A Multi-Channel Cooperative MIMO MAC Protocol for Wireless Sensor Networks

Dawei Gong, Miao Zhao and Yuanyuan Yang
Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

Abstract—Recently, several multi-channel MAC protocols have been proposed for wireless sensor networks (WSNs) to improve network capacity and boost energy efficiency. In addition, cooperative multiple-input multiple-output (MIMO) technique has been shown to be able to significantly enhance the energy efficiency of WSNs if properly configured. However, these two promising techniques have not been jointly utilized in WSNs. In this paper, we explore such a joint design by proposing a novel MAC protocol for WSNs that takes advantage of both multiple channels and cooperative MIMO. Specifically, sensor nodes in a WSN are organized into clusters and each cluster head selects some cooperative nodes to help forward traffic to or receive from other clusters by utilizing cooperative MIMO technique. For intra-cluster communications, different channels are assigned to adjacent clusters to reduce collisions, while for inter-cluster communications, cooperative MIMO links are scheduled to improve energy efficiency and concurrent transmissions are enabled by assigning different channels to them. We carry out extensive simulations and the results demonstrate that the proposed protocol can significantly increase throughput and improve energy efficiency compared to other schemes. For example, for a WSN with 150 nodes in a 250m × 250m field, with five or more available channels, the protocol can achieve over three times saturated throughput while saving up to 17% energy per bit for inter-cluster communications.

Keywords: Wireless sensor networks (WSNs), multiple-input multiple-output (MIMO), cooperative MIMO, channel assignment, medium access control (MAC).

I. INTRODUCTION AND PREVIOUS WORK

Wireless sensor networks (WSNs) have been deployed for a wide range of applications, such as medical care, smart building and various industrial applications. Typically, sensor nodes are provisioned with low-capacity, irreplaceable batteries, while communications is the major source of energy consumption in WSNs. Therefore, improving the energy efficiency of communications has been the key objective of a multitude of research efforts in WSNs. One approach to achieving this objective is to improve the Medium Access Control (MAC) efficiency by reducing control overhead and avoiding unnecessary energy expenditure caused by overhearing and collisions. Moreover, designing MAC protocols jointly with cross layer information can further improve the efficiency. Another effective approach is to boost the energy efficiency in the physical layer by adopting the cooperative Multiple Input Multiple Output (MIMO) technology. We first discuss these approaches briefly in the following subsections.

A. Related Work on MAC Protocols

Medium access control is critical in WSNs where a large number of sensor nodes access the network and transmit the sensed data. With limited power supply on sensor nodes, traditional MAC protocols for wireless networks may not work well in WSNs as a lot of energy would be consumed on idle listening, overhearing, collision and control overhead [1].

To reduce the energy consumed by overhearing and idle listening, S-MAC was proposed in [2], in which sensor nodes sleep most of the time and wake up only when the nodes need to transmit/receive messages. However, as the traffic load increases, the energy consumed due to collisions becomes non-negligible. To alleviate the intense collisions, a quasi-static, multi-channel MAC protocol for WSNs was introduced in [3], in which each sensor node assigns the least used channel to itself and the order of the assignment is determined by its remaining energy level. In [4], [5], [6], several dynamic multi-channel MAC protocols were presented, where sensor nodes negotiate the communication channel before each transmission. A time synchronization mechanism was also provisioned in [5], [6] to avoid the channel assignment conflict problem, which refers to the situation that two interfering links are assigned the same channel when there exist other interference-free channels.

In practice, there are usually not a sufficient number of orthogonal channels in WSNs. In this case, the overhead caused by channel contention in the above contention-based protocols would waste too much channel capacity under heavy traffic load. Thus, this type of MAC protocols cannot achieve good performance in terms of throughput and energy efficiency under heavy traffic. On the contrary, time-division scheduling can fully avoid performance degradation under heavy traffic load. A time-division multiple access (TDMA) MAC protocol for WSNs was proposed in [8], in which the time slot allocation problem was proved to be NP-hard and a heuristic algorithm based on graph edge coloring was introduced to allocate time slots. In addition, in [9] a multi-channel time-division MAC protocol was presented for ad-hoc networks to take full advantage of time-division scheduling and multiple channels, where both time slot and communication channel are negotiated before each transmission.

B. Related Work on MAC Layer Joint Design

MAC protocols may be jointly designed by taking account of other available information in the network. In particular, traffic patterns of many specific applications in WSNs are predictable. Such predictability can be utilized in multi-channel MAC protocol designs to further enhance energy efficiency. In [7], a multi-channel MAC protocol was presented, where the traffic requirement at each node is used as the metric for channel assignment. Nodes with higher traffic demands get channel assigned earlier so that the overall conflict can be greatly reduced. A tree-based multi-channel MAC protocol was introduced in [11] for data gathering applications in WSNs, which organizes nodes into vertex disjoint sub-trees.
Channels are assigned to different sub-trees to avoid inter-subtree interference. Then sensed data is forwarded to the data sink along the subtrees. A heuristic algorithm was given for forming subtrees and minimizing intra-tree conflict.

Clustering is another effective approach to boosting energy efficiency in WSNs by reducing redundant transmissions. Assigning different channels to neighboring clusters can further improve the performance by eliminating interference. In [10], a distributed channel assignment scheme for clustered WSNs was presented, in which cluster heads assign channels for each cluster member and the order of assignment depends on the weight of the cluster head. A dynamic multi-channel MAC protocol for clustered WSNs was proposed in [1], where each sensor node negotiates the communication channel with its cluster head over the default control channel before data transmissions.

C. Related Work on Physical Layer

In recent years, multiple-input multiple-output (MIMO) technology has been widely adopted by various wireless networks due to its promising potential to enhance network bandwidth utilization and improve transmission efficiency at physical layer. Though it may be difficult to mount multiple antennas on a sensor node due to the limited size of sensor nodes, WSNs can utilize MIMO technology by organizing multiple nodes into a MIMO array, which is called cooperative MIMO. It was shown in [12] that considerable energy saving can be achieved by using cooperative MIMO when the transmission distance is greater than a threshold. In [13], [14], it was further illustrated that cooperative MIMO can outperform single-input single-output (SISO) even for short range transmissions by using optimal modulation parameters. A cluster-based cooperative MIMO protocol for WSNs was proposed in [15], in which the best cooperative nodes are selected for inter-cluster communications. Specifically, cluster heads first broadcast an inter-cluster message to these cooperative nodes and then these cooperative nodes transmit the message to the destination cluster head using cooperative MISO. In [16], a clustering/routing infrastructure with cooperative MIMO for WSNs, named CMIMO, was presented. In this work each cluster selects two cluster heads: master cluster head and slave cluster head. The master cluster head operates as a normal cluster head, while the slave cluster head is used for cooperative MIMO communications. TDMA is used for intra-cluster communications and negotiation is conducted among the cluster heads of neighboring clusters to select the most energy efficient transmission mode for inter-cluster communications.

To the best of our knowledge, most existing MAC protocols with cooperative MIMO communicate over a single channel, and there is no scheduling scheme that combines both cooperative MIMO and multiple channels so far. Such a joint design has the potential to significantly improve the performance of intra-cluster communications and multi-hop inter-cluster communications. Based on this observation, in this paper we propose a multi-channel MAC protocol with cooperative MIMO (referred to as MCCMIMO) for WSNs to reduce the energy consumption on collision and improve system throughput. We first introduce a novel clustering algorithm that aims at minimizing intra-cluster distance. We then propose the channel assignment algorithms for both clusters and cooperative MIMO links among clusters, aiming at prolonging network lifetime. After that, we present a synchronized MAC protocol for intra-cluster and inter-cluster communications. Finally, we conduct extensive simulations to evaluate the performance of MCCMIMO in terms of energy efficiency, throughput and end-to-end delay. As will be seen from the simulation results, the proposed protocol outperforms other compared schemes by a big margin.

The rest of this paper is organized as follows. Section II presents the cluster formation and cooperative MIMO link selection algorithms. Section III gives the channel assignment algorithms. Section IV describes the details of the MAC protocol. Section V analyzes the energy efficiency of the proposed protocol. Section VI provides the simulation results and Section VII concludes the paper.

II. CLUSTER FORMATION AND COOPERATIVE MIMO LINK SELECTION

The overall framework of MCCMIMO is shown in Fig. 1. In MCCMIMO, sensor nodes are first organized into clusters and cooperative MIMO links among clusters are selected for inter-cluster communications. Channels are then assigned to clusters and cooperative MIMO links respectively to reduce interference and enable concurrent communications for both intra-cluster and inter-cluster communications. Data transmissions are carried out in MAC intervals, with intra-cluster and inter-cluster communications occurring sequentially in each MAC window. The cluster formation and channel assignment algorithms are invoked periodically based on the status of remaining energy of sensor nodes.

<table>
<thead>
<tr>
<th>Cluster Formation</th>
<th>Cooperative Node Selection</th>
<th>Channel Assignment</th>
<th>MAC Interval 1</th>
<th>MAC Interval 2</th>
<th>...</th>
</tr>
</thead>
</table>

Fig. 1. The framework of MCCMIMO.

During the cluster formation and cooperative MIMO link selection period, all communications are over the default channel and each node uses the carrier sense multiple access with collision avoidance (CSMA/CA) to access the channel. We assume that no collision occurs during this time and the receiver can estimate the distance from the transmitter using the received signal strength indication (RSSI) of the received packets. We also assume that sensor nodes are able to transmit packets at two different power levels: \( P_{\text{low}} \) and \( P_{\text{high}} \). The cluster formation and cooperative MIMO link selection state consists of five phases: 1) 1-hop neighbor discovery phase, in which each node broadcasts its remaining energy to 1-hop neighbors; 2) 1-hop weight discovery phase, in which each node calculates and broadcasts its weight to 1-hop neighbors; 3) cluster formation phase, in which clusters are constructed based on the information received in 1-hop weight discovery phase; 4) cluster neighbor discovery phase, during which each cluster member notifies its cluster head about the cluster
information of all its 1-hop neighbors such that the cluster head knows all its neighboring clusters and which nodes are adjacent to them; 5) cooperative MIMO link selection phase, in which each pair of neighboring clusters select the optimal cooperative MIMO links for inter-cluster communications. The details of these phases are discussed in the following subsections.

A. 1-Hop Neighbor Discovery

In this phase, each node broadcasts a Residual Energy (RE) message including its remaining energy to its 1-hop neighbors at power \( P_{\text{low}} \). The format of the message is shown in Fig. 2(a). Each sensor node keeps listening to the default channel to receive the broadcast messages from all its 1-hop neighbors. Once receiving a RE message from a neighbor, a node adds an entry to its 1-hop neighbor list including the neighbor’s residual energy and the estimated distance. When the channel is idle for \( CW_{\max} \) time, a node concludes that it has received RE messages from all neighbors and moves to the 1-hop weight discovery phase.

<table>
<thead>
<tr>
<th>MAC Type</th>
<th>Subtype</th>
<th>Node ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>ID</td>
<td>Remaining Energy</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>MAC Type</th>
<th>Subtype</th>
<th>Node ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>ID</td>
<td>Weight</td>
</tr>
</tbody>
</table>

(b)

Fig. 2. The formats of Residual Energy (RE) message and Node Weight (NW) message. (a) RE message. (b) NW message.

B. 1-Hop Weight Discovery

In 1-hop weight discovery phase, the weight of each node is calculated and broadcast to prepare for cluster formation in the next phase. As communication hubs, cluster heads consume energy faster than cluster members. Therefore, the remaining energy should be reflected in the cluster head election weight. Moreover, as the majority of transmissions in most of clustered WSNs are intra-cluster communications, and the minimum communication energy consumption is positively correlated to the transmission distance, the average distance to 1-hop neighbors should be reflected in the weight as well. As a result, the weight of each node is defined as in Equation (1), in which \( N(i) \) is the 1-hop neighbor set of node \( i \), \( d(i,j) \) denotes the distance between node \( i \) and node \( j \) and \( E(i) \) is the energy of node \( i \).

\[
\text{Weight}(i) = \left( \sum_{j \in N(i)} d(i,j) / (E(i) | N(i)) \right)
\]

(1)

After the weight calculation, each node broadcasts a Node Weight (NW) message at power \( P_{\text{low}} \). The format of the Node Weight message is shown in Fig. 2(b). Similar to the 1-hop neighbor discovery phase, each node keeps listening and updating its 1-hop neighbor list with the newly received weight information. A node enters into the cluster formation phase after receiving all NW messages from its neighbors in the 1-hop neighbor list.

C. Cluster Formation

In this phase, sensor nodes with the minimum weight in their 1-hop neighborhoods elect themselves as cluster heads. The cluster head election procedure is executed on each node as every node is aware of the weights of its 1-hop neighbors. A node broadcasts a Cluster Head Announcement (CHA) message to announce itself as a cluster head. A node sends out a Cluster Join (CJ) message to join a cluster. The formats of CHA and CJ messages are shown in Fig. 3(a) and Fig. 3(b), respectively. At the beginning of this phase, the node with the minimum weight broadcasts a CHA message. Other nodes keep listening until all the 1-hop neighbors with lower weights have updated their statuses. For each sensor node, if only one neighbor with lower weight becomes a cluster head, the node chooses it as the cluster head by sending out the CJ message; if more than one neighbors with lower weight announce themselves as cluster heads, the node chooses the closest one and joins its cluster; Otherwise, if all neighbors with lower weight announce themselves as cluster members by sending out CJ messages, the node elects itself as a cluster head. The cluster head election procedure terminates when all nodes have updated their statuses. Fig. 4 illustrates the cluster formation result for a randomly generated WSN with 150 nodes in a 250m \( \times \) 250m field.

<table>
<thead>
<tr>
<th>MAC Type</th>
<th>Subtype</th>
<th>Node ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>ID</td>
<td>Cluster ID</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>MAC Type</th>
<th>Subtype</th>
<th>Node ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>ID</td>
<td>Cluster ID</td>
</tr>
</tbody>
</table>

(b)

Fig. 3. The formats of cluster head announcement (CHA) message and cluster join (CJ) message. (a) CHA message. (b) CJ message.

Fig. 4. A WSN clustered by MCCMIMO, in which cluster heads are denoted by red dots, cluster members are denoted by blue dots, and cluster members are connected to their cluster heads via lines.

D. Cluster Neighbor Discovery

After the cluster formation phase, each cluster member updates the cluster head information of its 1-hop neighbor list.
by overhearing the CHA/CJ messages of its 1-hop neighbors. In this phase, all cluster members send a Cluster Forward (CF) message to their cluster heads, in which the updated 1-hop neighbor list is included. The format of CF message is shown in Fig. 5. After receiving all the CF messages from its cluster members, a cluster head knows all the neighboring clusters and their boundary nodes, and moves to the cooperative MIMO link selection phase.

<table>
<thead>
<tr>
<th>MAC Type</th>
<th>Subtype</th>
<th>Node ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>ID</td>
<td>Neighbor’s ID, cluster, distance to neighbor 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Neighbor’s ID, cluster, distance to neighbor 2</td>
</tr>
</tbody>
</table>

Fig. 5. The format of cluster forward (CF) message.

E. Cooperative MIMO Link Selection

In this phase, each cluster head negotiates with the cluster heads of neighboring clusters to select the optimal cooperative MIMO links, as more than one such links may exist between two neighboring clusters. For instance, in Fig. 6, both ((3,4),(6,7)) and ((5),(8,9)) can act as the cooperative MIMO link between cluster A and cluster B. In general, on one hand, the cooperative MIMO link with high energy efficiency should be selected to save transmission energy; on the other hand, a link with low remaining energy should not be selected even if it has high energy efficiency, to avoid exhausting the link. We define $E_f(l)$ as the energy efficiency of a cooperative MIMO link $l$, which is determined by Equation (7) to be given in a later section. We use $En(l)$ to represent the remaining energy of link $l$, which is set to the least remaining energy of all nodes involved. To balance the effect of both factors, an empirical influence factor $\delta$ ranging from 0 to 1 is introduced, which can be adjusted according to the type of the application. Thus the weight of an cooperative MIMO link is defined as follows

$$Weight(l) = \delta E_f(l) + (1 - \delta) En(l)$$  (2)

The link with the highest weight should be selected as the cooperative MIMO link for inter-cluster communications. The selection of cooperative MIMO links between two clusters can be done at either end of the link, since each cluster head is aware of the information of all neighboring clusters and their adjacent boundary nodes after the cluster discovery phase. However, to avoid inconsistency, the cluster with the smaller ID of two neighboring clusters is designated to select the cooperative link between them. After selecting the cooperative MIMO link, the cluster head with the smaller ID sends out a Cooperative MIMO Link Request (CMIMOLR) message, which is acknowledged by a Cooperative MIMO Link ACK (CMIMOLA) message from the other neighboring cluster head. For the sake of simplicity, all CMIMO link negotiation messages are transmitted at power $P_{\text{high}}$, such that all neighboring cluster heads can be reached directly. The formats of the CMIMO negotiation messages are shown in Fig. 7.

After this phase, clusters have been formed and the cooperative MIMO links have been selected for each pair of neighboring clusters. In reality, collisions cannot be fully avoided by the CSMA/CA mechanism. Suboptimal cluster heads or suboptimal cooperative MIMO links may be selected if collision occurs during the above phases. On the other hand, the cluster formation procedure is executed periodically and optimal cluster heads and cooperative MIMO links are reselected every time the procedure is performed. Thus the effect of collisions in these phases on energy efficiency and network lifetime is negligible in a long run.

III. CHANNEL ASSIGNMENT

In MCCMIMO, both intra-cluster communications and inter-cluster communications are synchronized. Hence, neighboring clusters may interfere with each other in the intra-cluster communication phase, while nearby cooperative MIMO links may interfere with each other in the inter-cluster communication phase. To avoid the throughput degradation and low energy efficiency caused by interference, different channels should be assigned to neighboring clusters and interfering cooperative MIMO links. Unfortunately, the channel assignment problems for both clusters and cooperative MIMO links are NP-hard. This can be proved by reducing each cluster into a vertex and showing its equivalence to the general channel assignment problem in wireless networks, which has been proved to be NP-hard [18]. To provide practical solutions, we propose two distributed greedy algorithms, which are for cluster channel assignment and cooperative MIMO link channel assignment, respectively. During the channel assignment stage, all communications are on the default channel and all nodes access the channel using a CSMA/CA like mechanism with weighted backoff time to avoid collision. The same available channel set is used for both algorithms since intra-cluster communications and inter-cluster communications do not occur...
simultaneously. When cluster reformation is performed, the cluster channel assignment procedure is triggered, followed by the cooperative MIMO link channel assignment procedure. We describe these two algorithms in detail in the next two subsections.

A. Channel Assignment for Clusters

It is intuitive that the lifetime of WSNs can be extended by avoiding unnecessary energy expenditure in the clusters with low remaining energy. Motivated by this, in MCCMIMO, clusters with less residual energy are assigned with channels earlier so as to obtain less interfered channels. We define the remaining energy of a cluster as the weighted average of the remaining energy of all its cluster members. The cluster head and cluster members are given different weights since the cluster head consumes energy much faster. Let \( h \) be the cluster head of cluster \( I \), and \( C(I) \) be the set of cluster members of cluster \( I \). The remaining energy of cluster \( I \) is defined as follows

\[
En(I) = 0.5En(h) + 0.5\left( \sum_{i \in C(I)} En(i)/|C(I)| \right)
\]  

(3)

In MCCMIMO, a cluster head needs to know the remaining energy of all neighboring clusters so as to determine its order of channel assignment in a distributed manner. The remaining energy of all neighboring clusters can be piggybacked in the CMIMO ACK message during CMIMO link selection phase. If a cluster has the least remaining energy in its 1-hop neighborhood, the cluster head assigns the least used channel to itself and broadcasts a Channel Assignment (CA) message at power \( P_{high} \) to neighboring clusters. Otherwise, it keeps listening and updating its channel usage table until all neighboring clusters with less remaining energy have assigned their channels. The channel assignment for the clusters in Fig. 4 is illustrated in Fig. 8.

![Channel assignment for clusters](image)

B. Channel Assignment for Cooperative MIMO Links

Similar to channel assignment for clusters, cooperative MIMO links with less remaining energy should be assigned earlier to reduce their energy expenditure caused by collisions, thus prolong network lifetime. However, the energy efficiency of cooperative MIMO links heavily depends on the interference level. The efficiency drops dramatically as the interference level increases. By taking both these factors into consideration, the weight of cooperative MIMO links for channel assignment is defined by Equation (4). The weights of cooperative MIMO links are piggybacked in both the CMIMO Request and the CMIMO ACK messages such that all neighboring clusters at both ends of a cooperative MIMO link can obtain them via overhearing the CMIMO messages.

\[
Weight_{CA}(l) = En(l)/Ef(l)
\]  

(4)

The cooperative MIMO link channel assignment is implemented by a 3-way handshaking negotiation mechanism. The advantage of this mechanism is two folds: First, the least used channel at both ends of the link can be sensed and assigned. Second, the neighboring clusters at both ends of the link can update their channel usage tables. A timer is set for each link based on its link weight. Once the timer expires, the lower ID end of the link initiates the channel assignment procedure by transmitting a Channel Assignment Request (CA Request) message including its preferred channel list. After receiving the CA Request message, the other end of the link selects the least interfered channel based on the information in the CA Request message and its own channel usage table, and sends out a Channel Assignment Ack (CA ACK) message. The lower end cluster head then transmits a Channel Assignment Response (CA Response) message to acknowledge the CA ACK message. Each cluster head keeps listening and updating its channel usage table until all its adjacent cooperative MIMO links have been assigned a channel. We assume that the link weight value is unique such that no conflict occurs in this phase. If the weight value for each cooperative MIMO link is not unique, multiple channel negotiation messages may be transmitted at the same time and lost due to collisions. Consequently, a channel may be assigned to multiple interfering cooperative MIMO links. Nevertheless, since a collision avoidance scheme is used in the MAC for inter-cluster communications, the performance will not be noticeably affected in such a case as will be seen in the simulation results. The detail of this MAC will be illustrated in the next section.

An example of channel assignment for cooperative MIMO links of the WSN in Fig. 4 is shown in Fig. 9.

IV. MEDIUM ACCESS CONTROL

In this section, we introduce the media access control mechanism of MCCMIMO. As shown in Fig. 10, the time is divided into fixed length MAC intervals and each interval consists of four windows: 1) Sync Window, during which each node synchronizes its clock; 2) Cluster Members to Cluster Head Window, during which cluster members transmit messages to their cluster heads; 3) Cluster Head to Cluster Member Window, during which cluster heads transmit messages to cluster members and cooperative MIMO nodes; 4) Inter-Cluster Communication Window, during which inter-cluster messages are transmitted over the cooperative MIMO
links. The specifications of these windows are described in the following subsections. The detailed synchronization scheme is beyond the scope of this paper, as many algorithms have already been proposed in the literature.

![Fig. 9. The channel assignment for cooperative MIMO links of the WSN in Fig. 4. Red dots denote cluster heads, blue dots denote cluster members; the links between cluster heads and cluster members are represented by dotted lines, and cooperative MIMO links are denoted by solid lines. The colors of solid lines represent different channels assigned for cooperative MIMO links.](image)

**A. Cluster Members to Cluster Head Window**

At the beginning of this window, corresponding nodes of cooperative MIMO links set their channel to the cluster channel for intra-cluster communications. To avoid collision and overhearing energy consumption, TDMA is used to schedule data transmissions in this window. Since the number of time slots is set to the maximum cluster size in the network, the length of the window is fixed. This number can be approximately obtained, given the number of sensor nodes and the deployment field. As shown in Fig. 11, there is a short traffic indication window in each time slot at the beginning of the window. Each cluster member determines its traffic indication window by hashing its node ID and transmits a notification message to the cluster head if it holds pending traffic. After that, the cluster member sleeps until the beginning of the Cluster Head to Cluster Members Window if it has no packet to send; Otherwise, it only wakes up during its corresponding time slot and sends the packet to the cluster head.

![Fig. 10. The components of a MCCMIMO MAC interval.](image)

**B. Cluster Head to Cluster Members Window**

In this window, cluster heads transmit messages to its cluster members, including both intra-cluster packets and packets to be relayed to other clusters via cooperative MIMO links. It is unnecessary for all cluster members to be active during this window since only a few packets are for each of them. To avoid overhearing energy consumption, the cluster head broadcasts a traffic indication message indicating the time slot allocation for each cluster member at the beginning of this window. The number of time slots for each cluster member is determined by the traffic demand. After receiving the traffic indication message, the cluster member sleeps except for the time slots that it is scheduled to receive packets from the cluster head. The cluster head sends out packets in the order indicated in the traffic indication message. The structure of this window is shown in Fig. 12.

![Fig. 11. The Cluster Members to Cluster Head Window of MCCMIMO MAC intervals.](image)

**C. Inter-Cluster Communication Window**

At the beginning of this window, the corresponding nodes of cooperative MIMO links switch their channel to the channels assigned for inter-cluster communications. Given limited channel resources, interference may still exist among cooperative MIMO links even after channel assignment. Furthermore, collisions may occur when both ends of a CMIMO link try to transmit at the same time. Therefore, each end of a CMIMO link should sense the channel before transmitting a packet. However, the exponential back-off scheme of CSMA/CA mechanism is not suitable here, as the start time of the transmission is unpredictable, which makes it hard to fix the length of the Inter-Cluster Communication Window.

We design a semi-deterministic CSMA/CA mechanism to solve this problem. As shown in Fig. 13, the Inter-Cluster Communication Window is further divided into fixed-length sub-windows. Each sub-window consists of $CW_{fix}$ number of contention slots and a period that is long enough to transmit a maximum-length packet. At the beginning of each sub-window, each node sets up a timer with a random value ranging from zero to $CW_{fix}$ time slots and keeps listening to its assigned channel. It transmits a packet starting with a cooperative MIMO synchronization preamble when the timer expires. If a node senses a transmission before its own timer expires, it first checks whether the packet is sent out by the cooperative MIMO transmitter. If so, it transmits the packet simultaneously when the synchronization preamble is over;
Otherwise, it cancels its own transmission, turns to sleep mode or receives the packet being transmitted based on whether the packet is for it or not. A node goes to sleep after a successful transmission till the beginning of the next sub-window if the length of the packet is shorter than the maximum. A node without packets to transmit and not being a receiver would turn to sleep after sensing a transmission. At the beginning of the next sub-window, each node resets its timer to a random value to begin a new round of contention. The value of $CW_{fix}$ and the number of sub-windows are adjusted periodically during the cluster reformation phase based on traffic demand.

![Inter-Cluster Communication Window](image)

---

**V. ENERGY EFFICIENCY ANALYSIS**

In this section, we analyze the energy efficiency of MCCMIMO protocol. We assume that both intra-cluster and inter-cluster communications are over Rayleigh fading channels.

**A. Cluster Formation and Channel Assignment Overhead**

Under the conflict-free assumption in these stages, each node simply needs to transmit three packets to complete the cluster formation and at most $\delta$ packets for the cooperative MIMO link selection, where $\delta$ is the maximum cluster degree (i.e., the number of neighboring clusters). Every node needs one transmission at power level $P_{high}$ for the channel assignment of clusters and at most $(3/2)\delta$ transmissions at power level $P_{high}$ for the channel assignment of cooperative MIMO links.

Both the cluster formation and channel assignment algorithms in MCCMIMO are quasi-static, as we assume that the network topology remains stable and no dramatic energy consumption occurs thus it is unnecessary to perform these procedures for each MAC interval. Therefore, the energy consumption of these two stages is negligible compared with that of data transmissions.

**B. Energy Efficiency of Intra-Cluster Communication**

For intra-cluster communications (including the communications between cluster members via the cluster head and the communications between cluster members and the cluster head), the energy efficiency of MCCMIMO is at the same level as other cluster-based MAC protocols since only traditional SISO transmissions are involved. Nevertheless, MCCMIMO may perform better since the average distance between the cluster head and cluster members in MCCMIMO is shortened by considering the average distance to 1-hop neighbors during the cluster formation.

---

**C. Energy Efficiency of Inter-Cluster Communication**

As discussed in [12], [13], [14], significant energy efficiency improvement can be achieved in WSNs by using cooperative MIMO links. We formulate the energy efficiency model of MCCMIMO and compare its performance with other protocols in this subsection.

1) **MIMO Link Energy Model**: Following [12], the energy consumption of MIMO transmission in WSNs includes two parts: the power consumption of all power amplifiers $P_{PA}$ and the power consumption of other circuit blocks $P_C$. The amplifier power consumption can be approximated by the transmit power $P_{out}$ as in Equation (5), where $\alpha$ is determined by the modulation scheme and the constellation size.

$$P_{PA} = (1 + \alpha)P_{out} \quad (5)$$

The circuit power consumption can be represented by Equation (6), where $P_{DAC}$, $P_{ADC}$, $P_{fitr}$, $P_{filr}$, $P_{LNA}$, $P_{IFA}$, $P_{mix}$ and $P_{syn}$ stand for the power consumption values for the DCA, ADC, the active filters at transmitter side and receiver side, the Low Noise Amplifier (LNA), the Intermediate Frequency Amplifier (IFA), the mixer and the frequency synthesizer, respectively.

$$P_C \approx M_t(P_{DAC} + P_{mix} + P_{fitr}) + 2P_{syn}$$

$$+M_r(P_{LNA} + P_{mix} + P_{IFA} + P_{filr} + P_{ADC}) \quad (6)$$

Using Equation (5) and Equation (6), the total power consumption per bit for both the MIMO link and the referenced SISO link over a Rayleigh channel can be calculated by Equation (7), where $M_t$, $G_t$, $G_r$, $\lambda$, $R_b$, $N_f$, $d$, $M_t$, $N_0$ and $P_b$ are the link margin compensation, the transmitter antenna gain, the receiver antenna gain, the carrier length, the bit rate, the receiver noise figure, the transmission distance, the number oftransmitter, single-sided thermal noise and average BER, respectively. We can obtain the total energy consumption per bit for SISO, MISO and MIMO schemes by setting different $M_t$, $G_t(t)$ and $G_r(r)$ parameters accordingly. With other parameters prefixed, we denote the total energy consumption per bit for SISO, MISO and MIMO as a function of transmission distance $d$ by $E_{siso}(d)$, $E_{miso}(d)$ and $E_{mimo}(d)$, respectively.

$$E_{bt} = \frac{16M_tN_0\pi^2d^2}{P_b^{1/\lambda_t}G_tG_r\lambda^2}M_tN_f + P_e/R_b \quad (7)$$

2) **Inter-Cluster Communication Energy Model**: It takes more than one transmissions to communicate between two cluster members in different clusters. As illustrated in Fig. 14(a), to transmit a packet between cluster members A and B, which are located in neighboring clusters, five intermediate transmissions are required in MCCMIMO, which are: 1) From cluster member A to cluster head C via SISO transmission; 2) From cluster head C to cooperative MIMO transmitters via SISO transmission (broadcast); 3) Inter-cluster cooperative MIMO transmission; 4) From cooperative MIMO receivers to cluster head D via MISO transmission; 5) From cluster head D to cluster member B via SISO transmission. Accordingly, the expected energy consumption per bit for such a transmission can be represented by Equation (8), in which $d_{ij}$ denotes the expected distance between a cluster head and its cluster.
members, and $\bar{d}_{cc}$ denotes the expected distance between the cooperative MIMO transmitters and receivers. The expected energy consumption per bit for the cooperative MIMO protocol presented in [16], traditional multi-hop SISO clustered WSNs, and SISO clustered WSNs can be derived similarly as follows by using Fig. 14(b), (c) and (d).

$$E_{1-hop} = 3E_{\text{siso}}(\bar{d}_{ch}) + E_{\text{miso}}(\bar{d}_{ch}) + E_{\text{mimo}}(\bar{d}_{cc})$$ (8)

For communications between two cluster members separated by more than one cluster, we can derive the expected energy consumption per bit easily by extending Equation (8). This is because that three transmissions are always needed to relay a packet from one cluster head to the next cluster head. The expected energy consumption per bit for multi-hop communication is given in Equation (9), in which hops is the number of intermediate clusters.

$$E_{\text{hops}} = 2E_{\text{siso}}(\bar{d}_{ch}) + \text{hops} \cdot E_{\text{siso}}(\bar{d}_{ch}) + \text{hops}(E_{\text{miso}}(\bar{d}_{ch}) + E_{\text{mimo}}(\bar{d}_{cc}))$$ (9)

is because that the average intra-cluster distance becomes longer as more nodes are contained in each cluster. From Fig. 15(b), we can observe that when the average cluster size is 8, the energy cost of MCCMIMO drops as the node density increases, since both the distance from the cluster head to cluster members and the length of cooperative MIMO links are shortened. The superiority of MCCMIMO on energy efficiency over CMIMO becomes less evident as the node density increases, since the advantage of MIMO over SISO shrinks when the transmission distance becomes shorter.

We also evaluate the energy efficiency of MCCMIMO and compare it with existing schemes for communications between cluster members separated by more than one cluster. Randomly generated 150-node WSNs over a $250m \times 250m$ field are used to determine the expected intra-cluster distance and the length of cooperative MIMO link. The average cluster size is set to 8. From Fig. 15(c), we can see that MCCMIMO always outperforms other schemes in terms of energy efficiency, and such superiority becomes more evident as the number of hops increases.

VI. PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of the MCCMIMO protocol through simulations and compare it with the CMIMO protocol proposed in [16]. We implemented both protocols in NS-2 simulator and conducted three sets of experiments that study the all-clusters-to-sink traffic scenario, the peer-to-peer inter-cluster traffic scenario, and the impact of varying system parameters, respectively.

In the simulations, 150 sensor nodes are randomly scattered over a $250m \times 250m$ field. The average cluster size of both MCCMIMO and CMIMO is around 8 by setting the maximum intra-cluster transmission range to $50m$. The maximum inter-cluster transmission range is set to $40m$ for MCCMIMO and $80m$ for CMIMO to ensure the connectivity of the network. It is notable that the maximum inter-cluster transmission range for MCCMIMO is much smaller than that of CMIMO, because inter-cluster communications for MCCMIMO are among cooperative MIMO nodes at the boundary of clusters, while inter-cluster communications for CMIMO are among cluster heads, which are typically located at the center of clusters. The link bandwidth is set to $2Mbps$ and the maximum packet size is set to $100bytes$. 15 time slots are allocated for the Cluster Members to Cluster Head Window and each time slot lasts $405\mu s$. The lengths of the Cluster Head to Cluster Members Window and the Inter-Cluster Communication Window are set to $6ms$ and $8ms$, respectively. The number of available channels is fixed to 5 and the maximum contention window size $CW_{\text{fix}}$ is set to 32 unless otherwise stated.

A. All-Clusters-to-Sink Traffic

In this set of experiment, we compare MCCMIMO with CMIMO in the scenario where the traffic is from cluster members to a sink node located inside a cluster. To study the impact of the position of the sink node, we place the sink node either in a randomly selected cluster or in the cluster with the maximum degree. At most 1 CBR flow is generated in each cluster. The simulation results are plotted in Fig. 16, where
that MCCMIMO can achieve over CMIMO. From Fig. 17(a), we observe while MCCMIMO-2 represents the cases of CMIMO in terms of system throughput and end-to-end delay under the peer-to-peer inter-cluster traffic. The number of communicating cluster pairs is varied from 1 to 10, and the number of CBR flows between each communicating cluster pair is evaluated respectively. The simulation results are shown in Fig. 17, where MCCMIMO-1 stands for the 1 CBR flow case, while MCCMIMO-2 represents the 2 CBR flows case. Similar notation are used for CMIMO. From Fig. 17(a), we observe that MCCMIMO can achieve over 150 Kbps throughput for 2 CBR flows, which corresponds to more than 5 times performance gain over CMIMO. We also see that as the CBR flows between each cluster pair increase from 1 to 2, MCCMIMO achieves higher throughput, while CMIMO has roughly the same throughput. This is because that the cooperative MIMO links are already saturated in the 1 CBR flow case due to severe co-channel interference in CMIMO. The average end-to-end delay of MCCMIMO and CMIMO is shown in Fig. 17(b). The trends are similar to the all-clusters-to-sink traffic scenario and the conclusion applies as well. Nevertheless, it is noticeable that the delay for MCCMIMO under maximum traffic load is shorter compared to the previous scenario, which is reasonable because the inter-cluster traffic loads are distributed over all cooperative MIMO links in this scenario, while converged to the cooperative MIMO links near the sink node in the previous scenario.

B. Peer-to-Peer Inter-Cluster Traffic

In this set of experiment, we compare MCCMIMO with CMIMO in terms of system throughput and end-to-end delay under the peer-to-peer inter-cluster traffic. The number of communicating cluster pairs is varied from 1 to 10, and the cases of 1 CBR flow between each communicating cluster pair and 2 CBR flows between each communicating cluster pair are evaluated respectively. The simulation results are shown in Fig. 17, where MCCMIMO-1 stands for the 1 CBR flow case, while MCCMIMO-2 represents the 2 CBR flows case. Similar notation are used for CMIMO. From Fig. 17(a), we observe that MCCMIMO can achieve over 150 Kbps throughput for 2
the network separately. First, the number of available channels is varied between 1 and 11. The system throughput is shown in Fig. 18, in which R-sink stands for the situation where the sink is randomly placed, M-sink is the case that the sink is placed in the maximum-degree cluster, and P2P represents the peer-to-peer traffic scenario. We can see that in all-clusters-to-sink traffic scenarios, the system throughput is higher if assigned more channel resources, which leads to less co-channel interference; However, the throughput stops increasing if more than 9 channels are provisioned, as interference no longer exists after that. Second, the maximum contention window size is changed from 2 to 64 and the simulation result are plotted in Fig. 18(b). It is clear that there is a trade-off between the system throughput and the maximum contention window size. This is because that collisions are avoided with the increase of the contention window size at first; however, the channel capacity wasted by the large contention window surpasses the capacity gain from collision avoidance when the contention window size further increases. As a result, the throughput starts to decline. From the figure, we also notice that different traffic scenarios have different optimal sizes for the maximum contention window.

Fig. 17. Simulation results for peer-to-peer inter-cluster traffic scenario. (a) System throughput vs. Number of CBR flows; (b) End-to-end delay vs. Number of CBR flows.

Fig. 18. System throughput for varying system parameters. (a) System throughput vs. Number of available channels; (b) System throughput vs. Maximum contention window size.

VII. Conclusions

In this paper, we have designed an energy-efficient, high-throughput MAC protocol, called MCCMIMO, which utilizes both multiple channels and cooperative MIMO. In MCCMIMO, sensor nodes are organized into clusters using a new metric and different cooperative MIMO links are selected for communications among neighboring clusters. To reduce collisions and interferences, different channels are assigned to neighboring clusters and interfering cooperative MIMO links, respectively. We also proposed a medium access control mechanism, which employs TDMA scheme for intra-cluster communication to avoid overhearing and synchronized CSMA/CA for inter-cluster communications to reduce collisions. We also gave numerical analysis and extensive simulation results which demonstrate that MCCMIMO can achieve significant improvement on both energy efficiency and system throughput compared to other schemes.

ACKNOWLEDGMENTS

This research work was supported in part by the U.S. National Science Foundation under grant number ECCS-0801438, and U.S. Army Research Office under grant number W911NF-09-1-0154.

REFERENCES


20