to its SCH (node 6).

It should be noted that one reason why non-CH nodes send their data to the MCH only (and not to the SCH) is that some non-CH nodes may not have the ability to directly communicate with the SCH. For example, in Figure 7, node 8 is a non-CH node that cannot directly communicate with its SCH (node 6). However, by design all non-CH nodes must be able to directly communicate with the MCH.

3) Inter-cluster communications with cooperative MIMO: We now discuss how to establish an inter-cluster virtual MIMO link. The purpose of the following steps is to decide on the appropriate MIMO mode to be used. The operational details are explained in the following steps:

Step 1: An MCH in a given cluster accesses the channel using the CSMA/CA scheme. Once admitted, the MCH transmits a request-to-send (RTS) packet at a fixed power (P_{inter}) to the MCH and SCH (if any) in a next-hop cluster. The purpose of this RTS packet is to notify the CHs of the receiving cluster that the SCH of the source cluster will be sending another RTS packet. The MCH's RTS should be heard by the SCH of the transmitting cluster, so that the latter knows when to send its own RTS. Note that P_{inter} is chosen so that CMIMO produces a connected CH graph. Graph connectivity is discussed in detail in section IV.

Step 2: The SCH of the transmitting cluster sends its RTS to the CHs (MCH and SCH) of the receiving cluster. This RTS is sent at P_{inter}. It also serves as an indication to the transmitting MCH that the SCH has already heard the MCH's RTS. An example that illustrates control-packet exchanges between two clusters (steps 1 and 2) is shown in Figure 8.

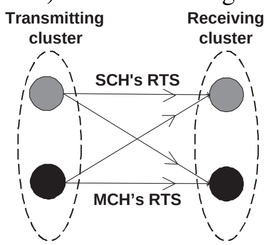

Fig. 8. Control-packet exchanges between two clusters (MCHs are indicated by dark circles, whereas SCHs are indicated by grey circles).

Step 3: Upon receiving the two RTS packets from the transmitting MCH and SCH, the receiving MCH and SCH estimate the channel gain between the CHs in the two clusters and communicate such information with each other. From that, the receiving CHs calculate the required power that is needed to communicate between the CHs of the transmitting and receiving clusters using one of four possible modes (SISO, MISO, SIMO, MIMO).

Step 4: The MCH and SCH in a receiving cluster calculate the optimal transmission mode that minimizes the total energy (which includes both transmission and circuit components) among the four modes. The expressions of the total energy consumption are discussed in section III. The graph is then pruned by eliminating all parallel links and keeping only the one with the least required total energy value, as shown in Figure 9. The MCH of that cluster then sends this information back to the CHs of the transmitting cluster via a clear-to-send (CTS) packet that is sent at P_{inter}.

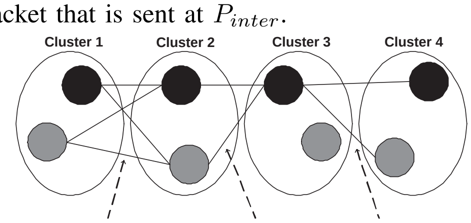

Fig. 9. Example of WSN after step 4 (MCHs are indicated by dark circles, whereas SCHs are indicated by grey circles).

Step 5: Upon agreeing about the transmission mode and transmission power via the control-packets exchange, the CHs of the transmitting cluster transmit their data to the CHs of the receiving cluster using the negotiated mode and power.

Step 6: The MCH of the receiving cluster acknowledges the reception of data via an ACK. If such an ACK is not received, the CHs of the transmitting cluster retransmit their data (up to a given maximum number of retransmissions).

Step 7: The transmitted data is forwarded hop-by-hop (where each hop represents one cluster) until it eventually reaches the sink. Our design uses an energy-efficient routing algorithm over the topology of virtual MIMO nodes to determine the end-to-end path that minimizes the total energy consumption among all possible paths between the transmitting CHs and the sink. This algorithm consists of two steps. In the first step, all pairs of virtual MIMO nodes that can directly communicate at P_{inter} using at least one of the four modes are determined. For a given pair, we establish as many parallel links as the number of feasible transmission modes between the two virtual MIMO nodes. We then prune the graph and keep only the one with the least required total energy value. In the second step, we run a modified version of Dijkstra's algorithm, where the weight of a link is taken as its total energy value determined from the first step. The returned path has the minimum sum of total energy values among all possible paths between the transmitting CHs and the sink.

We now explain how the sink explores the most energy-efficient path between the transmitting CHs and itself. The key idea here is that when an MCH sends an "SCH confirmation message" to its own non-MCH nodes and other neighboring MCHs, such a message is flooded by neighboring MCHs. This way, the sink will eventually learn about the various MCH nodes. The sink can then determine the optimal route (based on the total energy values) from every MCH to itself, and can inform every MCH with its next inter-cluster hop (in one message that is flooded throughout the network).

C. *Properties of CMIMO*

CMIMO has several features. First, it is completely distributed. This is because every node in the WSN independently takes its decisions based on local information. Second, at the end of the clustering process, a node is either a CH (master or slave) or a non-CH node that belongs to a cluster. In other words, the clustering process is guaranteed to terminate. This can be easily proven by noting that the node within the highest remaining energy in its neighborhood is elected as an MCH. Such an MCH then selects an SCH according to the neighbor list criterion. Next, each non-CH node selects one of the MCHs to join. Third, an SCH cannot belong to more than one cluster. This is because an invited SCH responds to only one of the received requests from MCHs. Fourth, the probability that two nodes within each other's cluster range are both MCHs is zero, i.e., MCHs are well distributed. This is attributed to the fact that MCHs are selected in an iterative manner and using a real-valued parameter (remaining energy).

III. Energy Model

In this section, we briefly analyze the energy consumption model of the CMIMO scheme. Following [5], the total power consumed for sending a packet consists of transmission and circuit powers. The transmission power for inter-cluster data transmissions is adjustable and is given by $P_t = (1+\delta)P_{out}$, where δ is a factor that depends on the drain efficiency of the power amplifier, and P_{out} is the total transmit power at the air interface. This P_{out} can be expressed as:

$$P_{out} = \gamma(M_t, M_r) N_o B N_f G_o M_l d^n \quad (1)$$

where $\gamma(M_t, M_r)$ is the required SNR at the receiver when M_t and M_r antennas are used for transmission and reception, respectively, N_o is the single-sided thermal noise power spectral density (PSD), B is the passband bandwidth, N_f is the receiver noise figure ($N_f \stackrel{\text{def}}{=} \frac{N_r}{N_o}$, with N_r being the PSD of the total effective noise at the receiver input), G_o is a constant that depends on the transmitter and receiver antenna gains, M_l is a link margin that compensates for hardware variations and other sources of interference, n is the path-loss exponent, and d is the transmitter-receiver distance. Note that $\gamma(M_t, M_r)$ depends on the target BER and the specific transmission mode.

As for the circuit power (P_c), it is given by [5]:

$$P_c \approx M_t(P_{DAC} + P_{mix} + P_{filt}) + 2P_{syn} + M_r(P_{LNA} + P_{mix} + P_{IFA} + P_{filr} + P_{ADC}) \quad (2)$$

where P_{DAC}, P_{mix}, P_{LNA}, P_{IFA}, P_{filt}, P_{filr}, P_{ADC}, and P_{syn} are the power consumption values for the digital-to-analog converter, the mixer, the low noise amplifier, the intermediate frequency amplifier, the active filters at the transmitter and the receiver sides, the analog-to-digital converter, and the frequency synthesizer, respectively.

Accordingly, the total energy consumption per bit is:

$$E_{bt} = \frac{P_t + P_c}{R_b} \quad (3)$$

where R_b is the bit rate. Using (1) and (2), E_{bt} can be written in terms of d, M_t, M_r, and R_b as follows:

$$E_{bt} = \frac{C_1\gamma(M_t, M_r)d^n + C_2M_t + C_3M_r + C_4}{R_b} \quad (4)$$

where C_1, C_2, C_3, and C_4 are circuit-specific constants.

Note that the transmission mode defined by M_t and M_r, $\gamma(M_t, M_r)$, and d have significant impacts on E_{bt}. A detailed discussion of this energy model and its implications on our work can be found in our technical report [6].

IV. CMIMO Design Issues

In this section, we discuss some issues related to the design of CMIMO. These include connectivity, listening cost, synchronization, reclustering, and MAC.

A. Connectivity

We now show how CMIMO constructs a connected graph of MCHs. Since any node in the network is either an MCH or within one hop from an MCH, the maximum number of hops between the MCHs of two neighboring clusters is three (see Figure 7). Therefore, to ensure that the MCH graph is connected, the following condition must apply: $R_t \geq 3R_c$, where R_c is the intra-cluster range (the distance between an MCH and the farthest non-MCH node in its cluster) and R_t is the inter-cluster range between two neighboring MCHs. As a result, each MCH can at least reach the closest MCH using P_{inter} that satisfies the above condition.

We now compare the connectivity condition for CMIMO with that of a previously proposed clustering scheme, namely HEED [7]. In HEED, the system model is close to that of CMIMO, but with one CH per cluster. The authors of [7] found that any two CHs can communicate if $R_t \geq 6R_c$ so that the network is connected. CMIMO's connectivity requirement gives higher chances for inter-cluster MCHs to take place than HEED, as a smaller range is reserved for inter-cluster communications (i.e., there is more spatial reuse).

B. Listening Cost

We now discuss several approaches that can be used by CMIMO to reduce the listening cost of active nodes. To ensure inter-cluster routing, MCHs and SCHs should always be available. On the other hand, non-CH nodes can be put to sleep after they send their data to their MCHs. CHs can also follow some duty cycle to reduce their energy consumption, as follows. When a packet is intended to a CH, the CH should wakeup for a duration of time that is needed to receive the coming data. This, however, requires coordination between the CH and its non-CH nodes. Several general-purpose approaches for coordinating communications can be used in this context, such as S-MAC [8], T-MAC [9], and TRAMA [10].

C. Synchronization

Extensive research has been done on quantifying the tradeoff between implementing synchronous network operation and the overhead and inaccuracy associated with such operation. Our CMIMO design can be in either synchronous or asynchronous modes.

Several approaches were proposed to solve the synchronization issue. One of them suggests using a reference-broadcast synchronization (RBS) technique [11], which we can adapt to CMIMO as follows. An MCH asks one of its cluster nodes (except the SCH) to send RBS beacons. After exchanging the RBS beacons, the MCH and SCH start sending the data simultaneously to the CHs in a receiving cluster. Another approach is to use the transmission delay and channel estimation scheme proposed in [12]. The main drawbacks of these approaches are the large number of generated messages, the long elapsed time in overall synchronization, and that they do not consider the energy requirement of sensor nodes.

To overcome the above drawbacks, we proposed a solution to the synchronization issue. The key idea of this solution is to select a node in the network to act as a beacon cluster head (BCH). Such a node sends beacons to its neighboring MCHs so that they adjust the start time of their frames accordingly. A detailed description of the proposed solution can be found in [6].

D. Reclustering

The key idea for reclustering is that once the remaining energy for any MCH falls below a specific threshold (e.g., 20% of its initial value), this MCH sends a "reclustering" message to its neighboring MCHs at power level P_{inter}. This power value ensures that the reclustering message sent by the MCH is heard by its non-CH nodes, its SCH, and its neighboring MCHs. The purpose of these reclustering messages is to inform other nodes that this MCH requests reclustering. It should be noted that these reclustering messages are similar to the "hello" messages used in the neighborhood discovery process in the "cluster formation" phase, except that the reclustering messages are sent at a higher power (P_{inter}). As a result, the neighboring MCHs that hear these messages respond as in the previously discussed neighborhood discovery process. The rationale behind restricting the reclustering request to MCHs is that in most cases, MCHs are the ones that deplete their batteries first (before SCHs and non-CH nodes), as MCHs are responsible for aggregating data, sending it to the SCH, and forwarding it to CHs in neighboring clusters.

E. Medium Access Control (MAC)

We now discuss two issues related to the MAC layer, namely how to have reliable communications (i.e, taking packet losses into account) and how to select transmission ranges for intra- and inter- cluster communications. Recall that the main emphasis of our work is on the clustering/routing aspects of cooperative MIMO, which take place in layers that are above the MAC layer.

Reliable communication is established by using ACKs at the MAC layer for all network transmissions. These include transmissions for neighborhood discovery (where hello messages represent implicit ACKs for previously heard hello messages), between a non-CH node and its MCH (e.g., a "membership request message" is acknowledged by a "membership list message"), between an SCH and its MCH (e.g., an "SCH acceptance message" is acknowledged by an "SCH confirmation message"), and between an SCH/MCH in a transmitting cluster and an SCH/MCH in a receiving cluster (e.g., MCH's and SCH's RTSs are acknowledged by MCH's CTS). In addition, CMIMO retransmits the packets that have not been acknowledged (limited to a specific number of retransmissions) to increase the chances of correctly receiving these packets.

For selecting transmission ranges, one approach is to use code division multiple access (CDMA) technique [13], where different coding schemes for intra- and inter-cluster transmissions are used. However, the hardware implications of such a technique may render it infeasible. Therefore, we propose another approach, in which the transmission range of inter-cluster communications (R_t) is selected a priori. The intra-cluster range is then selected so that the connectivity condition ($R_t \geq 3R_c$) is satisfied. We argue that this approach has less complexity than the CDMA approach.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of CMIMO via simulations. We also compare it with the DCA scheme [14], which is one of the fundamental clustering schemes. The purpose of choosing DCA to compare our scheme with is that DCA resembles CMIMO in the criterion used to select CHs. Also, the primary goal of this comparison is to demonstrate the benefits of using cooperative MIMO over a single-antenna system (DCA). Recall that CMIMO adapts the transmission mode and transmission power on a per-packet basis. On the other hand, all transmissions in DCA take place using the SISO mode, where each cluster has only one CH.

For both CMIMO and DCA, we use the following simulations setup, unless stated otherwise. We consider 100 nodes that are randomly deployed within a square of 1000 meter $\times$ 1000 meter. The sink is located outside (to the right of) the square field. Each node generates packets according to a Poisson process of rate λ[3]. The sink is the only node that is equipped with two antennas. The rationale behind this design is to achieve MIMO gain for the last hop. Multi-hop operation based on an energy-efficient routing is used for inter-cluster communications. The values of $\gamma(M_t, M_r)$ that are required to achieve a BER of 0.001 are taken from [5]: 24.4 dB for SISO, 10.6 dB for SIMO, 14.1 dB for MISO, and 6.9 dB for MIMO. For the wireless channel, we assume Rayleigh fading model along with a distance-dependent path loss, which has a power falloff of d^4. R_c is set to 60 meters, and R_t is set based on the connectivity condition ($R_t = 3R_c$). For figures that vary either R_c or R_t, the other range is still determined based on the connectivity condition. Other simulation parameters are given in Table III. We repeated each experiment 100 times with different seed numbers and averaged the results. Our results are based on simulation experiments conducted using CSIM [15].

A. Energy Consumption

Figure 10 shows the impact of inter-cluster range (the maximum range for direct communications between two clusters) on the total energy consumption for CMIMO and DCA. Such a range is controlled via P_{inter}. The results illustrate that the total energy consumption increases with the inter-cluster range. CMIMO outperforms DCA in energy consumption. The superiority of CMIMO over DCA becomes more significant as the inter-cluster range increases. The reason is that increasing the inter-cluster range forces clusters that are far away from each other to communicate using multi-antenna modes. In this case, the transmission energy dominates the circuit energy, making MIMO/MISO/SIMO more energy-efficient than the SISO mode. Thus, exploiting cooperative MIMO systems is more energy-efficient under large inter-cluster ranges.

[3]Other traffic models, such as Markov-modulated Poisson process (MMPP), can also be used by CMIMO to generate packets.

TABLE III

SIMULATION PARAMETERS

Data-packets size	2000 bytes
Control-packets size	20 bytes
R_b	1 Mbps
λ	20 packets/sec
P_{DAC}, P_{ADC}	15 mW
P_{mix}	30.3 mW
P_{filt}, P_{filr}	2.5 mW
P_{syn}	50 mW
P_{LNA}	20 mW
P_{IFA}	2 mW
M_l, N_f	10 dB

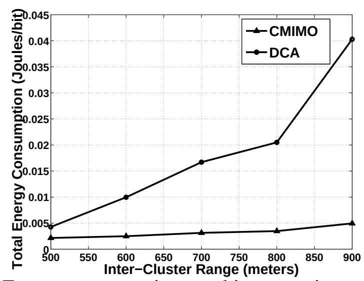

Fig. 10. Energy consumption per bit versus inter-cluster range.

We also study the energy consumption of CMIMO and DCA under different network sizes. The variation in the transmission distances leads to different ratios between transmission and circuit energies. Specifically, the transmission energy increases with distance while the circuit energy remains roughly the same. Figure 11 depicts the energy consumption versus the network-side length, i.e., the length of the square. We conclude from this figure that CMIMO results in better energy performance than DCA, especially under larger areas. This is attributed to the fact that under large network areas, the transmission energy dominates the circuit energy, forcing CMIMO to use MIMO/SIMO/MISO modes to minimize the transmission energy for data packets.

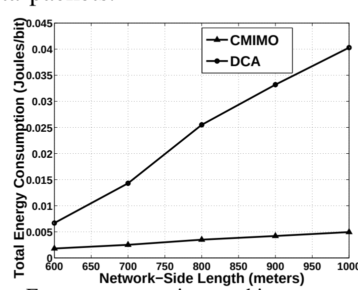

Fig. 11. Energy consumption per bit versus network size.

B. Network Throughput

We define throughput as the total number of correctly received bits within a duration of time. The throughput performance of CMIMO and DCA is shown in Figure 12. The figure shows the network throughput versus the total number of nodes in the network. It reveals that the energy-efficient operation of CMIMO does not reduce its network throughput, as compared to that of DCA. Note that throughput increases with the total number of nodes.

C. Histogram of Transmission Modes

Figure 13 shows the fraction of time (as a percentage) that each transmission mode (MIMO, MISO, SIMO, and SISO)

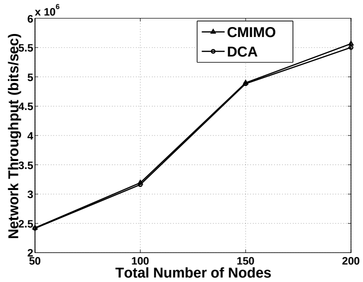

Fig. 12. Network throughput versus total number of nodes.

is used at different intra-cluster ranges. Recall that the intra-cluster range is the distance between an MCH and the farthest non-MCH node in its cluster. As the intra-cluster range increases, fewer clusters are formed. Thus, transmission distances between CHs become larger and circuit energy becomes less significant than transmission energy, which results in making other modes than SISO more favorable.

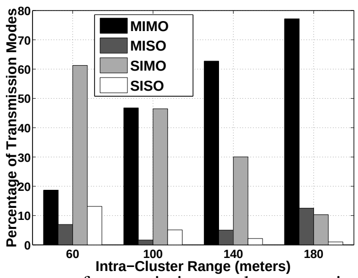

Fig. 13. Percentage of transmission modes versus intra-cluster range.

The inter-cluster range also affects how often various transmission modes are used. Recall that the inter-cluster range represents the maximum range for direct (1-hop) communications between two clusters. Figure 14 shows that the MIMO mode is used more frequently at higher inter-cluster ranges. This is due to the fact that the distances between the CHs of different clusters become larger, and thus the transmission energy dominates the circuit energy, resulting in more chances for the CHs to communicate using the MIMO mode. On the other hand, the percentage of using the SISO transmission mode decreases as the inter-cluster range increases.

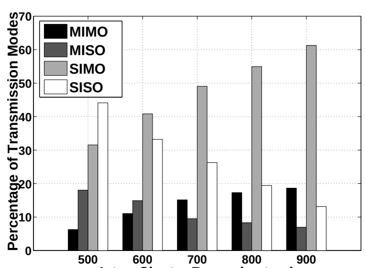

Fig. 14. Percentage of transmission modes versus inter-cluster range.

D. Number of Hops

We now study the impact of inter-cluster range and network area on the number of hops used for inter-cluster communications to the sink. Figure 15 shows a histogram of the number of hops needed for routing under different inter-cluster ranges. As expected, under small inter-cluster ranges (e.g., 300 meters), routes with large numbers of hops exist (10 hops). On the other hand, when the inter-cluster range becomes large (e.g., 900 meters), fewer hops are needed (2 hops).

The impact of the network area on the number of hops is shown in Figure 16. The figure reveals that large network areas require more hops between the communicating CHs and the sink. For example, a path with 18 hops exists when the network field is 1000 meter $\times$ 1000 meter. However, no more than 9 hops are needed for a 600 meter $\times$ 600 meter network.

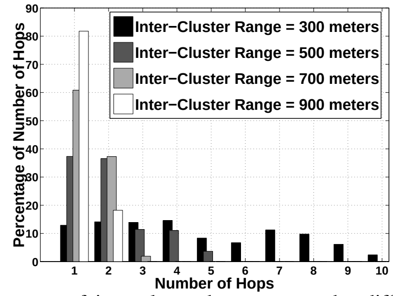

Fig. 15. Histogram of inter-cluster hop count under different inter-cluster ranges.

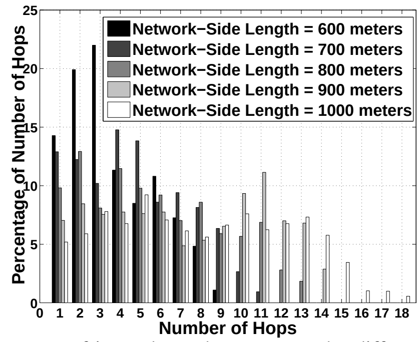

Fig. 16. Histogram of inter-cluster hop count under different network areas.

VI. RELATED WORK

In this section, we briefly describe recent work on cooperative MIMO systems and node clustering in WSNs. Several cooperative MIMO systems for WSNs have been proposed in the literature (e.g., [5], [12], [16]). In these systems, several single-antenna nodes cooperate on information transmission/reception to achieve energy-efficient communications. The authors in [5] studied a cooperative MIMO scheme with Alamouti code for *single-hop* transmissions in WSNs. A new cooperative MIMO scheme for delay and channel estimation was proposed in [12]. This scheme uses two transmitting sensors and space-time block code to provide transmission diversity in distributed WSNs. It should be noted that neither antenna arrays nor transmission synchronization were used. In [16] energy efficiency and training overhead of cooperative MIMO WSNs were analyzed.

In all these contributions, clustering and multi-hop routing were not taken into consideration, which limits the scalability of these schemes in large WSNs.

One of the fundamental clustering schemes is the distributed clustering algorithm (DCA) [14], which clusters nodes in an iterative way. In DCA, nodes divide themselves into groups according to a weight-based criterion. The main assumptions behind DCA are: (1) the network topology is static, and (2) each transmitted message is correctly received by all neighbors within a specific duration of time. As the first assumption is reasonable for WSNs, the second one opens several issues about reliability and how such a scheme can deal with collisions. The key idea in DCA is that a node decides its role in the network after it hears the decisions of its neighbors that have higher weights.

It should be noted that all these clustering schemes do not exploit cooperative MIMO with clustering. Moreover, some of these schemes do not consider multi-hop communications and early data aggregation to save energy. On the other hand, CMIMO is a cooperative-MIMO based scheme, where the decisions of the nodes that follow CMIMO are based on MIMO-related parameters. Moreover, multi-hop operation and early data aggregation are considered in our design.

The authors in [17] extended the LEACH scheme to build a cluster-based cooperative MIMO scheme for WSNs, namely MIMO-LEACH. The main differences between CMIMO and MIMO-LEACh can be summarized as follows. First, MIMO-LEACH is based on an existing clustering scheme (LEACH) that does not take into account MIMO operations in the clustering process, whereas CMIMO has its own clustering scheme that exploits MIMO operations in its selecting criteria. Second, MIMO-LEACH contains one CH per cluster, which is responsible for aggregating data, and broadcasting it to two other "cooperative" nodes. These nodes are responsible for forwarding the data to the CH of a receiving cluster. It is clear that in [17], three nodes (one CH and two cooperative nodes) are needed to aggregate and forward the data, whereas only the two CHs (MCH and SCH) are the ones that perform these functions in CMIMO. As a result, MIMO-LEACH involves more overhead than CMIMO. Finally, four transmission modes are available in CMIMO for each link between any two clusters. However, in MIMO-LEACH no complete MIMO communications (2×2) exist among the network, as the best transmission mode that can be used is SIMO/MISO.

A detailed discussion of the above research as well as other related works can be found in [6].

VII. Conclusions and Future Extensions

In this paper, we proposed a distributed and adaptive clustering/routing scheme (CMIMO) to minimize the total energy consumption for a multi-hop WSN. CMIMO produces clusters that have two CHs, which are responsible for routing traffic between clusters (i.e., inter-cluster communications). The proposed CMIMO scheme has the ability to adapt the transmission mode (SISO, MISO, SIMO, MIMO) and the transmission power on a per-packet basis for inter-cluster transmissions. We studied the performance of the CMIMO scheme via simulations. The results indicate that the proposed scheme achieves a significant improvement in the overall energy consumption of the network compared to non-adaptive WSNs that are designed with one CH per cluster and using SISO mode (e.g., WSNs under the DCA scheme). The significance of CMIMO becomes more attractive under high intra/inter-cluster ranges and network sizes. CMIMO supports multi-hop scenarios and performs better when the number of hops increases.

One possible extension to our work is to study the impact of the modulation order (b) on the total energy consumption. The value of b represents the number of bits per symbol, which varies with the modulation scheme. As shown in Figure 18 [5], a higher value of b necessitates a higher $\gamma(M_t, M_r)$, i.e., more transmission power is required. However, it also means a higher transmission rate, which in turn reduces the energy consumption. The confluence of the two effects determines the optimal modulation order (b^*). As shown in Figure 17, b^* generally decreases with the transmitter-receiver distance, as transmission energy becomes more dominant.

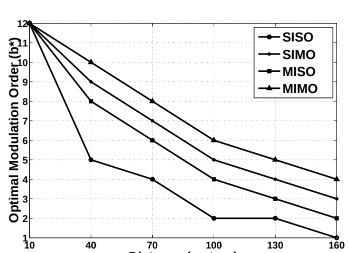

Fig. 17. b^* vs. transmitter-receiver distance for various antenna configurations.

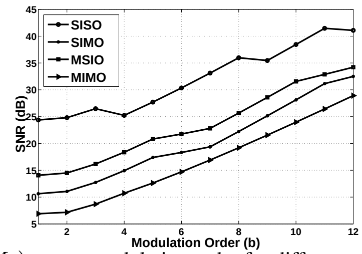

Fig. 18. $\gamma(M_t, M_r)$ versus modulation order for different transmission modes (obtained from [5]).

In this work, we mainly focused on the diversity gain. However, we plan on exploiting the other gains in future research. Our goal is to combine different types of MIMO gains, allowing for dynamic switching between diversity, array, and multiplexing modes, so as to maximize a utility function that depends on both energy consumption and throughput.

Although we only considered cooperative MIMO with two CHs per cluster, our design can be in principle extended to higher-order MIMO systems. Note that going beyond 2×2 systems (e.g. three CHs per cluster; MCH1, SCH1, and SCH2), results in more overhead on intra-cluster part. As a result, it is not clear if adding more CHs per cluster improves the overall performance compared to that of 2×2 systems. This issue is left to future work.

References

[1] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, Cambridge, U.K., 2003.
[2] *http://www.ee.ucla.edu/∼babak/research.html*.
[3] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2349–2360, Sept. 2005.
[4] O. Younis and S. Fahmy, "An experimental study of routing and data aggregation in sensor networks," in *Proceedings of the IEEE International Workshop on Localized Communication and Topology Protocols for Ad Hoc Networks (LOCAN)*, Nov. 2005.
[5] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-efficiency of MIMO and cooperative MIMO in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1089–1098, Aug. 2004.
[6] M. Z. Siam, M. Krunz, and O. Younis, "Energy-efficient clustering/routing for cooperative MIMO operation in sensor networks," Tech. Rep. TR-UA-ECE-2008-2, University of Arizona, Aug. 2008, http://www.ece.arizona.edu/∼krunz.
[7] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, Oct.-Dec. 2004.
[8] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the IEEE INFOCOM Conference*, June 2002, pp. 1567–1576.
[9] T. V. Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the First ACM SenSys Conference*, Nov. 2003, pp. 171–180.
[10] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," in *Proceedings of the First ACM SenSys Conference*, Nov. 2003, pp. 181–193.
[11] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, Dec. 2002, pp. 147–163.
[12] X. Li, "Energy efficient wireless sensor networks with transmission diversity," *IEE Electronics Letters*, vol. 39, no. 24, pp. 1753–1755, Nov. 2003.
[13] T. Shu, M. Krunz, and S. Vrudhula, "Joint optimization of transmit power-time and bit energy efficiency in CDMA wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3109–3118, Nov. 2006.
[14] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, June 1999, pp. 310–315.
[15] Mesquite Software Incorporation, *http://www.mesquite.com*.
[16] S. K. Jayaweera, "Energy analysis of MIMO techniques in wireless sensor networks," in *Proceedings of the 38th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2004.
[17] Y. Yuan, M. Chen, and T. Kwon, "A novel cluster-based cooperative MIMO scheme for multi-hop wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2006, no. 72493, pp. 1–9, May 2006.