A Cooperative Clustering Protocol for Energy Constrained Networks

Diep N. Nguyen and Marwan Krunz
Department of Electrical and Computer Engineering, University of Arizona
E-mail:{dnnguyen, krunz}@ece.arizona.edu

***Abstract*— Multiple-input multiple-output (MIMO) technology is known to improve energy efficiency in energy-constrained wireless networks, such as wireless sensor networks (WSN). Although in WSNs, a node is often equipped with a single antenna, nodes can be clustered into virtual antenna arrays that can act as virtual MIMO (VMIMO) nodes. In this paper, we propose a distributed cooperative clustering protocol (CCP) that aims at conserving energy and prolonging network lifetime by taking advantage of VMIMO communications. In contrast to previously proposed protocols, CCP fully exploits the diversity gain of the VMIMO technique by optimally selecting the cooperating nodes (CNs) within a cluster and balancing their energy consumption. We first formulate the problem of optimal CN selection at the transmit and receive clusters as a nonlinear binary program, and show the problem is NP-hard. Aiming at minimizing the imbalance in the residual energy at various nodes, we reduce the problem into two sub-problems: finding the optimal number of CNs (ONC) in a cluster and the CN assignment problem. To analytically address the ONC problem, we analyze the energy efficiency of two existing VMIMO methods: distributed Space Time Block Code (DSTBC) and distributed Vertical-Bell Laboratories-Layered -Space-Time (DVBLAST). The second sub-problem is addressed by assigning CNs to nodes with stronger residual energy. To make CCP scalable to large WSNs, we propose a multi-hop energy-balanced routing mechanism for clustered WSNs with a novel cost metric. Our routing method is also applicable to other clustering protocols (e.g., CMIMO, MIMO-LEACH). Extensive simulations are used to validate our analysis.**

***Index Terms*— Energy constrained network, cooperative communication, virtual MIMO, MAC protocol, clustering, routing.**

I. Introduction

The underlying work targets energy-constrained systems such as wireless sensor networks (WSNs). In a WSN, it is often difficult or expensive to replace/recharge sensor batteries after deployment. Hence, it is critical to design the network in an energy-efficient manner. Recently, cooperative communications has attracted substantial research interest as a means of conserving energy and/or increasing network throughput by having groups of nodes cooperate in transmitting or receiving the signal. In principle, cooperative communications exploit the spatial diversity obtained from transmitting the same signal (or highly correlated ones) over multiple, spatially separated antennas. As such, the theory of cooperative communications is closely related to multiple-input multiple-output (MIMO) technology. Under a given power budget and fading conditions, MIMO communications can offer a much higher throughput (through the spatial multiplexing) or more reliable communications (by exploiting diversity gain) than single input single output (SISO) communications. For a conventional MIMO system, space time block codes (STBCs) [1] can be used to provide diversity gain. To achieve multiplexing gain, the VBLAST technique [2] is often used. However, implementing multiple antennas on a sensor node can be impractical. Instead, one could use the concept of virtual MIMO (VMIMO) [3], in which several sensors are grouped to function as a virtual MIMO node. This concept led to distributed VBLAST (DVBLAST) [4] and distributed STBC (DSTBC) [5] [6] [7], as virtual counterparts of conventional VBLAST and STBC MIMO schemes.

This research was supported in part by NSF (under grants CNS-1016943, CNS-0721935, CNS-0904681, IIP-0832238), Raytheon, and the Connection One center. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

WSNs that involve a large number of nodes are often organized into clusters, each with its own cluster head (CH). Clustering provides scalability with regard to communications and processing tasks, facilitating various functions such as data aggregation. It can also be used to facilitate VMIMO communications, whereby a subset of the nodes in a cluster, called the *cooperating nodes* (CNs), serves as a virtual transmit (Tx) or receive (Rx) antenna array [8] [9] [10]. Within a cluster, sensors communicate their sensed data to the CH, which together with other CNs in the cluster forwards the data to the sink, either directly or by relaying it through a multi-hop inter-cluster path. Here, for each inter-cluster link comes the problem of optimal CNs selection (OCS), defined as identifying the appropriate CNs in each cluster that minimize the VMIMO energy consumption of that link.

In the literature, researchers considered the problem of optimizing the number of CNs in each cluster (ONC) (e.g., [9] [7] [11]). Note that ONC is a less general problem than OCS. There are three main limitations for these works. First, in [7] the authors relied on Chernoff bound to approximately compute the per-bit transmission energy (E_b) for STBC in the high SNR regime. Such a regime does not apply to WSNs, whose transmissions are characterized by low SNR and low bit rates. Secondly, the size of the set of candidate nodes from which the CNs are to be selected has been assumed to be known [9] [11]. Moreover, the cooperation overhead has not been accounted for in [7] [10]. Due to the absence of an analytical solution to the ONC problem, the number of CNs per cluster has often been limited to two [10] or that the set of transmitting or receiving CNs (but not both) contains only one node. This limits the diversity gain to the maximum of 2 x 2 in [10] and to 3x1 in [9]. The authors in [10] [7] claimed their works are extendible to more than two CNs per cluster. However, we show that extending the treatment to more CNs requires solving the OCS problem, which is actually NP hard.

Another challenge that has so far confined the number of CNs to two is that the performance depends on the "transmission distance" between the two clusters, defined here as the farthest distance between any node in the set of transmitting CNs and any node in the set of receiving CNs. Previous works (e.g., [9] [7] [11] [10] [8]) overlooked such dependence and assumed that the transmission distance is known a priori. Moreover, the transmission distance often used in the literature is the average distance between the two clusters (e.g., [12]), which has been recently reported to underestimate the energy consumption of a WSN [13]. In this paper, we do not assume prior knowledge of the transmission distance when addressing the OCS and ONC problems.

Addressing the OCS problem requires understanding the tradeoff among diversity gain, multiplexing gain, circuit energy consumption, and cooperation overhead. Regarding the

978-1-4577-0093-4/11/$26.00 ©2011 IEEE

tradeoff between multiplexing and diversity gains, it is still not known how to judiciously use DSTBC or DVBLAST to minimize energy consumption, though both have been proposed as energy-efficient solutions for WSNs. Because nodes are spatially separated, the overhead of coordinating their VMIMO operation (e.g., estimating the transmission distances between the Tx and Rx CNs and solving ONC in real time) may overshadow the potential diversity/multiplexing gain.

To circumvent the NP-hardness of the OCS problem, we approximately decompose it into two sub-problems: the ONC and CN assignment problems, which are solved in a distributed fashion with reasonable overhead. ONC determines how many CNs a cluster should have. Knowing that, the CN assignment problem aims at minimizing the imbalance (variance) in the residual battery energy of various nodes within each cluster, by letting nodes with higher residual energy act as CNs. Regarding the ONC problem, we first obtain an upper bound on the optimal number of CNs in a cluster. This is done offline and only once before network deployment. The optimal number of CNs is then determined by solving ONC subject to this upper bound. As shown later, the processing overheard of this step is small. Taking advantage of our analysis, we design a fully distributed cooperative clustering protocol (CCP) that performs both clustering and optimal selection of CNs in each cluster.

We then extend CCP to multi-hop WSNs and propose a clustering energy-balancing routing (C-EBR) mechanism with a novel cost metric. The proposed C-EBR follows the approach in [14], which attempts to address the traffic implosion problem [15] (i.e., nodes around the sink deplete their battery much faster than other nodes). C-EBR uses the inverse of the minimum residual energy of CNs within a cluster as its cost metric. Our method can be applied to other clustering protocols (e.g., CMIMO, MIMO-LEACH) to prolong network lifetime. The effectiveness of C-EBR is demonstrated via simulation. At the time when the first node runs out of energy, the average remaining energy of all nodes that can reach the sink directly is as low as 1% of the initial battery levels, compared to 48% for the case without C-EBR.

In Section II, we formulate the OCS problem, and decompose it into two sub-problems. In Section III, we analytically evaluate the energy efficiencies of DSTBC and DVBLAST, and optimize the number of CNs. The C-EBR scheme is proposed in IV. In Section V, we introduce CCP. We investigate its performance in VI, and compare it with the protocols in [16] [9] [10]. Finally, concluding remarks are provided in Section VII.

II. Problem Formulation

In this section, we first formulate the OCS problem and discuss its computational complexity. Consequently, we approximately decompose this problem into two sub-problems: the ONC problem and the CN assignment problem.

Consider a WSN with N Poisson distributed nodes of density μ. The intra-cluster communications range of a node is R_{intra}, corresponding to Tx power P_{intra}. Nodes are assumed to be capable of controlling their transmission powers to adjust their transmission range up to R_{inter}, the inter-cluster transmission range (which corresponds to Tx power P_{inter}). R_{intra} and R_{inter} are input parameters, used to ensure that the set of VMIMO nodes forms a connected dominating set. In our simulation, let $R_{\text{inter}} = 4R_{\text{intra}}$, which can be easily shown to satisfy the connectivity criterion. Let the residual battery energy of a node i at a given time be e_i. To facilitate VMIMO operation, we assume a synchronization mechanism is in place. For example, the recently developed method in [17] can realize VMIMO with a packet synchronization level. Time is divided into slots, where each slot represents the interval between two adjacent re-clustering instances. In a given time slot, there are three types of nodes in the network: ordinary nodes (ONs), CNs, and a CH, which is also a CN. All nodes can collect data from the field. The CH is responsible for determining the CNs in its cluster. Each slot starts with a clustering mini-slot t_{cl}, followed by a cooperation mini-slot t_{co}, and finally multiple data transmission slots t_{data}. CH election and cluster formation (during the clustering mini-slot) are discussed in Section V. In the cooperation mini-slot, the CH selects several CNs in such a way that the resulting CN set can communicate as a VMIMO node with its neighbor clusters (whose CHs are within R_{inter} distance) using the minimum possible energy.

For a given link between two neighboring clusters, suppose that the number of nodes in the Tx and Rx clusters are m_t and m_r, respectively. Define $b_t^{(i)}$ as a binary variable such that $b_t^{(i)} = 1$ if node i of the Tx cluster is chosen as a CN and $b_t^{(i)} = 0$, otherwise. Similarly, the variable $b_r^{(j)}$ is defined for the Rx cluster. Let $\boldsymbol{b}_t \stackrel{\text{def}}{=} (b_t^{(1)}, b_t^{(2)}, \ldots, b_t^{(m_t)})$ and $\boldsymbol{b}_r \stackrel{\text{def}}{=} (b_r^{(1)}, b_r^{(2)}, \ldots, b_r^{(m_r)})$. The VMIMO methods on link l can be either DSTBC or DVBLAST. The OCS problem is formulated as follows:

$$\underset{\{\boldsymbol{b}_t,\boldsymbol{b}_r\}}{\text{minimize}} \; \left\{E^*(\boldsymbol{b}_t, \boldsymbol{b}_r) + \left(\frac{PL}{R} + H\right)(M_t + M_r)\right\}$$
$$s.t.\;\; b_t^{(i)} \text{ and } b_r^{(j)} \text{ are } 0 \text{ or } 1 \text{ for } i{=}1,\ldots,m_t \text{ and } j{=}1,\ldots,m_r \tag{1}$$

where $E^*(\boldsymbol{b}_t, \boldsymbol{b}_r)$ is the per-packet RF energy consumption using either DSTBC or DVBLAST, L is the packet size in bits, R is the transmission rate in bps, H is the energy overhead to employ one CN, P is the circuit power consumption of one CN. Here we use the fact that a node consumes approximately the same amount of circuit energy when transmitting or receiving [18] [19]. $M_t \stackrel{\text{def}}{=} \sum_{i=1}^{m_t} b_t^{(i)}$ and $M_r \stackrel{\text{def}}{=} \sum_{j=1}^{m_r} b_r^{(j)}$. We emphasize that $E^*(\boldsymbol{b}_t, \boldsymbol{b}_r)$ does not only depend on the number of antennas M_t and M_r at the Tx and Rx clusters, but also on the sets of CNs on both sides.

As seen later, the objective function in (1) is implicitly nonlinear in the variables $\boldsymbol{b}_t$ and $\boldsymbol{b}_r$ (due to the nonlinearity of $E(\boldsymbol{b}_t, \boldsymbol{b}_r)$, discussed in Section III). Thus, the problem is a nonlinear binary optimization problem, which in general is NP-hard [20]. To develop a computationally affordable, distributed solution to the OCS problem, we decompose (1) into two sub-problems: ONC and CN assignment. Regarding the ONC sub-problem, let M_t^* and M_r^* denote, respectively, the optimal numbers of CNs at the Tx and Rx clusters at a given transmission distance d. These M_t^* and M_r^* are the solution to the following problem:

$$\underset{\{M_t,M_r\}}{\text{minimize}} \; \left\{E(d, M_t, M_r) + (\frac{PL}{R} + H)(M_t + M_r)\right\} \tag{2}$$

where $E(d, M_t, M_r)$ is the per-packet RF energy consumption using either DSTBC or DVBLAST at the transmission distance d, assuming M_t and M_r antennas at the Tx and Rx clusters, respectively. It should note that we neither assume any constraint on the number of CNs nor knowledge of the transmission distance. The estimation of transmission distance is incorporated in the CCP protocol in Section V.

For given M_t^* and M_r^* values, there may be several possible Tx and Rx CNs. Thus, once M_t^* and M_r^* are computed from (2), the sets of M_t^* and M_r^* CNs are selected in such a way that the variance of residual energy is minimized. Specifically, at the Rx cluster, we aim at:

$$\underset{\{\boldsymbol{b}_r\}}{\text{minimize}}$$
$$\frac{1}{m_r}\sum_{j=1}^{m_r}\left(e_j - b_r^{(j)}\frac{PL}{R} - \frac{\sum_{k=1}^{m_r} e_k - M_r^*(\frac{PL}{R}+H)}{m_r}\right)^2 \quad (3)$$
$$s.t. \quad \sum_{j=1}^{m_r} b_r^{(j)} = M_r^*.$$

After knowing the set of CNs at the Rx cluster, vector $\boldsymbol{b}_t$ is found by solving:

$$\underset{\{\boldsymbol{b}_t\}}{\text{minimize}}$$
$$\frac{1}{m_t}\sum_{i=1}^{m_t}\left(\begin{array}{c} e_i - b_t^{(i)}\left(\frac{E^*(\boldsymbol{b}_t,\boldsymbol{b}_r)}{M_t^*}+\frac{PL}{R}+H\right) \\ -\frac{\sum_{k=1}^{m_t} e_j - E^*(\boldsymbol{b}_t,\boldsymbol{b}_r) - M_t^*(\frac{PL}{R}+H)}{m_t}\end{array}\right)^2 \quad (4)$$
$$s.t. \quad \sum_{i=1}^{m_t} b_t^{(i)} = M_t^*.$$

In (3) and (4), the first and second terms of the objective functions represent the residual energy after cooperation. The third term is the updated mean of the residual energy of all nodes.

Problems (3) and (4) are addressed in Section IV. Problem (2) is a nonlinear integer programming. To tackle it, we need to get some insight into VMIMO techniques, namely DSTBC and DVBLAST.

III. Optimizing the Number of CNs

A. Energy Consumption of DSTBC

For DSTBC, data bits are modulated into S symbols with b bits per symbol. These symbols are then mapped into an M_t by T matrix, whose columns are transmitted sequentially over T channel uses (hence, the code rate $r{=}S/T$). The number of bits per channel use is bS/T and the transmission rate is $R = BbS/T$, where B is the channel bandwidth in Hz. Following [21], the symbol error rate (SER) for an $M_t \times M_r$ STBC is:

$$\begin{aligned} P_s(E_{\text{b-STBC}}) = &\frac{2(1-1/\sqrt{M})\phi_\eta(1.5/(M-1))}{\sqrt{\pi}}\frac{\eta(M_tM_r+0.5)}{\eta(M_tM_r+1)} \\ &\times {}_2F_1\left\{M_tM_r, 0.5; M_tM_r{+}1; \frac{1}{1{+}\bar{\bar{\eta}}\,1.5/(M{-}1)}\right\} \\ &- \frac{2(1{-}1/\sqrt{M})^2}{\pi}\frac{\phi_\eta(3/(M-1))}{2M_tM_r+1}\times \\ &F_1\left\{1,M_tM_r,1;M_tM_r{+}1.5;\frac{1{+}\bar{\bar{\eta}}\,1.5/(M{-}1)}{1{+}\bar{\bar{\eta}}\,3/(M{-}1)},\frac{1}{2}\right\} \end{aligned} \quad (5)$$

where

$E_{\text{b-STBC}} = \frac{E_s}{b}$ $\equiv$ Per-bit transmission energy of STBC

$\zeta \overset{\text{def}}{=} \|\hbar\|_F^2$ $\equiv$ Frobenius norm of the channel matrix $\hbar$

$\eta = \zeta\frac{E_s}{M_tRN_o}$ $\equiv$ Instantaneous symbol-to-noise energy ratio

$\bar{\bar{\eta}} = \frac{E_s}{M_trN_o}$

$M = 2^b$ $\equiv$ Modulation order

$\phi_\eta(s) \overset{\text{def}}{=} \text{MGF}(\eta) = E[e^{s\eta}] = (1+s\,\bar{\bar{\eta}})^{-M_tM_r}.$

${}_2F_1$ and F_1 are hypergeometric functions with one and two variables, defined as [22] [23]:

$$F_1\{a,b,c;d;x,y\} \overset{\text{def}}{=} \sum_{n,m=0}^{\infty}\frac{(a,m+n)(b,m)(c,n)}{(d,m+n)(1,m)(1,n)}x^my^n$$
$${}_2F_1\{a,b;c;x\} \overset{\text{def}}{=} \sum_{n=0}^{\infty}\frac{(a,n)(b,n)}{(c,n)}\frac{x^n}{n!}$$
$$\text{with} \quad (a,n) \overset{\text{def}}{=} a(a+1)...(a+n-1).$$

It is easy to verify that the variables in (5) that correspond to the variables x and y in the definitions of ${}_2F_1$ and F_1 are less than 1 and satisfy the convergence condition of hypergeometric functions. Using recently suggested algorithms in [24], we can compute hypergeometric functions up to 13-digit accuracy. Specifically, for relatively small M_t and M_r (less than 10), a truncated Taylor series method can be used. For larger M_t and M_r, the Gauss Jacobi quadrature or recurrence relation method yields higher accuracy. Built-in hypergeometric functions are also available in numerical tools, e.g., MATLAB.

If Gray mapping is used to map bit patterns into the modulation constellations, then the BER can be induced from the SER, as follows:

$$P_b(E_{\text{b-STBC}}) = \frac{P_s(E_{\text{b-STBC}})}{\log_2(M)}. \quad (6)$$

For a given target BER, $E_{\text{b-STBC}}$ is obtained by inverting (6). The transmission energy to send L bits at distance d is given by:

$$E_{\text{Tx-STBC}} = \psi E_{\text{b-STBC}} L d^\alpha \quad (7)$$

where ψ depends on design parameters (e.g., operating frequency, Tx/Rx antenna gains, etc.) and α is the attenuation factor, ranging from 2 to 6. It is worth noting that the $E_{\text{b-STBC}}$ obtained by inverting (6) is exact. It is a nonlinear function in $\boldsymbol{b}_t$ and $\boldsymbol{b}_r$ due to their coupling in the product M_tM_r in (5). In the literature, Chernoff bound is used to approximately compute $E_{\text{b-STBC}}$ under the assumption that the system operates in the high SNR regime. As explained before, such an approximation is not practical in WSNs.

The time duration needed to send L bits is $T_{\text{on}}{=}\frac{L}{R}$. Thus, the circuit energy consumption under DSTBC, denoted by $E_{\text{C-STBC}}$, is P_cT_{on}, where the total circuit power P_c is

$$P_c = (M_t + M_r)P. \quad (8)$$

Accordingly, the total energy consumption (RF transmission plus circuit) to send L bits under the DSTBC scheme is:

$$E_{\text{DSTBC}}(d, M_t, M_r) = E_{\text{Tx-STBC}} + E_{\text{C-STBC}} = \psi E_{\text{b-STBC}} L d^\alpha + \frac{P_cL}{R}. \quad (9)$$

Notice that E_{DSTBC} is a function of the number of CNs at both the Tx and Rx. In Section III-C, we use (9) to solve the ONC problem. Before going further, we take a detour to justify the use of DSTBC instead of DVBLAST for VMIMO communications in WSNs.

B. DVBLAST Versus DSTBC

Both DVBLAST and DSTBC can be used to conserve energy in WSNs [4] [7]. In this section, we compare the energy efficiency of DVBLAST and DSTBC. At the maximum diversity gain M_tM_r, DSTBC requires less E_b than DVBLAST. However, for multiplexing gain, DVBLAST can offer a significantly higher transmission rate (ideally, M_t folds). The transmission time is then $1/M_t$ that of DSTBC, allowing DVBLAST to save energy by shortening the circuit active time.

The energy per bit for VBLAST, denoted by $E_{\text{b-VBLAST}}$, was derived in [4], where M-QAM was used between the M_t and M_r CNs. It can be obtained by inverting the BER $P_b(E_{\text{b-VBLAST}})$ in the following expression:

$$P_b(E_{\text{b-VBLAST}}) \approx [1 - \prod_{t=1}^{M_t}(1-\wp(t))]\left(\frac{1}{8}+\frac{1}{bM_t}\right) \quad (10)$$

where

$$\wp(t) = 4\left(1-\frac{1}{\sqrt{M}}\right)\left(\frac{1-\eta_t}{2}\right)^{M_r-M_t+t} \times \sum_{j=1}^{M_r-M_t+t-1}\binom{M_r-M_t+t-1+j}{j}\left(\frac{1-\eta_t}{2}\right)^j$$

$$\eta_t = \frac{3bE_{\text{b-VBLAST}}}{3bE_{\text{b-VBLAST}}+2(M-1)N_0}.$$

Similar to (7), the transmission energy $E_{\text{Tx-VBLAST}}$ to send L bits via VBLAST is $\psi E_{\text{b-VBLAST}} L d^\alpha$. The circuit energy for DVBLAST is:

$$E_{\text{C-VBLAST}} = \frac{P_c}{R_{\text{b-VBLAST}}}L \quad (11)$$

where P_c is the same as in (8) and $R_{\text{b-VBLAST}}$ is the transmission rate under VBLAST. Ideally $R_{\text{b-VBLAST}}$ is M_t times greater than that of STBC, so $E_{\text{C-VBLAST}}$ is M_t times less than $E_{\text{C-STBC}}$. Thus, the total required energy to send L bits under DVBLAST is

$$E_{\text{DVBLAST}}(d, M_t, M_r) = E_{\text{Tx-VBLAST}} + E_{\text{C-VBLAST}} = \psi E_{\text{b-VBLAST}} L d^\alpha + \frac{P_c L}{R M_t}. \quad (12)$$

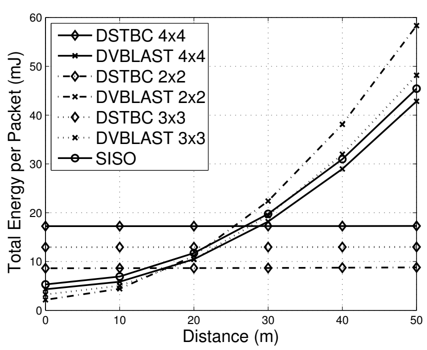


Fig. 1. Energy efficiency of DVBLAST vs. DSTBC.

In Figure 1, we compare (9) and (12) when BER=10^{-4}, under Rayleigh fading and using the parameters in Table I. Notice that DSTBC is more energy efficient than DVBLAST for $d \geq 25$ meters. Hence, the reduction in the transmission time (consequently, circuit energy consumption) in DVBLAST cannot compensate for the energy consumed in sending data at a higher rate. DSTBC outperforms DVBLAST as it maximizes the diversity gain, therefore requiring significantly lower energy, though the transmission duration is M_t times longer than that of DVBLAST.

C. Optimal Number of Cooperating Nodes

Previous works (e.g., [7]) showed that for long-haul transmissions, more CNs are needed, as the transmission power dominates the total power consumption in (9). On the other hand, shorter distances favor less CNs or even a SISO transmission, as circuit power becomes dominant. For a given transmission distance between two sets of CNs, we seek to find the pair (M_t^*,M_r^*) that gives the lowest total energy consumption. This optimal (M_t^*,M_r^*) is the solution of (2), a nonlinear integer

Transmission rate	400 Kbps
Operating frequency	$f_c = 2.5$ GHz
P	105 mW
ψ	$(1+\tau)\frac{(4\pi)^2}{G_t G_r \lambda^2}$
M	4 (4-QAM)
$G_t G_r$	5 dBi
τ	0.45658
L	2000 Bytes
H	$160 \times 3 \times E_b = 480\ E_b$
R_{intra}	180 m
μ	6.10^{-4}

TABLE I
PARAMETER VALUES.

programming problem. It can be solved by the branch-and-bound method, with exponential complexity in the worst case. Here, we use the method of strong inequalities [25], which first requires determining upper bounds on M_t^* and M_r^*. We emphasize that these bounds are found offline and are embedded into CCP as design parameters.

Consider (2) at the maximum possible transmission distance $d = R_{\text{inter}}$.

$$\underset{\{M_t, M_r\}}{\text{minimize}} \left\{E_{\text{Tx-STBC}}(R_{\text{inter}}, M_t, M_r)+(M_t+M_r)(\frac{PL}{R}+H)\right\} \quad (13)$$

The above formulation is an integer programming problem. The two variables M_t and M_r are upper bounded by the number of nodes in a cluster of radius R_{intra}, which depends on node density μ. Note that the problem can be solved offline, once before deploying the network. Thus, its complexity is not of great concern. Problem (13) can be solved using branch-and-bound method (or even via exhaustive search). The solution to (13), denoted by (M_t^+, M_r^+), is the best tradeoff between transmission energy and circuit energy at distance R_{inter}. Let $U \stackrel{\text{def}}{=} \max(M_t^+, M_r^+)$; $u \stackrel{\text{def}}{=} \min(M_t^+, M_r^+)$.

Theorem 1: For a given transmission distance $d \leq R_{\text{inter}}$ and given system parameters (e.g., transmission rate, modulation order, etc.) there is no energy benefit to have more than U CNs at either the Tx or Rx side, in problem (2).

Proof (by contradiction): Assume that for a given distance $d < R_{\text{inter}}$, (M_{t1}, M_{r1}) is the optimal solution to (2), and suppose that M_{t1} (and/or M_{r1}) is greater than U.

Case 1*:* Suppose that $M_{t1} + M_{r1} \geq U + u$. As assumed, (M_{t1}, M_{r1}) is the optimal solution to (2) at distance d. Then,

$$E_{\text{DSTBC}}(d, M_{t1}, M_{r1}) = \psi E_{\text{b-STBC}}^{(M_{t1},M_{r1})} L d^\alpha + (M_{t1}+M_{r1})(\frac{PL}{R}+H) \leq \psi E_{\text{b-STBC}}^{(u,U)} L d^\alpha + (U+u)(\frac{PL}{R}+H).$$

Hence,

$$E_{\text{b-STBC}}^{(u,U)} - E_{\text{b-STBC}}^{(M_{t1},M_{r1})} \geq \frac{\Theta}{\psi L d^\alpha} \quad (14)$$

where

$$\Theta \stackrel{\text{def}}{=} (M_{t1}+M_{r1}-U-u)(\frac{PL}{R}+H) \geq 0, \quad \text{as } M_{t1}+M_{r1} \geq U+u.$$

At distance R_{inter}, assume (u, U) is the optimal solution to (13). Then,

$$E_{\text{DSTBC}}(R_{\text{inter}}, u, U) = \psi E_{\text{b-STBC}}^{(u,U)} L R_{\text{inter}}^\alpha + (U+u)(\frac{PL}{R}+H) \leq \psi E_{\text{b-STBC}}^{(M_{t1},M_{r1})} L R_{\text{inter}}^\alpha + (M_{t1}+M_{r1})(\frac{PL}{R}+H).$$

Accordingly:

$$E_{\text{b-STBC}}^{(u,U)} - E_{\text{b-STBC}}^{(M_{t1},M_{r1})} \leq \frac{\Theta}{\psi L R_{\text{inter}}^{\alpha}}$$

which contradicts (14), as $R_{\text{inter}} > d$ and $\Theta \geq 0$.

Case 2: Suppose $M_{t1} + M_{r1} < U + u$. Let $K \stackrel{\text{def}}{=} M_{t1} + M_{r1}$ and $m \stackrel{\text{def}}{=} \lceil \frac{K}{2} \rceil$. If the E_b requirement under (M_{t1}, M_{r1}) is lower than or equal to that under (u, U) (or (U, u)), then (M_{t1}, M_{r1}) would be the optimal combination at $d = R_{\text{inter}}$, leading to a contradiction (as (U, u) or (u, U) is the optimal solution of (13)).

If the E_b requirement under (M_{t1},M_{r1}) is higher than that under (U, u) or (u, U), then $K = 2m$ if K is even, or $K = 2m - 1$ if K is odd. Clearly, $m \leq U$ and $m^2 \geq M_{t1}M_{r1}$ if K is even, or $m(m-1) \geq M_{t1}M_{r1}$ if K is odd (Cauchy's inequality). We show the contradiction by proving that (m,m) or (($m-1$),m) DSTBC conserves more energy than (M_{t1},M_{r1}) DSTBC does. For $2m = K$, (m,m) and (M_{t1},M_{r1}) DSTBC consume the same amount of circuit and overhead energy. However, (m, m) DSTBC offers higher diversity gain than (M_{t1}, M_{r1}) DSTBC (m^2 compared with $M_{t1}M_{r1}$). Then, the (m, m) combination is more energy efficient than (M_{t1}, M_{r1}). Similarly, if $2m = K+1$, (($m-1$),m) DSTBC is more energy efficient than (M_{t1}, M_{r1}). This leads to a contradiction (as (M_{t1}, M_{r1}) is the optimal solution to (13)). This completes the proof. ∎

Using the parameters in Table I, Figure. 2 depicts the total energy per packet for different DSTBCs versus the transmission distance. At the transmission range $R_{\text{inter}} = 380$ meters, the optimal number of CNs is bounded by 5. This bound is 7 at $R_{\text{inter}} = 580$ meters and 8 at $R_{\text{inter}} = 720$ meters.

Now, the ONC problem becomes:

$$\begin{aligned} \underset{\{M_t,M_r\}}{\text{minimize}} \; & \left\{ E_{\text{Tx-STBC}}(d, M_t, M_r) + (M_t + M_r)(\frac{PL}{R} + H) \right\} \\ s.t. \; & M_t \leq U \\ & M_r \leq U. \end{aligned} \tag{15}$$

Note that although the complexity of the ONC problem depends on the node density and the intra-cluster range, the size of (15) is independent of these parameters, and is determined by U. The optimal pair (M_t^*,M_r^*) is found by inspecting U^2 possible combinations.

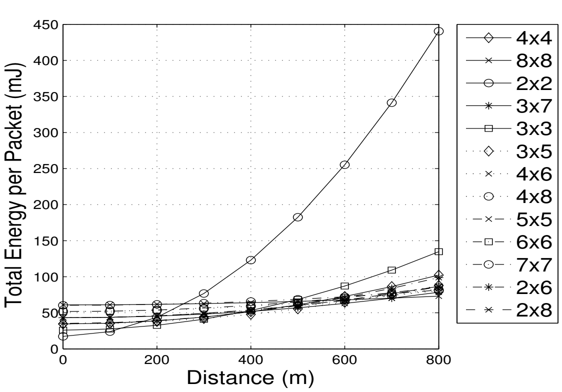


Fig. 2. Energy per packet vs. distance for DSTBC under different (M_t,M_r) combinations.

The solution space of (15) is now small enough (e.g., 25, 49, 64 for the above R_{inter} values) to be embedded on every sensor. Under the parameters in Table I, there are at most 89 nodes per cluster with probability of 0.998. Therefore, without the upper bounds, in the worst case, a sensor node would have to solve problem (2) with 7921 possible combinations.

The transmission distance varies for different sets of CNs at both sides. The next section addresses the CN assignment, which determines the transmission distance.

D. CN Assignment Problem

In this section, energy balancing is realized within each cluster to solve problems (3) and (4). It can be seen that (3) is a binary programming problem. The following procedure solves (3) with complexity of $O(m_r \log m_r)$:

1) Sort m_r nodes of the Rx cluster in a descending order of their residual energy.
2) Pick the first M_r^* nodes to act as CNs for the Rx cluster.

Given the set of CNs at the Rx cluster, (4) is still a nonlinear binary programming problem. A heuristic solution to (4) can also be obtained by using the above procedure. The rationale behind our approach is to balance energy drainage among different nodes at the Tx cluster. CNs and CHs, which are more energy demanding, should have higher residual batteries.

In [10], CNs are selected based on the distance to the CH. In [8] [9], CNs are selected based on the ratio of their residual energy and distances to the CH. We follow a different CN assignment approach, which is shown to yield better energy balancing and subsequently up to 80% longer network lifetime compared with the method using the CNs assignment method in [10].

By considering U^2 possible (M_t^*,M_r^*) pairs, along with our novel CN assignment procedure, we can determine the sets of CNs for both the Tx and Rx clusters with computational complexity of $O(m_r \log m_r + m_t \log m_t + U^2)$.

IV. ENERGY-BALANCED ROUTING IN CLUSTERED WSNS

In this section, we propose a distributed energy-balanced routing mechanism for clustered WSNs. At given intra-cluster range and node density, the intra-cluster traffic is almost the same for all clusters. However, the closer a cluster is to the sink, the more inter-cluster traffic it must relay, leading to faster energy drainage of its CNs. This phenomenon is known as *traffic implosion* [15]. In [15], the authors proposed a method to balance power consumption of CHs of all clusters by balancing inter- and intra-cluster traffic. Consequently, CHs that are on more favorable routes to the sink (i.e., serve more inter-cluster traffic) have smaller cluster sizes, i.e, support less intra-cluster traffic, and vice versa. However, this method does not apply to our setup for three reasons. First, it does not consider node cooperation as there is only one node, the CH, which is responsible for inter-cluster communications. Second, the authors in [15] differentiated CHs and CNs from ordinary nodes while our clustering approach selects CHs and CNs from them. Thirdly, the approach in [15] was intended for a centralized design.

Energy-balanced routing (EBR) has drawn substantial attention in the last decade. In their pioneering work, Chang and Tassiulas [14] formulated EBR problem as a linear programming problem, with the goal of maximizing network lifetime. A heuristic algorithm called Maximum Residual Energy Path Routing (MREP) was proposed which provides 96% performance of the optimal solution. MREP routes packets on the path that has the maximum remaining energy.

In the context of clustered WSNs, in addition to the per-packet energy consumption between two clusters, we propose to incorporate the inverse of the minimum residual energy (e_v) of all CNs within the receiving cluster into the link cost. For two neighboring clusters u and v, we define the weight of their inter-cluster link as:

$$w(u, v) = \beta c(u, v) + (1 - \beta)(\frac{1}{e_v}) \tag{16}$$

where β is a tradeoff factor (between 0 to 1) between energy-efficient routing and EBR and $c(u, v)$ is the per-packet energy consumption for a transmission from from u to v. If $\beta = 1$, a shortest path algorithm using the above cost metric would find the least energy consumption path to the sink. On the other hand, if $\beta = 0$, the algorithm reduces to *pure* EBR, which favors the path whose CNs' minimum residual energy is the maximum among all paths. We refer to a routing strategy that uses the metric w as clustering EBR (C-EBR). Implementation of C-EBR does not require establishing the lexicographical ordering of possible paths, as in [14]. C-EBR's complexity is equal to that of the distributed Bellman Ford algorithm, i.e., $O(|E||V|)$, where $|V|$ is the number of clusters in the network and $|E|$ is the number of communication links among these clusters. In the next section, we provide the operational details of the CCP protocol, which implements C-EBR and OCS algorithms.

V. Cooperative and Clustering Protocol (CCP)

CCP consists of three phases: clustering/re-clustering, cooperation, and transmission. The first two phases are executed less frequently than the last phase. These two phases are required only at the beginning of network deployment, or if a cluster reaches its re-clustering threshold (a threshold on the residual energy of a CN). In the clustering phase, each node learns about its neighbors and their residual energy. The network is then partitioned into clusters, each of which has one CH, at most $(U-1)$ other CNs (U is found from (13)), and member nodes. In the cooperation phase, the CH of each cluster calculates and updates other clusters the cost (as defined in (16)) from it to its direct neighbors. After a few rounds of message exchanges, each CH establishes its cluster's optimal route to the sink. The transmission phase is reserved for intra- and inter-cluster communications, in which member nodes send data to their CHs for aggregation. CHs then cooperate with CNs to forward traffic to the sink along paths established during the cooperation phase.

Phase 1: Clustering

Step 1: Neighborhood discovery

Each node u accesses the channel using a CSMA/CA channel access procedure. It then broadcasts to its neighbors a *hello message* (HM) at power P_{intra}. HM contains the node's ID, its residual energy, and a list of its neighbors. After receiving the HM, a neighboring node v updates its neighbor list with u's ID and residual energy. The retransmission of a HM is used in case of collisions of previous HMs. Sending of the second or third HM from node u may be triggered if and only if it receives a HM from a node whose neighbor list does not include u. Each node decides the termination of the neighbor discovery stage after sensing some free mini-slots of the channel. At the end of the neighborhood discovery stage, each node u maintains a list of its neighbors plus itself in an descending order of residual energy. Let such a list be denoted by $\mathcal{N}(u)$.

Step 2: CH election

Node u checks its position in the list $\mathcal{N}(u)$. It declares itself as a CH (change its status to CH) if it is in the first position in $\mathcal{N}(u)$ (highest residual energy in its neighbor list) by contending for channel access and broadcasting a CH announcement message (CHM). The CHM contains u's ID, and is sent at power P_{intra}). Upon hearing a CHM, a node v changes its status to *member*, estimates the distance from itself to that CH, updates its tentative CH if u's CHM is the first one v receives or if the new CH is closer to v than the current CH, and finally broadcasts a *member* announcement message (MAM). The MAM contains v's ID. After hearing a MAM from v, its neighbors, whose statuses have not been determined, mark node v's status (i.e., do not consider v in the CH election process anymore) and reorder their neighbor list. Again, each node whose status has not been determined checks its position in the neighbor list, and declares itself as CH if it is the first in the list. The process is repeated until there is no unmarked nodes in any neighbor list. At the end of the CH election stage, there are two types of nodes: CHs and members. Each member node has one tentative CH.

Step 3: Member association and clustering

A member node contends for the channel and sends a membership request message (MRM) with its ID to its tentative CH. A CH may receive multiple MRMs intended to it. After a certain interval, the CH sorts its member list according to their residual energies and broadcasts a membership confirmation message (MCM). The MCM contains a list of all IDs from which the node received MRMs and their time slot assignment for sending data to the CH. At the end of this stage, the network consists of multiple clusters, each of which has one CH and zero or more member nodes. The sink itself is assumed to be as a cluster with only one node, equipped with U antennas.

Step 4: CNs invitation

Using power P_{intra}, each CH broadcasts an CN invitation message (CIM) to the first $(U-1)$ nodes in its member list if its list contains at least $(U-1)$ members (if the list contains less than $(U-1)$ nodes, the CIM contains the IDs of the whole list). Upon receiving a CI, each member node checks if the message is from its CH by comparing its CH's ID with the sender's ID. It then sends a CN confirmation message to its CH. These selected CNs and CHs are ready to enter the cooperation phase.

Phase 2: Cooperation

When a cluster u is freshly formed, the CH and the CNs of that cluster sequentially broadcast a cot update request (CURM) message at power P_{inter}. This message ensures that all neighboring CNs of cluster u can hear the message. These CURMs are used by CNs at neighbor clusters to estimate the distance (e.g., using receive signal strength) between the two farthest nodes of two sets of CNs. The transmission distance d is then fed into (2). After solving problems (2), (3) and (4), the CHs of neighboring clusters determines the optimal CNs for the sening and receiving clusters. Each CH in a neighboring clusters v sequentially broadcasts the link cost (as in (16)) and the optimal number of CNs of u at power P_{inter} to inform cluster u as well as to update other neighbor clusters of the cost from u to v. The CH of each cluster maintains a cost and a forwarding table to any cluster that it may have learnt about, including the sink.

Phase 3: Transmission

Each node with data to send transmits its data to its CH in the node's time slot (specified in the MCM) for aggregation purposes. After receiving data from various members, the CH performs aggregation/data fusion. It then broadcasts the fused data to its CNs at power P_{intra}. The CH then sends a request-to-send (RTS) message at power P_{inter} to the cluster in its forwarding table. The CH of this Rx cluster, if not busy, sends a clear-to-send message (CTS) to the source cluster and coordinates its CNs to receive data. Upon receiving the CTS, the CH of the source cluster synchronizes its CNs to send data to the destination cluster.

Phase 4: Re-clustering

At some point, the energy of a CN of a cluster may reach the cluster's re-clustering threshold. The CN will send a re-clustering request message (RCRM) at power P_{inter}. CHs that receive a RCRM finish their transmission phase if they are transmitting/receiving and broadcast RCRMs at power P_{inter}. After a few time slots, all nodes become aware of the RCRM and begin the clustering phase (phase 1). How to choose a good threshold to recluster is left for future work. In CCP, the re-clustering threshold is set to 50% of the average residual energy of the cluster's CNs. Hence, it always adapts to the instantaneous battery status and changes from a cluster to another. An overview of CCP is given in Algorithm 1.

Algorithm 1 Cooperative and Clustering Protocol

For every time slot T:
At clustering mini-slot t_{cl}**:**
Neighborhood discovery: Node u maintains a neighbor list $\mathcal{N}(u)$.
CH election based on $\mathcal{N}(u)$ lists.
Member association and clustering.
CNs invitation. There will be U or less CNs. U is found from (13).
At cooperation mini-slot t_{co}**:**
CHs and CNs exchange cost update messages. Problems (2), (3) and (4) are executed at the receive clusters. Optimal paths to the sink of clusters are obtained by executing the distributed Bellman Ford using the cost metric (16)
At transmission mini-slot t_{data}**:**
Intra/inter cluster transmissions, data compression take place using parameters and paths determined during the clustering and cooperation mini-slots. Eventually, re-clustering happens, the network moves to the next time slot $T+1$.

VI. Simulations

A. Simulation Setup

In this section, we evaluate the performance of the OCS algorithm, the CN assignment algorithm, and the C-EBR mechanism. Our simulation programs were coded in C using the CSIM library [26]. The performance metrics include network lifetime, the energy consumption per packet and the residual energy variance (averaged over the network lifetime). As in [14], the network lifetime is defined by the instance the first node runs out of its battery. It is measured in the number of transmission slots (*rounds*). In each transmission slot, a node may have a packet to send with probability q, between 0.3 to 0.9. This q reflects the traffic intensity. We use the following simulation setup: 600 sensors are randomly distributed on a field of 1000×1000 square meter. The data collection center is located at the corner of the field, and is equipped with U antennas. The value of U is obtained from (13). We use the same parameters as in Table I, which results in $U = 8$ for $R_{\text{inter}} = 720$ meters. To only highlight the diversity gain of VMIMO, we do not consider data fusion at CHs, whose gain has been extensively demonstrated in the literature. The routing method is C-EBR, where the tradeoff parameter β is varied from 0 (pure EBR) to 1 (energy-efficient routing). We set the target BER to 10^{-4}. Each control packet is 160-bits long. In CCP, we need 3 control messages (CIM, CN confirmation and CURM) to employ a CN. Thus, $H = 480\ E_b$ in Table I. We use a Rayleigh channel fading model with attenuation factor of 4. At the beginning of a simulation run, each node is equipped with a battery of 500 Joules. The data output is averaged over 13 simulation runs with different seeds.

B. Optimal Selection of CNs

To demonstrate the judiciousness of the OCS algorithm, we compare the network lifetime under CCP with its predecessors: MIMO-LEACH [9] and CMIMO [10]. Because CCP is a clustering protocol, it is worth comparing its performance with other clustering protocols that were intended for SISO communications, e.g., DCA [16]. For MIMO-LEACH, we set the number of CNs to three. To isolate the effect of the routing mechanism, this comparison is made using energy efficiency routing ($\beta = 1$).

Figure 3 shows that, on average, the network lifetime under CCP is about 105%, 90%, and 50% higher than that of DCA, MIMO-LEACH, and CMIMO, respectively. For all protocols, the network lifetime decreases with the traffic intensity. However, when the traffic intensity increases, the CCP's network lifetime improvement over the other protocols becomes more pronounced. This is due to the fact that CCP considers the cooperation overhead when searching for optimal CNs, whereas

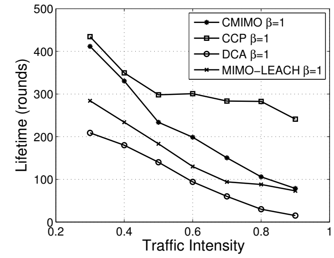


Fig. 3. Network lifetimes vs. traffic intensity for CCP, CMIMO, MIMO-LEACH, and DCA.

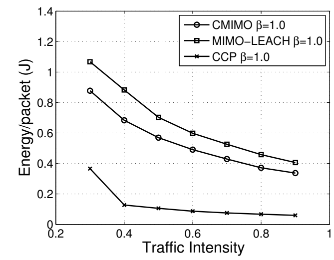


Fig. 4. Average energy per packet from a node to the sink vs. traffic intensity.

the other protocols do not. As the traffic intensity increases, the relative (per-packet) cooperation overhead to form VMIMO links goes down.

It is not surprising that CCP outperforms DCA, as it consumes less energy for every link (as demonstrated in [10] [7]). The per-packet energy consumption for CCP, CMIMO, and MIMO-LEACH are compared in Figure 4. CCP is superior to CMIMO and MIMO-LEACH, as it enjoys full diversity gain of DSTBC and it adapts the number of CNs to the transmission distance, the current nodes' residual energies and the cooperation overhead. As can be observed from the number of MIMO modes in Figure 6, the maximum diversity gain of CCP can be up to $7 \times 8 = 56$. For MIMO-LEACH, its transmission mode is Multiple-Input Single-Output (MISO) or Single-Input Multiple-Output (SIMO), so the maximum diversity order is mostly less than or equal to three ($3 \times 1 = 3$ or $1 \times 3 = 3$). MIMO-LEACH's link diversity gain is less than that of CMIMO $2 \times 2 = 4$. That explains the inferior performance of MIMO-LEACH to CMIMO.

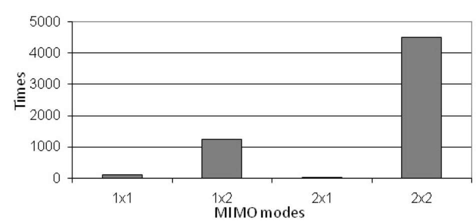


Fig. 5. Histogram of the MIMO modes in CMIMO.

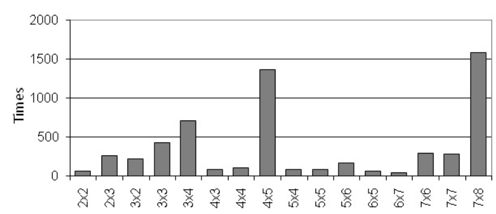


Fig. 6. Histogram of the MIMO modes in CCP.

Figure 5 and 6 show the number for MIMO modes of CMIMO and CCP cases, respectively. As we can see, there is a trend in Figure 5 that the number of 2x2 mode is very high, compared to other modes. That trend suggests that it is more beneficial to have more than 2 CNs. Figure 6 confirms that conjecture, using OCS, the number of CNs can be up to 8 and there are a variety of communication modes, other than 2x2. The maximum diversity gain now is 7x8.

C. Clustering Energy Balancing Routing

To evaluate the proposed C-EBR, we incorporate the mechanism into CMIMO and MIMO-LEACH as well. In Figure 7, we compare network lifetime of CCP, CMIMO and MIMO-LEACH for $\beta = 0.9$ with their counterparts when $\beta = 1$ (energy efficiency routing). As seen, C-EBR improves the network lifetime by about 110%, 103% and 97% on average for MIMO-LEACH, CMIMO and CCP, respectively.

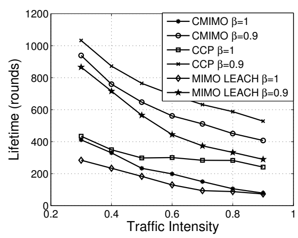


Fig. 7. Network lifetime under different protocols using C-EBR vs. traffic intensity.

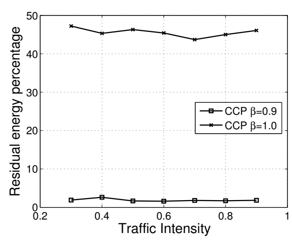


Fig. 8. Average residual energy percentage of nodes that are one hop from the sink.

CCP with both OCS engine and C-EBR, improves the network lifetime by 225% and 170% on average, compared to original MIMO-LEACH and CMIMO protocols. Beside the higher diversity gain, these results can be interpreted as the effect of energy consumption balancing by routing packets on paths that have CNs with higher energy. This fact can be re-confirmed by comparing nodes' remaining energy variance.

Figure 12(a)(b) depict residual energy variance of CMIMO and CCP with and without using C-EBR. For both protocols, C-EBR reduces energy variances. From Figure 12(c) that compares energy variance of CMIMO with CCP while using C-EBR, energy variance of CCP is less than that of CMIMO. It is attributed to the fact that having more CNs not only increases link diversity gain but also spreads out traffic more evenly among nodes in the network, yielding better energy balance.

Figure 11 compares per-packet energy consumption between the energy efficiency routing and C-EBR. As expected, the energy per packet of the former is less than that of C-EBR as it just searches for the least cost path, regardless of the energy status of CNs on that path. By contrast, C-EBR balances residual energy among CNs of the path at the expense of extra cost per packet.

To further investigate the energy balancing effectiveness of C-EBR, in Figure 8, we show the percentage of residual energy averaging over all nodes that can directly reach the sink with power of P_{inter} (one-hop neighbors of the sink) when the lifetime is up. As seen, C-EBR depletes almost completely energy of these nodes (the remaining energy is only about 1% of the initial battery level, $500Js$). However, for energy efficiency routing, the average remaining energy of these nodes is about 45%.

Fig. 9. Histogram of hop count for energy-efficient routing.

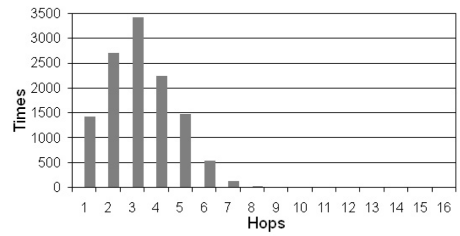


Fig. 10. Histogram of hop count for C-EBR.

We also investigate the number of hops for C-EBR. Figure 10 represents the number of hops of C-EBR which is higher than that of energy efficiency routing (Figure 9). The reason is that sometimes a packet may travel on a "longer" path to the sink to balance energy consumption among nodes. However, the increase in number of hops of C-EBR is not significant, compared with energy efficiency routing, hence it does not much adversely affect the delay of packets.

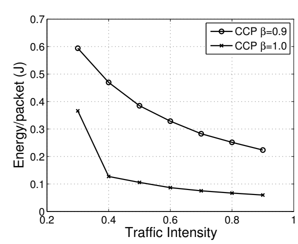


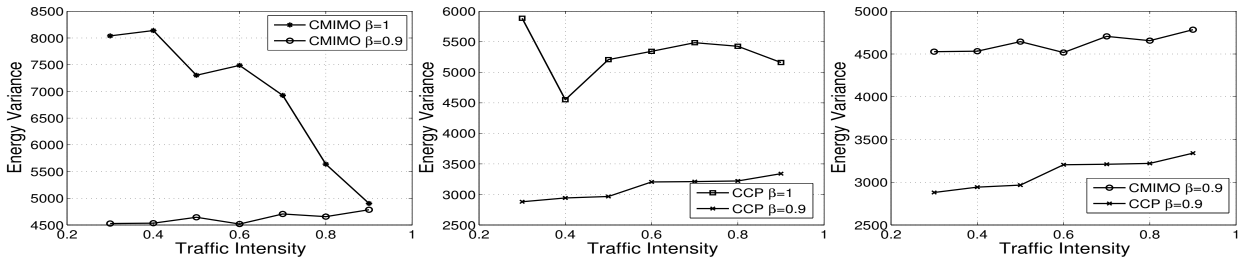
Fig. 11. Energy per packet of energy efficiency routing and C-EBR.

D. CNs Assignment Problem

When not using the proposed CN assignment algorithm, CNs in a cluster are selected based on their distance to the cluster's CH [10]. The rationale behind such criterion of [10] is that having CNs which are closer to their CHs would facilitate their coordination and reduce energy spent on cooperation overhead. Figure 13 shows that using the proposed CN assignment algorithm, CCP significantly reduces the energy variance. Subsequently, as shown in Figure 14, its lifetime is improved by about 80%.

E. Network Lifetime vs. Tradeoff Factor β

Figure 15 shows the network lifetime of CCP against different values of β. The network lifetime of CCP is very consistent with different values of β other than 1. The consistency of C-EBR against various tradeoff factors β confirms that it is important to consider energy of CNs when searching for a path to the sink and the tradeoff factor does not significantly affect the performance as long as it is different from 1 (energy efficiency routing).



(a) CMIMO (b) CCP (c) CMIMO vs CCP

Fig. 12. Energy variance of CMIMO and CCP with and without C-EBR.

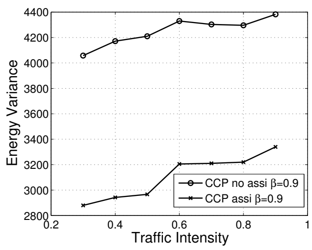


Fig. 13. CCP's energy variance with and without using CN assignment algorithm.

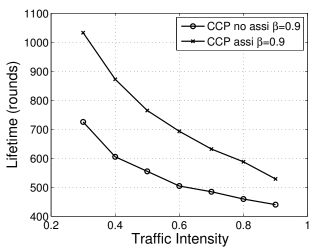


Fig. 14. CCP's lifetime with and without using CN assignment algorithm.

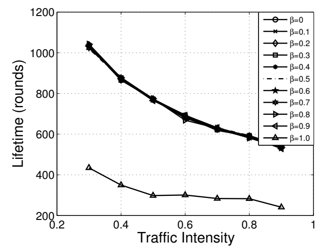


Fig. 15. CCP's lifetime versus tradeoff factor β.

VII. Conclusions and Future Work

In this work, we have developed a cooperative clustering protocol for WSNs. Our protocol takes advantage of VMIMO and enjoys the maximum diversity of DSTBC. The key engine behind CCP is the optimal CN selection algorithm. We showed that the OCS problem is NP-hard and decomposed it into two sub-problems: optimal number of CNs and CN assignment problem using energy balancing approach. The ONC algorithm serves as a framework for protocol designers in deciding the number of CNs per cluster for clustered WSNs. We also proposed the energy-balanced routing mechanism for clustered WSNs, which is applicable to existing clustering protocols. CCP prolongs the network lifetime about three times that of existing cooperative protocols (MIMO-LEACH, CMIMO).

References

[1] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1456–1467, July 1999.

[2] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, Nov. 1996.

[3] M. Dohler, E. Lefranc, and H.Aghvami, "Virtual antenna arrays for future wireless mobile communication systems," in *Proceedings of the ICT Conference, Beijing, China*, June 2002.

[4] S. Jayaweera, "V-BLAST-based virtual MIMO for distributed wireless sensor networks," *IEEE Transactions on Communications*, vol. 55, no. 10, pp. 1867–1872, Oct. 2007.

[5] M. Dohler, E. Lefranc, and H.Aghvami, "Space time block codes for virtual antenna arrays," in *Proceedings of the PIMRC Conference, Lisbon, Portugal*, September 2002.

[6] J. Laneman and G. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2415–2425, Oct. 2003.

[7] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1089–1098, 2004.

[8] W. Chen, Y. Yuan, C. Xu, K. Liu, and Z. Yang, "Virtual MIMO protocol based on clustering for wireless sensor network," in *Proceedings of the 10th IEEE Symposium on Computers and Communications*, June 2005.

[9] Y. Yuan, M. Chen, and T. Kwon, "A novel cluster-based cooperative MIMO scheme for multi-hop wireless sensor networks," *EURASIP J. Wirel. Commun. Netw.*, no. 2, pp. 38–38, 2006.

[10] M. Z. Siam, M. Krunz, and O. Younis, "Energy-efficient clustering/routing for cooperative MIMO operation in sensor networks," in *Proceedings of the IEEE INFOCOM Conference*, April 2009.

[11] Y. Yuan, Z. He, and M. Chen, "Virtual mimo-based cross-layer design for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, pp. 856–864, 2006.

[12] J. P. Y. Zhuang and G. Wu, "Energy-optimal grid-based clustering in wireless microsensor networks," in *Proceedings of the IEEE ICDCS Workshop on Wireless Ad hoc and Sensor Networking (WWASN)*, 2009.

[13] J. P. Yanyan Zhuang and L. Cai, "Minimizing energy consumption with probabilistic distance models in wireless sensor networks," in *Proceedings of the IEEE INFOCOM conference*, 2010.

[14] J.-H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 609–619, 2004.

[15] T. Shu and M. Krunz, "Coverage time optimization for clustered wireless sensor networks: A power-balancing approach," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 202–215, 2010.

[16] S. Basagni, "Distributed clustering for ad hoc networks," in *International Symposium on Parallel Architectures, Algorithms, and Networks*, 1999.

[17] X. Zhang and K. G. Shin, "Dac: Distributed asynchronous cooperation for wireless relay networks," in *Proceedings of the IEEE INFOCOM Conference*, March 2010.

[18] M. Stemm and R. Katz, "Measuring and reducing energy consumption of network interfaces in hand-held devices," *IEICE Transactions on Communications*, 1997.

[19] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, Washington, DC, USA, 2000.

[20] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[21] H. Shin and J. H. Lee, "Exact symbol error probability of orthogonal space-time block codes," in *Proceedings of the IEEE GLOBECOM conference*, Nov. 2002.

[22] W. N. Bailey, *Appell's Hypergeometric Functions of Two Variables, Ch. 9 in Generalised Hypergeometric Series*. Cambridge University Press, 1935.

[23] M. Abramowitz and I. A. E. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, New York, NY, USA, 1972.

[24] J. Pearson, "Computation of hypergeometric functions," *MSc Thesis, University of Oxford, UK*, September 2009.

[25] D. Bertsimas and R. Weismantel, *Optimization Over Integers*, 2005.

[26] Mesquite Software Incorporation, http://www.mesquite.com.