AFRL-RY-WP-TP-2010-1152

MULTIPLE-OPTIMIZING DYNAMIC SENSOR NETWORKS WITH MIMO TECHNOLOGY (PREPRINT)

Wei Chen, Miao He, and Fletch Gregory

Tennessee State University

JUNE 2010

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
June 2010	Technical Paper Preprint	08 September 2006 – 31 August 2009

4. TITLE AND SUBTITLE
MULTIPLE-OPTIMIZING DYNAMIC SENSOR NETWORKS WITH MIMO TECHNOLOGY (PREPRINT)

5a. CONTRACT NUMBER
FA8650-05-D-1912-0007

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62204F

6. AUTHOR(S)
Wei Chen, Miao He, and Fletch Gregory

5d. PROJECT NUMBER
7622

5e. TASK NUMBER
11

5f. WORK UNIT NUMBER
7622110P

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Tennessee State University

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYRR

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TP-2010-1152

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Technical paper for general release. PAO Case Number: WPAFB 08-0030; Clearance Date: 08 Jan 2008. Paper contains color.

This work was funded in whole or in part by Department of the Air Force contract FA8650-05-D-1912-0007. The U.S. Government has for itself and others acting on its behalf a paid-up, nonexclusive, irrevocable worldwide license to use, modify, reproduce, release, perform, display, or disclose the work by or on behalf of the U.S. Government.

14. ABSTRACT

A Multiple-Input Multiple-Output (MIMO) transceiver provides extremely high spectral efficiencies by simultaneously transmitting multiple data streams in the same channel. Some research works have used MIMO technology to increase data rate or reduce energy consumption at communication links in wireless sensor networks. Since all layers of the protocol stack affect network performance, an efficient system should exploit the advantage of MIMO technology across all layers as well as the underlying hardware where the resource constraints actually come from. In this paper, we show a joint design of MIMO aware network architecture, routing and MAC protocols underlying physical hardware. Our network architecture is reconfigurable with two functions: node-joining and node-leaving. We show that our design can fully exploit the advantage of MIMO technology and multiple-optimize network throughput, network lifetime and network reconfiguration cost in a dynamic sensor network.

15. SUBJECT TERMS
MIMO, data rate, sensor network

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Nivia Colon-Diaz
Unclassified	Unclassified	Unclassified	SAR	18	19b. TELEPHONE NUMBER *(Include Area Code)* N/A

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Multiple-optimizing Dynamic Sensor Networks with MIMO Technology

Wei Chen, Miao He, Fletch Gregory
Tennessee State University, USA
(wchen@tnstate.edu, hmiao@tnstate.edu, havokjinx@gmial.com)

Abstract: A Multiple-Input Multiple-Output (MIMO) transceiver provides extremely high spectral efficiencies by simultaneously transmitting multiple data streams in the same channel. Some research works have used MIMO technology to increase data rate or reduce energy consumption at communication links in wireless sensor networks. Since all layers of the protocol stack affect network performance, an efficient system should exploit the advantage of MIMO technology across all layers as well as the underlying hardware where the resource constraints actually come from. In this paper, we show a joint design of MIMO aware network architecture, routing and MAC protocols underlying physical hardware. Our network architecture is reconfigurable with two functions: node-joining and node-leaving. We show that our design can fully exploit the advantage of MIMO technology and multiple-optimize network throughput, network lifetime and network reconfiguration cost in a dynamic sensor network.

1. Introduction

We consider a wireless sensor network (WSN) that consists of a large number of sensor nodes deployed in an unattended field, where each node has a sensor array, a controlling processor, a radio transceiver system and some batteries. In a WSN, a node operates not only as an end-system, but also as a router to process and forward application data. Since sensor nodes have severe resource constraints, network performance can be unstable and poor. Network reconfiguration is crucial in a dynamic WSN since such a WSN can change its geographical topology often. For instance, a sensor node leaves the network when its battery voltage is low and comes back when it is recharged, and if it is a mobile node it can move into or move out from the network at any time.

A hierarchically organized sensor network offers a better networking performance. Clustering has been used to induce a hierarchical structure over a flat WSN which minimizes communication overhead, facilitates energy efficient sensing and networking operation, and facilitates network self-reconfiguration. In many proposed approaches, cluster head nodes are selected through finding a small dominating set. It is known that finding a minimum dominating set (MDS) is an *NP*-complete problem. Several reconfigurable cluster-based architectures have been proposed. In [9], reconfigurable overlays is designed for multi-scale communication in a WSN; and their approaches depend on special MAC protocols proposed in [1, 10]. A dynamic WSN is a reconfigurable network which can support node-joining and node-leaving. A dynamic cluster-based sensor network is proposed in [16], where the architecture supports operations of node-joining and node-leaving and number of clusters in the architecture is less than 5×|MDS| when the network is a disk unit graph. In their protocol, a node-joining operation can be executed in local one-hop area; however, a node-move-out operation may cause a complete reconstruction of the architecture, which is a common problem for other cluster-based architectures. Furthermore, when all nodes in a one-hop area die or move out, the network will be unconnected and totally lose its functionality. Some literatures consider head-rotation technique for reducing the workload at head nodes [5, 6, 11]. However, the technique has the same problem; it does not work if all nodes in the local one-hop area died.

In this paper, we assume that each node is equipped with a MIMO transceiver. A MIMO transceiver employs digital adaptive transmitting and receiving antenna arrays and provides extremely high spectral efficiencies by simultaneously transmitting multiple data streams in the same channel. A MIMO link employs MIMO transceivers at both ends of the link. Such a link can provide two type of gain without using too much extra energy: *diversity gain* and *spatial multiplexing gain*. Diversity gains primarily provide range extension, while spatial multiplexing gain primarily provides higher data rate. The maximum diversity order afforded by a MIMO link with *M* transmit antennas and *N* receive antennas is

MN. Let $M = N$, the capacity grows linearly with M from spatial multiplexing gain. Theoretically, both gains can be obtained in the same channel without using. Practically, the transceiver circuit with more transmitter/receiver antennas consumes extra energy, and the encoding/decoding in multiple transmitters/receivers consumes extra energy too. Several research groups have worked on energy efficient MIMO transceivers. They embedded energy management into the transceiver to minimize the energy consumption in communication links [3, 4]. In [18], the authors investigated the trade-off of diversity gain and multiplexing gain in a MIMO transceiver and proposed the point of view that both types of gains can be simultaneously obtained. Some groups have worked on improving MIMO WSNs throughput by joint design of MIMO transceiver with MAC protocol [16, 17]. Recently, K. Sundaresan and his group have worked on design of a unified MAC framework with MIMO technology for improving network throughput [12, 14]. They have also introduced the mechanism of MIMO links in to routing protocols, where the diversity gain is used for extending communication range in routing, and multiplexing gain is used for increasing link date rate [13]. In their works, they have not considered energy and network life time issues. The part of the reason is that in their design there is no physical model that reflects the trade-off of energy, data rate, transmission distance, types of MIMO transceiver and other physical parameters in hardware.

Since all layers of the protocol stack affect the network performance, in order to fully exploit the potential in MIMO technology, in this paper, we consider a joint design across all layers as well as the underlying hardware where the resource constraints actually come from. Our design multiple-optimizes network throughput, network lifetime and network reconfiguration cost in a dynamic WSN. First, we propose a special cluster-based network architecture that can fully exploit diversity gain of MIMO transceivers with a number of nice properties: the architecture has a small number of clusters and enables efficient routing, it supports time and energy efficient self-reconfiguration in which both node-joining and node-leaving operations can be executed in local one-hop area, and it induces head rotating and link jumping which can maximize network lifetime. Head rotation is used for balancing the workload at nodes and link jumping is used for extending transmission distance when some nodes in the route dies. Then, we build a physical model that reflects the relation of energy, data rate, transmission distance, types of MIMO transceivers, and other parameter in hardware. We also propose MIMO aware MAC protocol and routing protocols. Our MAC protocol works as a local optimizer. By optimally selecting diversity gain and/or multiplexing gain based on the trade-off provided by the physical model, it can simultaneously minimize energy consumption and maximize data rate at communication links. Furthermore, the MAC protocol enables our routing algorithms perfectly matching the proposed cluster-based architecture, and therefore, leverages the advantage of MIMO technology from local links to the global network.

In order to evaluate the performance of the architecture, MAC protocols and routing algorithms, we developed a software simulator. After comparing the simulation results in SISO (Single Input Single Output), 2×2 MIMO and 4×4 MIMO WSNs for different filed size, communication range, number of sensor nodes, battery energy and we conclude that our joint design can fully exploit the advantage of MIMO technology and significantly improve the network throughput, network lifetime and network reconfiguration cost. We also compare the performance of our cluster-based network architecture with that of flat network architecture, and conclude that the proposed cluster-based architecture plays a crucial role in our joint design.

2. MIMO Technology and Its Physical Model

A MIMO transceiver employs digital adaptive transmitting and receiving antenna arrays. It can be used for increasing the amount of diversity or the number of degrees of freedom in wireless communication systems. Traditionally, multiple antennas have been used to increase diversity to combat channel fading. Each pair of transmitter and receive antennas provides a signal path from the transmitter to the receiver. By sending signals that carry *the same information streams* through different paths, multiple independently faded replicas of data symbol can be obtained at the receiver end; hence, more reliable reception is achieved. The diversity gain can be used to provide range extension or reduce error

rate. On other hand, if the path gains between individual transmit-receive antenna pairs fade independently, the channel matrix is well conditioned with high probability, in which case multiple parallel spatial channels are created. By transmitting *independent information streams* in parallel through the spatial channels, the data rate can be incurred. This effect is called as special multiplexing gain. The maximum diversity order afforded by an $M \times N$ MIMO link with M transmit antennas and N receive antennas is MN. Let $M = N$, the capacity grows linearly with M from spatial multiplexing gain [11, 12, 13]. In Fig.1, for example, assuming that the route from node *a* to node *d* is *abcd*, node *a* can use spatial multiplexing gain to increase data rate in link *ab*. On the other hand, when node *b* runs out of the battery or moves out, node *a* can use diversity gain to extend its communication range to node *c*. According to [18], a MIMO transceiver can achieve two types of gain at same time. For example, we can divide an 8×8 MIMO link into four 2×2 MIMO virtual links: each 2×2 MIMO link transmits coherent data signal for getting diversity gain and different virtual links transmit independent data signal for getting multiplexing gain.

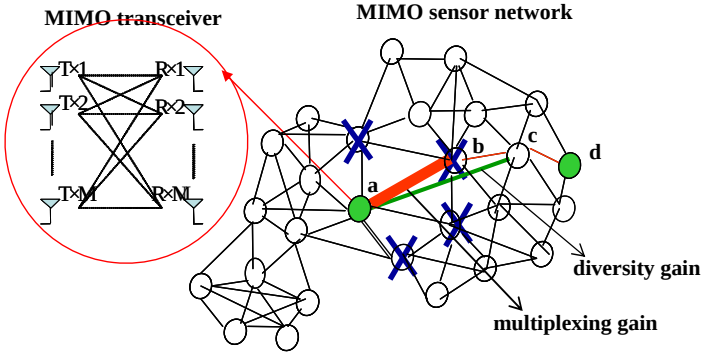

Fig. 1 MIMO transceiver and its gain: diversity gain and multiplexing gain

Theoretically, diversity gain and multiplexing gain can be obtained in the same channel without using extra energy. However, the transceiver circuit with more transmitter/receiver antennas consumes more energy, and the encoding/decoding the transceiver consumes energy too. In this paper, we assume that each node in a WSN is equipped with an $M \times M$ MIMO transceiver. We use the same assumption in [3, 4] that energy for encoding/decoding is small enough and can be ignored. The total energy consumption per bit, denoted as E_{bt}, for a fixed-rate system can be estimated approximately as follows:

$$E_b = (P_{PA} + P_c)/R_b \quad \text{---------- (a)}$$

where P_{PA} is the power consumption dependent on the transmit power, P_c is the power consumption dependent on the transceiver circuit, respectively. Furthermore, we can explain P_{PA} and P_c as follows:

$$P_{PA} = (1+\alpha)\frac{MR_b N_0}{p_b^{1/M}} \times \frac{(4\pi D)^2}{G_t G_r \lambda} M_l N_f \quad \text{---------- (a1)}$$

$$P_c = \phi M + \varphi \quad \text{---------- (a2)}$$

where R_b, D, M, p_b are the bit rate, transmission distance, number of transmit-antennas and number of receivers in the MIMO transceiver, and average bit error rate, respectively. Others

are system constants. Furthermore, the spatial multiplexing gain obtained from a $M \times M$ link can be explained as follows:

$R_b^M = \varepsilon M R_b$ ---------- (b)

where R_b is the bit rate of a single antenna, R_b^M is the bit rate of a $M \times M$ MIMO link when all antennas are used for spatial multiplexing gain, and $\varepsilon \leq 1$ is a system constant that can be 1 if the multiplexing fading on the different antenna elements is completely uncorrelated. The physical model will be used in MAC and routing protocol for optimizing the network performance and network lifetime.

3. MIMO Aware Cluster-based Network Architecture

We consider a MIMO WSN of each node equipped with an $M \times M$ MIMO transceiver operating a single channel. A MIMO WSN can be represented by an undirected graph $G = (V, E)$, where V is set of sensor nodes, and nodes u and v have an edge between them *iff* they are in their transmission distance with each other. Since G contains no hierarchical structure, G is called as a *flat* graph. We assume that G is connected.

3.1 Primary Clustering and General Clustering

In this section, we define a cluster-based hierarchical structure on G. Since a node can extend its transmission distance by using diversity gain, we call the transmission distance before any extension to be *primary transmission distance*, denoted as d. We first define a *primary clustering* for G. The primary clustering is defined by clusters and a backbone which connects the clusters. The nodes of G are partitioned into node-disjoint clusters. Each cluster is a complete subgraph of G (there is an edge between any two nodes) (Fig.1 (a)).There is a *head* in each cluster and other nodes are *members*. The edges from the head to members are called as *cluster-edges*. The transmission distance for a cluster edge is always d and never changes. A backbone is a rooted tree formed by cluster heads. The transmission distance for a *backbone-edge* between two heads is $2d$ (Fig. (b)).

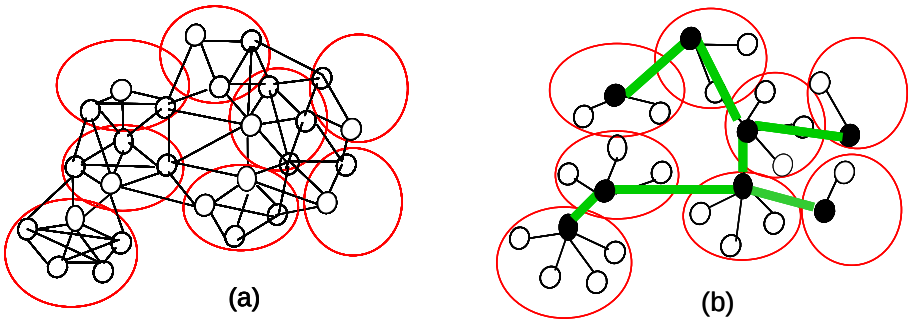

Fig. 2 (a) A primary clustering: each cluster is a complete subgraph of G; (b) A primary cluster-based structure formed by cluster-edges (thin ones) with transmission distance d and backbone-edges (thick ones) with transmission distance $2d$

The definition of a general clustering on G is the same as that of a primary clustering except that the transmission distances for backbone-edges can be different up to $2^h d$ $(h \geq 1)$.

Definition 1

(1) A cluster-based structure of a flat graph $G = (V, E)$ is a tree, denoted as $CNet(G) = (V, E_{CNet})$, where E_{CNT} consists of cluster-edges and backbone edges. In $CNet(G)$, the cluster-edges have a

same primitive transmission distance *d*, and the backbone edges can have different transmission distance up to $2^h d (h \geq 1)$.

(2) The subtree of *CNet*(*G*) formed by cluster heads and backbone edges is called as the backbone tree of *G*, denoted as *BT*(*G*).

The tree in Fig 2 (b) forms from cluster edges (black) and backbone edges (green). It is a cluster-based structure of the graph in Fig 2(a). The backbone tree consists of the cluster heads and the backbone edges (green).

In our cluster-based structure, we assume that each node *v* has a unique ID, *v.id*, and it keeps the following one-hop information:

(1) node *v* knows its status as a cluster member or a cluster head, denoted as *v.status*;
(2) if *v* is a cluster member, it knows its cluster head's ID, denoted as *v.head*; and
(3) if *v* is a cluster head
(i) it knows a list of its cluster members, denoted as *v.members* which contains members ID and their physical constraints such as batter energy level, and
(ii) it knows a list of its backbone neighbors, and transmission distance *d*(*v,w*) from *v* to each neighbor *w*.

3.1 WSN model

We have the following assumptions for our MIMO sensor networks:

1. Sensor nodes in the network work repeatedly in rounds. One round consists of one transmission or one reception, and some local computation.
2. When a node receives information signals from more than one node, collision happens and the node can not get any information.
3. When the nodes are deployed in a field, each node knows only its ID, i.e., they do not know any network knowledge such as neighbors and number of nodes.

Let G_0 be a flat sensor network formed by deploying *n* sensor nodes randomly and each node of G_0 knows only its ID. A *CNet*(G_0) can be self-initialized by using two ways: (1) one node invokes a gossip, all nodes get whole network knowledge of G_0 when the gossip completed,, and then each node uses *node-joining* operation (will be described in Section 3.2) to add the nodes one by one from G_0 to *CNet*(G_0); and (2) the nodes of G_0 move into *CNet*(G_0) one by one by using *node-joining* operation. A gossip be completed in $O(n)$ rounds [2]. The construction of *CNet*(G_0) at each node is a local computation. Therefore, the first one can be completed in $O(n)$ rounds. The second one can be completed in *n*×(rounds needed for one node-joining operation) and it is $O(\delta n)$ according to Section 3.2, where δ is the largest degree of the nodes in G_0.

3.2 Head-Rotation and Link-Jumping

Operations *head-rotation* and *link-jumping* are designed to maximize network lifetime. Head-rotation is used for evenly using nodes in a cluster, and link-jumping is used for overcoming the problem when a cluster dies which causes the network unconnected.

Head-Rotation Algorithm (*u*: head, *e*: energy threshold**)**

1. *u* checks its battery energy;
2. ***if*** *u*'s energy level is smaller than *e* ***then***
if there is no member whose energy lever is larger than *e* then *u* executes *link-jumping* operation;
else

u selects a member *v* from its cluster with the largest energy level;
u sends a *head-rotation* request and its member list to *v* with primary transmission distance *d;*
when *v* received the request from *u*, *v* change its status to be head, and delete *u* from its member list.

Link-Jumping Algorithms (*u*: head**)**

1. *u* sends a *link-jumping* request with the following knowledge: a list of *u*'s backbone neighbors, and transmission distance $d(u,w)$ from *u* to each backbone neighbor *w*.
2. When *u*'s backbone neighbor *v* receives the request with the knowledge, *v* deletes *u* from its backbone neighbor list, adds each *u*'s backbone neighbor, say *w*, into its backbone neighbor list, and change the transmission distance from *v* and *w* to be $d(v,w) = d(v,u)+d(u,w)$.

Theorem 1 The operation of head-rotation and link-jumping operation can be completed in $O(1)$ rounds.

Proof: In both operations, *u* transmits a request with knowledge once only, and *u*'s cluster members or its backbone neighbors correct their local knowledge when they receive the request. Therefore, only one round is needed.

3.2 Node-Joining and Node-Leaving

In this section, we show that our cluster-based architecture supports an efficient dynamic WSN. We show that node-joining and node-leaving operations can be completed efficiently.

3.2.1 Node-Joining Algorithm

Let $G = (V, E)$ be a flat WSN, and $CNet(G) = (V, E_{CNet})$ be *G*'s cluster-based WSN. We assume that that a node *new* wants to join *G*, and there is at least one node in *G* whose is in *new*'s primary transmission range *d*. The following algorithm reconfigures the cluster-based structure when *new* joins into *G*.

Node-Joining Algorithm(*new***)**

1. *new* runs a numbering algorithm to give an order to all its neighbors with primary transmission distance *d* (when it step finishes, *new* has a list of numbered neighbors in *G*);
2. *new* sends a request "I want to join" with a list of the neighbor and their order using primary transmission distance *d*;
3. When *new*'s neighbors receive the request, they send the knowledge they holds back to *new* one by one in turn according to their order;
4. ***if*** *new* can find a head *u* in its neighbors and all *u*'s members are *new*'s neighbors (in this case, *new* and *u*'s members form a complete graph), ***then***
   new sets *u* to be its head;
   new sends a request "join *u*'s cluster" to *u* with primary transmission distance *d*;
   when *u* received the request, *u* adds *new* into its member list;

else

new sets itself to be a head of new cluster and it selects a neighbors *w* with the smallest *id*.
new sends a request "tell me *w*'s head" with transmission distance 2*d*;
when *w*'s parent *p* received *new*'s request, *p* adds *new* into its backbone list and sets $d(p, new) = 2d$;
p sends the kowledge it holds to *new* with transmission distance 2*d*;
new sets *p* to be its backbone parent and sets $d(new, p) = 2d$.

Theorem 2 Give graph *G* and *CNet*(*G*), a *node-joining* operation can be completed in $O(\delta_{new})$, where δ_{new} is the number of *new*'s neighbors in *G*.

Proof: In Step 2, the numbering algorithm can be completed in $O(\delta_{new})$ rounds [7]. In Step 3, δ_{new} rounds are needed for each neighbor sending its knowledge to *new* in turn. Other part of the algorithm runs at most 3 rounds.

3.2.2 Node-leaving Algorithm

Let $G = (V, E)$ be a flat WSN, and $CNet(G) = (V, E_{CNet})$ be G's cluster-based WSN. We assume that that a node *old* wants to leave G, and G is still connected after *old* left. The following algorithm reconfigures the cluster-based structure when *old* leaves G.

Node-Leaving Algorithm

1. ***if*** *old* is a member node, ***then***
   old sends a request " member *old* is leaving" with primary transmission distance *d*;
   when *old*'s head receives the request, it deletes *old* from its member list;
2. ***else***
   old invokes *link-jumping* operation.

Theorem 3 Give graph G and $CNet(G)$, a node-leaving operation can be completed in $O(1)$ rounds.

Proof: Step 1 needs only 1 round. Step 2 invokes a link-jumping operation which can be completed in 1 round too.

4. MAC Protocol and Routing Algorithms

4.1 MAC Protocol as Local Multiple-Optimizer

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) has been considered as a default MAC protocol for ad-hoc network environments. A simple extension of CSMA/CA for a $k \times k$ MIMO link, denoted as CSMA/CA(k), can be realized that can provide a k fold improvement in throughput performance through spatial multiplexing gain comparing to a pure omni-directional environment []. According to formula (b) in Section 2, the bit rate gain from a $k \times k$ MIMO link with the CSMA/CA(k) MAC Protocols is $R_b^k = \varepsilon k R_b$, where $\varepsilon \le 1$ is a system constant. CAMA/CA(k) allows only fixed transmission to take place in a give time slot, it can not use MIMO links in optimal way.

In literature [13, 14], new CAMA/CA(k)-based MAC protocols were proposed which can achieve better network throughput by controlling the data streams. We want to have a MAC protocol that can improve not only network throughput but also network lifetime. Our MAC protocol is a CAMA/CA(k) embedded with a special multiple-optimizer, denoted as MultiOptimal-MAC. The idea in MultiOptimal-MAC is simple: according to [18] both types of gains can be simultaneously obtained; therefore, the antenna arrays at a transceiver are divided into s group such that each group has t (= k/s) antennas for the transmitter and t antennas for the receiver. Each group is used as a $t \times t$ MIMO transceiver to transmit coherent data streams for obtaining diversity gain, and the s groups are used to transmit independent data streams to obtain multiplex gain. As we see that in the formula (a) in Section 2, there are trade-offs among bit consuming energy, bit rate, transmission distance, error rate, and type of MIMO transceiver. The value of s and t can be decided by the current service requirement, network topology and usable resource for different multi-optimization purposes.

Recall that in our cluster-based WSN $CNet(G)$, the edges have transmission distance $2^h d$, where $h = 0$ for cluster edges and $h \ge 1$ for backbone edges. Our MultiOptimal-MAC assigns antenna channels based on the transmission distance of the edge (link): a $M \times M$ MIMO link is divided into $t = \frac{M}{2^h}$ groups and each group is a $2^h \times 2^h$ MIMO link. In each group the antennas transmit coherent data stream for getting diversity gain. Different groups transmit independent data stream for getting multiplex gain. According

to the physical model in Section 2, comparing with a SISO link, the MultiOptimal-MAC can raise the data rate about *t* times and extend transmission distance 2^h times without using too much extra energy.

4.2 Routing Algorithms for Maximizing Network Lifetime

Let $G = (V, E)$ be a WSN with $M \times M$ MIMO links. We show time and energy efficient routing algorithms for broadcast, data gathering and one-to-one communication in G on the proposed cluster-based structure $CNet(G)$. In order to maximize the network lifetime, the routing algorithms use *head-rotation* operation for balancing node workload, and use *link-jumping* operation for reconnecting the backbone tree when the nodes in the tree are out of energy. In the routing algorithm, a node u in $BT(G)$ does *reconfigurable transmission* using MO-MIMO MAC protocol as follows:

(1) If u't battery is lower than threshold than u invokes head-rotation. In the procedure of head-rotation, if there is no node whose energy is larger than the threshold, u invokes link-jumping to reconfigure backbone.

(2) According to MultiOptimal MAC Protocol, u assigns the antenna array based on its transmission distance to get multiple-optimization of energy consumption and bit rate.

The routing algorithms are based on *depth-first-search* called as *Eulerian tour* in the backbone tree $BT(G)$. In general, given an undirected tree T, by replace each edge of T with two opposite directed edges, an *Eulerain* tour of a message m traveling from a source node s to all other nodes in a tree T can be described as a procedure $Eulerian(s, T, m)$ as follows.

Procedure *Eulerian*(*T, s, m*)

Let v (at the beginning s) be the node with a token for relaying m. First, v selects a node u from v's neighbors to whom v has not send m yet, and then transmits m and u's id with transmission distance $d(v, u)$. When u received m with u's id, it got the token and it will relay m at next round. Other neighbors of v will discard the message when they received it. If v has already transmitted m to all its neighbors, v will pass the token to its parent p (i.e., it transmits m and p's id) with the transmission distance $d(v, p)$, where p is the node v received m first from. This procedure will repeat until the token turns back to source node s.

In the tour, each node transmits m exactly the times of its degree in T. In other words, m is relayed on each edge of T exactly twice. Therefore, the $Eulerian(T, s, m)$ can be completed in $2\times|\mathrm{T}|$ rounds without collision, where |T| is the number of edges in T.

In the following algorithm, source node s broadcasts a message m on G by using $BT(G)$.

Algorithm Broadcast(*BT*(*G*), *s, m***)**

If s is a head then it sets u to be s, otherwise sets s's head to be u. s sends message m to u with transmission distance d, and u invokes the procedure $Eulerian(BT(G), u, m)$. When the procedure finishes, all nodes of G receive m since $BT(G)$ consists of all cluster heads and when the node of $BT(G)$ transmits m on the *Eulerian* tour in turn one by one, their member receive m from their head without collision.

Algorithm $Eulerian(BT(G), s, m)$ can be completed in $2|BT(G)|$ rounds that is less than the twice of the number of clusters. Since the number of clusters is much smaller than the number of nodes in G, the broadcast can be completed very fast.

In the following algorithm, source node s collects required data from all node in G by using $CNet(G)$.

Algorithm DataGathering(*CNet*(*G*), *s***)**

If s is head it sets u to be s, otherwise sets s's head to be u. s sends message m to u with transmission distance d, and u invokes the procedure $Eulerian(CNet(G), u, m)$, where m is the message for collecting

required type of data. In the *Eulerian* tour, each node receives data from the previous node in the tour, adds its data together, and then transmits it to the next node. Since the data size will be larger and larger, if the data has *g* packets, then the node needs *g* rounds to transmit the data. When the procedure finishes, node *u* receives all required data. If u is not the source node, *u* sends the data with transmission distance *d* to its member *s*.

Algorithm **DataGathering(***CNet*(*G*), *s***)** can be completed in 2*b*|*CNet*(*G*)| rounds, where *b* is the number of packets for the required data, and |*CNet*(*G*)| is the number of edges of in *CNet*(*G*) that is (*n*-1). Therefore, the algorithm can be completed in *O*(*bn*) rounds.

The following algorithm finds a route for the communication between nodes *s* and *v*.

Algorithm RouteBetweenTwoNodes (*BT*(*G*), *s*, *v***)**

If *s* is a head it sets *u* to be *s*, otherwise it sets *u* to be *s*'head. *s* sends message *m* to *u* with transmission distance *d*, and *u* checks if *v* is its member. If it is, the route between *s* and *v* is (*s*, *u*, *v*). Otherwise, *u* invokes a broadcast *Eulerian*(*BT(G), u, m*), where *m* is message "finding a route from *u* to *v*". In the broadcast tour, when a node *w* in *BT*(*G*) is *v* or it has a member who is *v*, *w* stops the broadcast procedure and sends a message "set the route from *u* to *v*" back to its parent *x* in the tour. When *x* received the massage, it sets *w* to be its next node in the route from *u* to *v*, and then *x* sends the setting route message back to its parent *y*. When *y* received the massage, *y* will do the same thing to set up *x* to be its next node in the route. The route from *u* to *v* is completely set up when the route setting massage relays back to *u*. If *u* is not *s*, then *u* sends the message with transmission distance *d* back to *s*, and *s* sets *u* to be its next node in the route from *s* to *v*.

It is easy to see that algorithm RouteBetweenTwoNodes**(***BT*(*G*), *s*, *v***)** builds a route between *s*, *v* in O(|*BT*(*G*)|) rounds and the route is the shortest path from s to *v* in *BT*(*G*).

5. Performance Evaluation

In order to evaluate the average performance of the protocols, we use our software simulator to test them on the proposed cluster-based WSN in the field of $210m \times 210m$. The number of nodes for testing varies from 400 to 800 deployed randomly in the field. The primary transmission range is 30m. It can be extended up to 120m by *link-jumping* operation. The number of nodes for testing varies from 400 to 800. The energy at each node is initialized to be 843.75mJ ($\frac{1}{4 \times 10^3}$ of one AAAA battery). The size of a packet is set to be 38 byte (payload 29 byte + header 9 byte). The average error rate is set to be 10^{-3}. Data rate various from 57.6kbps, 19.2 kbps and 9.6 kbps. We compare the network throughput, network lifetime and reconfiguration cost between SISO WSNs, 2×2 MIMO WSNs, and 4×4 MIMO WSNs by respectively generating and broadcasting data packets until WSNs die (a WSN die if some link on the route has transmission distance larger than 120 m).

Also, in order to see how a critical role the proposed cluster-based architecture plays, we compare it with a flat WSN. In the flat WSN, we use the same MultiOptimal MAC protocol. The broadcast is based on a simplified flooding: each node transmits the received packet only once. Since the flooding can not promise all nodes receive the broadcast packet, we evaluate the performance of the broadcast in which more than 95% nodes received the packet in the sense that the broadcast who achieves more than 95% successful rate will consume more time and energy. Notice that the broadcast on our cluster-based architecture is a deterministic one that promises every node getting the broadcast packet.

Considering the space limitation, we only show the performance graphs for 800 nodes. In Fig 3, C and F in SISO (C), SISO (F), 2×2 (C), 2×2 (F), 4×4 (C) and 4×4 (F) represent the proposed cluster-based structure and flat structure, respectively. The performance for 400 nodes and 600 nodes are similar. As we described in Section 2, an $M \times M$ MIMO can improve the data rate at local links up to *M* times or can extend transmission distance. From Fig 3, we can see that our cross-layer design not only improve

the data rate more than *M* times, but also significantly extend network lifetime and reduce network reconfiguration cost. Therefore, our design achieves the goal of multiple-optimization.

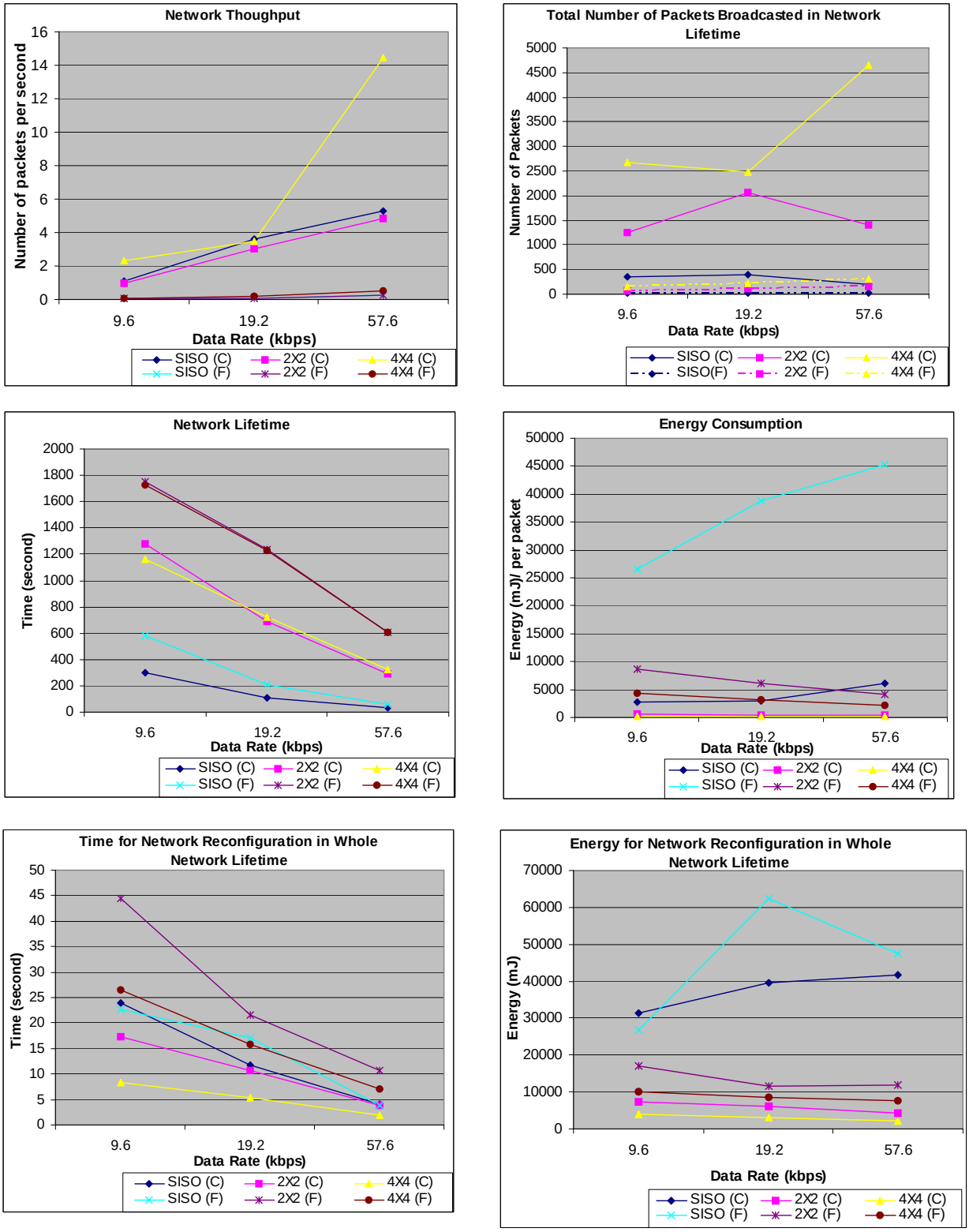

Fig 3. Performance Evaluation of MIMO Sensor Networks

1. L. Bao, J. J. Garcial-Luna-Aceves, “A new approach to channel access scheduling for ad hoc networks,” Proceedings of the 7th Annual International Conference on Mobile Computing and Networking,” pp. 210-221, 2001.
2. B. S. Chlebus, L. Gasieniec, A. M. Gibbons, A. Pelc, and W. Rytter, “Deterministic broadcasting in ad hoc radio networks”, *Distributed Computing* 15, pp. 27-38, 2002.
3. S. Cui, A. J. Goldsmith, “Energy-efficiency of MMIO and cooperative MOMO techniques in sensor networks,” IEEE Journal on Selected Areas in Communication, pp. 1-9, 2004
4. S. Cui, A. J. Goldsmith, A. Bahai, “Energy-constrained modulation opetimization,” IEEE Transactions A dynamic sensor network is a reconfigurable network which supports two functions: node-joining and node-leaving. on Wireless Communications, pp. 2349-2360, Vol. 4, No. 5, 2005.
5. W. Ke, P. Basu, A. Ayyash, T.E.C. Little, “Attribute based hierarchical clustering in wireless sensor networks,” MCL Technical Report TR-03-24-2003, 2003.
6. B. V. Nadimpalli, P. Mulukutla, R. Garimella, Srinivias MB, “Energy-aware routing in sensor networks using dual membership clusters and data highways,” TENCON 2004, 2004 IEEE Region 19 Conference Vol C, Issues 21-24, pp. 184-187, 2004.
7. K. Nakano and S. Olariu, “Randomized initialization protocols for radio networks,” *Handbook of Wireless Networks and Mobile Computing*, pp.195-218, 2002.
8. S. Y. Ni, Y. C. Chen, J. P. Sheu, “The broadcast storm problem in a mobile ad hoc network,” Proceedings of the 5th ACM/IEEE International Conference on Mobile Computing and Networking, pp. 151-162, 1999.
9. S. PalChaudhuri, R. Kumar, R. G. Baraniuk, “Desing of adaptive overlays for multi-scale communication in sensor networks,” *DCOSS*, *LNCS* 3560, pp. 173-190, 2005.
10. V. Rajendran, K. Obraczka, J. J. Garcial-Luna-Aceves, “Energy-efficient collision-free medium access control for wireless sensor networks,” Proceedings of the First International Conference on Embedded Networked Sensor Systems, pp. 181-192, 2003.
11. H. Su, X. Zhang, “Energy-efficient clustering system model and reconfiguration schemes for wireless sensor networks,” Annual Conference on Information Science and Systems, pp. 99-104, 2006.
12. K. Sundaresan, R. Sivakumar, “A unified MAC layer framework for ad-hoc networks with smart antennas,” Proceedings of ACM International Symposium on Mobile Ad-Hoc Networking and Computing, 2005.
13. K. Sundaresan, R. Sivakumar, “Routing in ad-hoc networks with MMO Links,” Proceedings of IEEE International Conference on Network Protocols, 2005.
14. K. Sundaresan, R. Sivakumar, “Medium access control in ad-hoc networks with MIMO links: optimization considerations and algorithms,” IEEE Transactions on Mobile Computing, 2004.
15. T. Tang, M. Park, R. W. Heath, Jr. and Scott M. Nettles, “A joint MIMO-ODF transceiver and MAC design for mobile ad hoc networking,” pp. 315-319, Proceedings of International workshop on Wireless Ad-Hoc Networks, 2004.
16. J. Uchida, I. Muzahidul, Y. Katayama, W. Chen, K. Wada, “Construction and Maintenance of A Cluster-based Architecture for Dynamic Radio Networks,” 39th Hawaii International Conference on System Sciences, 2006.
17. D. Wang, U. Tureli, “Joint MIMO-OFDM and MAC design for broadband miltihop ad hoc networks,” EURASIP Journal on Wireless Communication and Networking, pp. 1-9, Vol. 2006, 2006.
18. L. Zheng, David N. C. Tse, “Diversity and multiplexing: a fundamental tradepff in multiple-antenna channels,” IEEE Transactions on Information Theory, pp. 1073-1096, Vol. 49, No. 5, 2003.