

Energy-Efficient Protocol for Cooperative Networks

Mohamed Elhawary and Zygmunt J. Haas, *Fellow, IEEE*

Abstract—In cooperative networks, transmitting and receiving nodes recruit neighboring nodes to assist in communication. We model a cooperative transmission link in wireless networks as a transmitter cluster and a receiver cluster. We then propose a cooperative communication protocol for establishment of these clusters and for cooperative transmission of data. We derive the upper bound of the capacity of the protocol, and we analyze the end-to-end robustness of the protocol to data-packet loss, along with the tradeoff between energy consumption and error rate. The analysis results are used to compare the energy savings and the end-to-end robustness of our protocol with two non-cooperative schemes, as well as to another cooperative protocol published in the technical literature. The comparison results show that, when nodes are positioned on a grid, there is a reduction in the probability of packet delivery failure by two orders of magnitude for the values of parameters considered. Up to 80% in energy savings can be achieved for a grid topology, while for random node placement our cooperative protocol can save up to 40% in energy consumption relative to the other protocols. The reduction in error rate and the energy savings translate into increased lifetime of cooperative sensor networks.

Index Terms—Clustering, cooperative networks energy-efficient protocols, cooperative transmission, sensor networks.

I. INTRODUCTION

IN WIRELESS sensor networks, nodes have limited energy resources and, consequently, protocols designed for sensor networks should be energy-efficient. One recent technology that allows energy saving is cooperative transmission. In cooperative transmission, multiple nodes simultaneously receive, decode, and retransmit data packets. In this paper, as opposed to previous works, we use a cooperative communication model with multiple nodes on both ends of a hop and with each data packet being transmitted only once per hop.

In our model of cooperative transmission, every node on the path from the source node to the destination node becomes a *cluster head*, with the task of recruiting other nodes in its neighborhood and coordinating their transmissions. Consequently, the classical route from a source node to a sink node is replaced with a multihop cooperative path, and the classical point-to-point communication is replaced with many-to-many

Manuscript received October 08, 2008; revised May 10, 2009; November 28, 2009; and August 25, 2010; accepted August 25, 2010; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Sarkar. Date of publication February 04, 2011; date of current version April 15, 2011. This work was supported in part by NSF Grants ANI-0329905 and CNS-0626751 and by the AFOSR Contract FA9550-09-1-0121/Z806001.

M. Elhawary was with the Computer Science Department, Cornell University, Ithaca, NY 14853 USA. He is now with Google, Inc., Mountain View, CA 94043 USA (e-mail: elhawary@google.com).

Z. J. Haas is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: haas@ece.cornell.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNET.2010.2089803

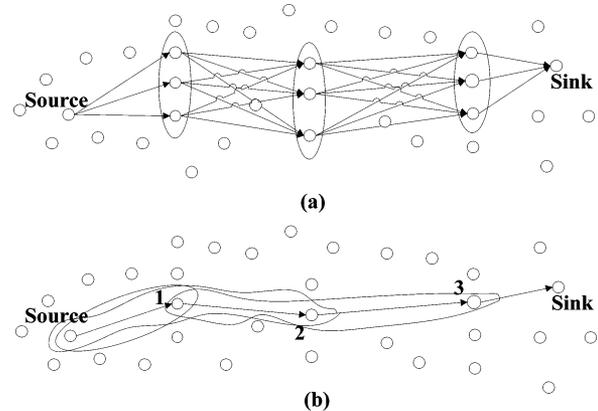


Fig. 1. (a) Our cooperative transmission protocol and (b) the CAN protocol.

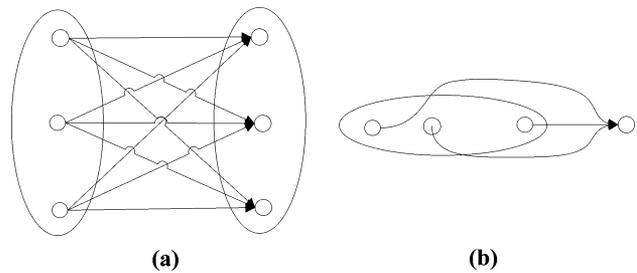


Fig. 2. (a) Our cooperative reception model and (b) the CAN reception model.

cooperative communication. The path can then be described as “having a width,” where the “width” of a path at a particular hop is determined by the number of nodes on each end of a hop. For the example in Fig. 1(a), the width of each intermediate hop is 3. Of course, this “width” does not need to be uniform along a path. Each hop on this path represents communication from many geographically close nodes, called a *sending cluster*, to another cluster of nodes, termed a *receiving cluster*. The nodes in each cluster cooperate in transmission of packets, which propagate along the path from one cluster to the next.

Our model of cooperative transmission for a single hop is further depicted in Fig. 2(a). Every node in the receiving cluster receives from every node in the sending cluster. Sending nodes are synchronized, and the power level of the received signal at a receiving node is the sum of all the signal powers coming from all the sender nodes. This reduces the likelihood of a packet being received in error. We assume that some mechanism for error detection is incorporated into the packet format, so a node that does not receive a packet correctly will not transmit on the next hop in the path.

Our cooperative transmission protocol consists of two phases. In the *routing phase*, the initial path between the source and the sink nodes is discovered as an underlying “one-node-thick” path. Then, the path undergoes a thickening process in the

“*recruiting-and-transmitting*” phase. In this phase, the nodes on the initial path become cluster heads, which recruit additional adjacent nodes from their neighborhood.

Due to the fact that the cluster heads recruit nodes from their immediate neighborhood, the inter-clusters distances are significantly larger than the distances between nodes in the same cluster [not shown in Fig. 2(a)].

Recruiting is done dynamically and per packet as the packet traverses the path. When a packet is received by a cluster head of the receiving cluster, the cluster head initiates the recruiting by the next node on the “one-node-thick” path. Once this recruiting is completed and the receiving cluster is established, the packet is transmitted from the sending cluster to the newly established receiving cluster.

During the routing phase, where the “one-node-thick” path is discovered, information about the energy required for transmission to neighboring nodes is computed. This information is then used for cluster establishment in the “recruiting-and-transmitting” phase by selecting nodes with lowest energy cost. Medium access control is done in the “recruiting-and-transmitting” phase through exchanges of short control packets between the nodes on the “one-node-thick” path and their neighbor nodes.

A key advantage of cooperative transmission is the increase of the received power at the receiving nodes. This decreases the probability of bit error and of packet loss. Alternatively, the sender nodes can use smaller transmission power for the same probability of bit error, thus reducing the energy consumption. One of the goals of this paper is to study the energy savings achieved through cooperation. We also study the increase in the reliability of packet delivery, given some level of cooperation among the nodes. Finally, we also study the capacity of the cooperative transmission protocol.

We compare our cooperative transmission protocol with another cooperative protocol, called Cooperation Along Non-cooperative path (CAN) [1], and with two other non-cooperative schemes: the “*disjoint-paths*” and “*one-path*” schemes. The equivalent of the “one-node-thick” path is called in [1] the “*non-cooperative path*” between the source and the sink nodes and is found first. However, instead of recruiting additional nodes, in CAN, the last m predecessor nodes along the non-cooperative path cooperate to transmit to the next node on the path, as depicted in Fig. 1(b). In this figure, $m = 3$ and the non-cooperative path is source–1–2–3–sink. The source node transmits to node 1; then the source and node 1 transmit to node 2; then the source, node 1, and node 2 transmit to node 3. Finally, nodes 1, 2, and 3 transmit to the sink. Each hop in this protocol consists of cooperative transmission of the last m nodes on the path in order to send the packet to the next node, as is illustrated in Fig. 2(b).

In the disjoint-paths scheme, nodes form a number of disjoint paths from source to sink. The same information is routed independently along the different paths with no coordination between the nodes on the different paths. In the one-path scheme, the “one-node-thick” path is discovered first. Then, each node on the path transmits with power equal to the sum of transmission powers of all the m cooperating nodes in a cluster. The analytical and simulation results of our cooperative transmission protocol are compared throughout the paper to the results of

the CAN protocol, the disjoint-paths scheme, and the one-path scheme.

In summary, in this paper we introduce a new protocol to facilitate cooperative transmission that minimizes the energy consumption and increases the transmission reliability in comparison to the other three schemes. The operation of our protocol is fully distributed in all its phases.

We derive analytical models to evaluate the performance of our protocol in terms of the end-to-end robustness to data loss, the energy consumption, and the capacity. We use the analytical models to compare the performance of our protocol against the other three schemes. Finally, we use simulations to extend our analytical results and to evaluate the performance of our protocol in scenarios not covered by our analytical models.

The rest of the paper is organized as follows. Section II covers the related work. Section III presents our proposed protocol along with the assumptions used to obtain the analytical results. Robustness to data loss is analyzed in Section IV. The analysis of the energy savings is carried out in Section V. The capacity bounds are derived in Section VI. The simulation results are presented in Section VII. Finally, Section VIII concludes our paper.

II. RELATED WORK

The problem of energy-efficient routing in wireless networks that support cooperative transmission was formulated in [1]. In [1], two energy-efficient approximation algorithms are presented for finding a cooperative route in wireless networks. The two algorithms for finding one cooperative route are designed such that each hop consists of multiple sender nodes to one receiver node. One of the algorithms (CAN) is used throughout this paper for performance comparison.

The works in [2]–[5] focus on MAC layer design for networks with cooperative transmission. In [2], when no acknowledgement is received from the destination after timeout, the cooperative nodes, which correctly received the data, retransmit it. Only one cooperative node retransmits at any time, and the other cooperative nodes flush their copy once they hear the retransmission. Hence, this work focuses on reducing the transmission errors, without benefiting from the energy savings of simultaneous transmissions. In [3], high-rate nodes help low-rate nodes by forwarding their transmissions. The work describes how the helper nodes are discovered. Similarly to [2], only one node can cooperate at a time, and simultaneous transmissions are not used, hence the energy savings are not considered. Likewise, in [4] only one node cooperates in forwarding the data. The IEEE 802.11 protocol was extended in [5] to support multiple antennas per node. The works in [6]–[10] use the model with only one helper node at each hop in addition to the sender and the receiver. The model in [11] utilizes multiple nodes to forward the data, but only one node can transmit at any time.

Several good tutorial papers on cooperative transmission have been published (e.g., [12] and [13]). As most of the current works look at the cooperation from the transmitter side only, our paper differs in that our communication model includes groups of cooperating nodes at both sides of the transmission link with the purpose of reduction in energy consumption.

Similar to multiple-input–multiple-output (MIMO) communications, the main gain of cooperative transmission comes from

the fact that there is limited correlation between communications from different transmitters. The increase in the degree of freedom of signal detection decreases the bit error rate [14]. Consequently, the gain of cooperation is similar in nature to what is achieved by MIMO techniques. Of course, there are substantial differences in the environment and in the operation between cooperative transmission and MIMO.

In the MIMO systems, each node is equipped with multiple antennas. Information is transmitted from the sender node by multiple antennas and received by multiple antennas at the receiver node [15], [16]. The close proximity of the antennas at the transmitting nodes and of the antennas at the receiving nodes makes synchronization easier to implement [17]. The ability of nodes to sense the carrier and to measure the interference level can be used to decide on the number of antennas that are employed for transmission.

On the contrary, in cooperative transmission, the synchronization of transmissions of the relatively dispersed cooperating nodes necessitates a more elaborate protocol [17]. A protocol is also required to identify the neighboring nodes as potential cooperators and to make a selection of the cooperating nodes.

Moreover, due to the geographical dispersion of the cooperating nodes, the protocols in cooperative networks need to be distributed in their operation. In [18], a MAC protocol for MIMO systems is described, which is based on centralized cluster architecture. This protocol uses clustering mechanisms like LEACH [19]. Nodes in a cluster cooperate to forward the data to only the next cluster head on the path to the sink. However, the centralized architecture leads to higher energy usage for the cluster maintenance. In contrast, distributed mechanisms are more efficient in the cluster maintenance operation and lack the single-point-of-failure vulnerability. Thus, they may be better suited for sensor or mobile networks.

Finally, the significant cost increase in MIMO to implement multiple antennas at each node would be most often considered impractical in many wireless networks and, in particular, in sensor networks.

The work in [29] proposes and evaluates the performance of a cross-layer framework that uses virtual multiple-input-single-output (MISO) links for MANET and shares some similarity with our paper. However, there are some major differences between the two works. On the physical layer, the architecture of [29] is based on “virtual MISO,” which is also referred to in [29] as “virtual antenna array.” As pointed out in that paper, “nodes [...] simultaneously transmit *and/or* jointly receive appropriately encoded signals.” This model is totally different from our model, where we use MISO system with orthogonal transmissions. On the MAC layer, [29] relies on the knowledge of the neighbors to select the cooperating nodes. To achieve this, [29] assumes that the list of neighbors is obtained by the HELLO messages of the routing protocol. In our paper, we do not assume any knowledge of the neighboring nodes. Rather, we design our own “recruiting” protocol. Furthermore, in [29], the selection of the nodes to cooperate is done randomly, without regard to how useful these nodes could be in improving the cooperative communication. In contrast, in our protocol, selection of cooperating nodes is done based on an elaborate calculation of the costs of the connections. These costs are

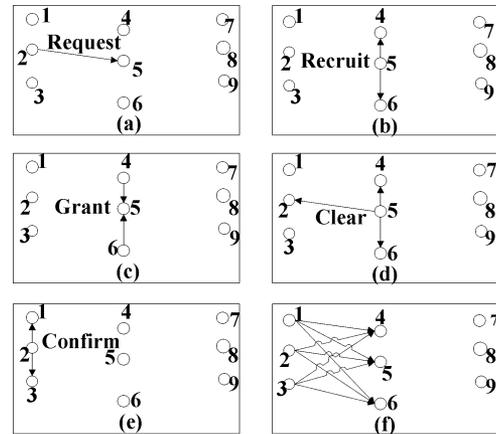


Fig. 3. Example of the recruiting phase operation. (a) Request-to-recruit (RR) packet. (b) Recruit (REC) packet. (c) Grant (GR) packet. (d) Clear (CL) packet. (e) Confirm (CF) packet. (f) Transmission of the data packet.

evaluated not only between the source and the collaborating node, but also between the collaborating nodes and the target nodes in the receiving cluster. Finally, our protocol avoids transmission collisions by reserving the recruiting nodes and preventing them from transmitting during the collaboration.

III. OUR COOPERATIVE PROTOCOL

The routing phase of the protocol, which is responsible for finding a “one-node-thick” route from the source node to the sink node, could be implemented using one of the many previously published routing protocols. For the purpose of performance evaluation, we chose to implement this phase using the Ad hoc On-demand Distance-Vector routing protocol (AODV) [20] with some modifications and with the links’ transmissions energy used as the links’ cost.

The main novelty of our paper—the “recruiting-and-transmitting” phase—is done dynamically per hop, starting from the source node and progressing, hop by hop, as the packet moves along the path to the sink node. Once a data packet is received at a receiving cluster of the previous hop along the path, the receiving cluster now becomes the sending cluster, and the new receiving cluster will start forming. The next node on the “one-node-thick-path” becomes the cluster head of the receiving cluster. The receiving cluster is formed by the cluster head recruiting neighbor nodes through exchange of short control packets. Then, the sending cluster head synchronizes its nodes, at which time the nodes transmit the data packet to the nodes of the receiving cluster.

A. Operation of the “Recruit-and-Transmit” Phase

The example in Fig. 3(a)–(f) demonstrates the operation of the “recruiting-and-transmitting” phase. In the current hop, node 2 is the sending cluster head and has a packet to be sent to node 5. Node 2 sends a request-to-recruit (RR) packet to node 5 [Fig. 3(a)], causing node 5 to start the formation of the receiving cluster, with node 5 as the cluster head. From the routing phase, node 5 knows that the next-hop node is node 8. Node 5 broadcasts to its neighbors a recruit (REC) packet [Fig. 3(b)]. The REC packet contains: the id of the previous node (2), the id of the next node (8), and the maximum time

to respond, denoted as T . Each node that receives the REC packet, which we call *potential recruits* (nodes 4 and 6 in our example), computes the sum of the link costs of the following two links: a link from the sending cluster head to itself (the *receiving link*) and a link from itself to the next node, such as the receiving cluster head or the sink node (the *sending link*). In our example, node 4 computes the sums of the energy costs of the links (2,4) and (4,8), i.e., $C_{2,4} + C_{4,8}$, while node 6 computes the sum of the energy costs of the links (2,6) and (6,8), i.e., $C_{2,6} + C_{6,8}$. A *potential recruit* replies to the REC packet with a grant (GR) packet that contains the computed sum [Fig. 3(c)] after a random backoff time drawn uniformly from $(0, T)$. The GR packets inform the cluster head that the nodes are available to cooperate in receiving on the current hop and in sending on the next hop.

After waiting time T and collecting a number of grants¹, the cluster head (node 5) selects $m - 1$ cooperating nodes with the smallest reported cost to form the receiving cluster of m nodes. (The value of m is protocol-selectable.) If the cluster head node received less than $m - 1$ grants, it forms a smaller receiving cluster with all the nodes that sent the grants. Node 5 then sends a clear (CL) packet [Fig. 3(d)] that contains the ids of the selected cooperating nodes (4 and 6 in our example). The CL packet serves two purposes: 1) it informs the sending cluster head (node 2) that the cluster has been formed; and 2) it informs the *potential recruits* whether they have or have not been chosen to cooperate.

Upon receiving the CL packet from node 5, node 2 sends a confirm (CF) packet to the nodes in its sending cluster (nodes 1 and 3) to synchronize their transmission of the data packet [Fig. 3(e)]. The CF packet contains the waiting-time-to-send and the transmission power level P_t . The transmission power level is the total transmission power (a protocol-selectable parameter) divided by the number of the nodes in the sending cluster. In the case of our example, the value of P_t is divided by 3 (nodes 1–3 are cooperating in sending). After the waiting-time-to-send expires, sending cluster nodes 1–3 send the data packet to the receiving cluster nodes 4–6 [Fig. 3(f)].

B. Calculation of the Cost of Links

The cost of a link from node i to node j , $C_{i,j}$, is calculated by node i as: $C_{i,j} = \frac{(e_{i,j})^\theta}{R_i/R_{\text{avg}}}$, where $e_{i,j}$ is the energy cost of the link, R_i is the residual battery energy of node i , and R_{avg} is the average residual battery energy of the neighbors of node i . Energy cost of a link is the transmission power required for reception at a particular bit error rate. Nodes determine the energy costs of links by listening (or overhearing) transmissions during the routing phase. (See Section III-D for how nodes can calculate the energy costs.) The protocol-selectable parameter θ controls the weight of each factor in the total cost. With this definition of the cost, nodes with small residual battery capacity are less likely to be recruited in this phase.

C. Details of the Control Packets

The format of an RR packet includes: the id of the sender node (node 2 in our example), the id of the receiver node (node 5 in

our example), the sink node id, and the NAV field that contains the estimated transmission time of the data packet. The NAV field serves to indicate when the channel will become available again for other transmissions. The REC packet contains the sender node id, the receiver node id, the id of the next node on the path (node 8 in our example), and the maximum time-to-respond T . The GR packet sent from node j contains the id of the originator of the REC packet and the sum of the link costs of the *receiving link* and the *sending link*. A node can be involved in a single recruiting process at any time; i.e., a node can have only one outstanding GR packet. A node chosen to cooperate cannot be involved in another recruiting process until the transmission of the current data packet is fully completed, i.e., received and sent to the next cluster by the cooperating node.

A CL packet contains the id of the cooperating nodes and an updated value of NAV. Nodes that see their ids in the CL packet form the receiving cluster for this hop and the sending cluster for next hop. Other neighbor nodes that sent GR packets but do not see their ids in the CL packet will not participate in the cluster.

To avoid interference, any node that receives an REC packet, whether cooperating or not, has to wait for the transmission of the data packet to be fully completed before it can get involved in another recruiting process. Similarly, to avoid interference, any node that overhears any of the control packets sent by any other node will not get involved in any recruiting or any transmission operation until the transmission of the data packet is fully completed.

If a data packet was not received at the receiving cluster head node, or was received in error, the packet is deemed lost, and the whole “recruit-and-transmit” phase will restart again.

A timer is associated with every exchange of control packets, so that if a critical control packet is lost, the “recruit-and-transmit” phase will restart again.

D. Assumptions of the Cooperative Models

We calculate the probability of error of the cooperative protocol based on a formula derived in the Appendix. Our model of cooperative communication assumes m transmitters located in the sending cluster and a single receiver located in the receiving cluster. In this sense, the model is similar to the MISO case. Note that each receiver in the receiving cluster creates an independent orthogonal system, which could be implemented through time-, frequency-, or code-orthogonality. In particular, we assume the MISO case of m transmitters over *Rayleigh*-faded channels with known *channel state information* (CSI). The CSI for a link from node i to node j is given in the form of a complex coefficient $h_{i,j}$, which accounts for signal fading. As considered in the Appendix, to minimize bit error rate, *maximal ratio combining* (MRC) should be used, followed by *maximum likelihood* (ML) [27] receiver. To minimize the bit error rate, the transmitted signal should be multiplied by a complex value $\alpha_{i,j}$, such that $\alpha_{i,j} = h_{i,j}^* \cdot (\sqrt{\sum_{\text{all } m \text{ links}} |h_{i,j}|^2})^{-1}$. Under these circumstances, and with known *signal-to-noise ratio* (SNR) at the receiver of SNR, the probability of an error at the receiver is given by

$$P(\text{error}) = f(\text{SNR}, m) = \left(1 + \frac{\text{SNR}}{2}\right)^{-m}. \quad (1)$$

¹We assume here that the processing time of the REC and GR packets is negligible compared to T .

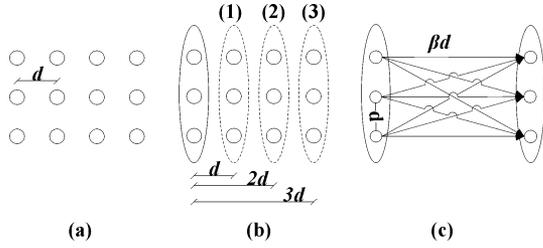


Fig. 4. Grid topology. (a) Placement of nodes. (b) Formation of clusters. (c) Intra- versus inter-cluster distances.

Because of the short intra-cluster distances relative to the inter-cluster distances, we assume that the fading is statistically the same over all the m paths (i.e., $\forall i \neq j, h_{i,j} = h$).

The complex coefficients $h_{i,j}$ account for amplitude attenuation and phase distortion due to multipath fading [22]. In our model, we assumed that the power attenuation due to distance is governed by the $d_{i,j}^{-\gamma}$ law [22], where $d_{i,j}$ is the distance between node i to node j , and γ is the attenuation exponent. In particular, let P_η be the noise power at the receiver, and P_t be the transmitter transmission power measured at nominal distance equal to 1. When a packet is transmitted from node i to node j , the SNR measured at the receiver j is computed as $\text{SNR} = \frac{P_t}{P_\eta} = \frac{P_t}{d_{i,j}^\gamma \cdot P_\eta}$. In other words, to achieve a certain value of SNR, the transmitter needs to transmit with the power of $P_t = \text{SNR} \cdot d_{i,j}^\gamma \cdot P_\eta$ (also see the model in [26]). The bit error probability is then determined by (1). We also assume that for a packet to be successfully received, all the bits in the packet must be successfully received. Finally, we assume that due to short transmission distances, the cooperating nodes can synchronize their transmitted signals at the receiver at the bit level.

We first assume that nodes are positioned in a grid placement as in Fig. 4(a). We present the grid network topology only as an example, where we can evaluate the performance analytically and compare those to the simulation. We assume that the horizontal and vertical distances in the grid are d and that every cluster consists of m nodes. Assuming that the leftmost oval in Fig. 4(b) is the sending cluster, the choice of the receiving cluster can be any of the other ovals in the figure. Thus, depending on the choice of the receiving cluster, the minimal distance between a node in the sending cluster and a node in the receiving cluster (inter-cluster distance) is $d, 2d$, or $3d$ for the three noted choices. To generalize, we assume that the inter-cluster distance is βd as in Fig. 4(c).

For the CAN protocol, we assume that the distance between each two successive nodes on the non-cooperative path is βd and that the last m predecessor nodes cooperate in transmitting. For the disjoint-paths scheme, we assume that the distance between any two adjacent nodes on the route is βd and that there is no interference between transmissions on the different paths. For the one-path scheme, we assume that the distance between any two adjacent nodes on the route is βd .

IV. PROTOCOL ROBUSTNESS

We compute the failure probability that a packet does not reach the sink due to reception error(s) along the path. We then

compare the failure probability of our cooperative transmission protocol to the failure probability using the CAN protocol, the disjoint-paths scheme, and the one-path scheme.

A. Cooperative Transmission Protocol

Let the nodes in the cluster be indexed from 0 to $m-1$. We denote the transmission pattern of nodes in a sending cluster by a binary representation $b_{m-1} \dots b_1 b_0$ according to which node j transmits if $b_j = 1$ and does not transmit if $b_j = 0$. A node does not transmit when it receives a packet in error from the previous hop. We denote the reception pattern of nodes in a receiving cluster by a binary representation $b_{m-1} \dots b_1 b_0$ according to which node j correctly receives the packet if $b_j = 1$ and receives the packet in error if $b_j = 0$. For example, for $m = 4$, the binary representation of 1010 of the sending cluster and the binary representation of 0101 of the receiving cluster means that nodes 1 and 3 in the sending cluster transmit the packet, while in the receiving cluster nodes 0 and 2 correctly receive the packet and nodes 1 and 3 incorrectly receive the packet.

Let g_I^J be the probability that nodes with binary representation $I = u_{m-1} \dots u_1 u_0$ transmit a packet of length L bits to nodes with binary representation $J = b_{m-1} \dots b_1 b_0$ across a single hop, and let SNR_j be the SNR of the received signal at node j . Then

$$\text{BER} = f\left(\text{SNR}_j, \sum_{i=0}^{m-1} u_i\right)$$

$$g_I^J = \prod_{j=0}^{m-1} [(1 - b_j)(1 - (1 - \text{BER})^L) + b_j(1 - \text{BER})^L].$$

Let vector $v(i)$ be the binary representation of integer i . We define: $g_{v(0)}^{v(0)} = 1$; $g_{v(0)}^J = 0$, $J \neq v(0)$.

Let A_{Jk} be the probability that a packet reaches the k th hop to nodes with binary representation J , given that at least one copy reaches hop $k-1$, then

$$A_{Jk} = \sum_{I=1}^{2^m-1} g_{v(I)}^J A_{v(I)k-1}$$

where

$$A_{J0} = \begin{cases} 1, & \text{for } J = v(2^{\lfloor m/2 \rfloor}) \\ 0, & \text{otherwise} \end{cases}$$

Now, let B_{CwR}^h be the probability of failure of a packet to reach any node by the h th hop

$$B_{CwR}^h = \sum_{k=1}^h A_{v(0)k}. \quad (2)$$

B. Disjoint-Paths

The probability of a bit error at a receiver is computed as $f\left(\frac{P_t}{P_\eta d^\gamma \beta^\gamma}, 1\right)$. Then, the probability that a packet of length L bits successfully reaches the sink over one path of h hops is $(1 - f\left(\frac{P_t}{P_\eta d^\gamma \beta^\gamma}, 1\right))^L$. If we let B_{noC}^h be the probability of failure of

a packet to reach any node by the h th hop of the disjoint-paths scheme, then

$$B_{\text{noC}}^h = \left(1 - \left(1 - f \left(\frac{P_t}{P_\eta d^\gamma \beta^\gamma}, 1 \right) \right)^{Lh} \right)^m. \quad (3)$$

C. One-Path

The analysis in this case is similar to the disjoint-paths case, but with one path only and each node transmitting with power of $\sum_{j=1}^m P_t(j)$, where $P_t(j)$ is the transmission power of the j th node. Let B_{One}^h be the probability of failure of a packet to reach the h th node of the one-path scheme, then

$$B_{\text{One}}^h = 1 - \left(1 - f \left(\frac{mP_t}{P_\eta d^\gamma \beta^\gamma}, 1 \right) \right)^{Lh}. \quad (4)$$

D. CAN

Let $X_i = 0$ represent the event that a packet is not received at the i th hop along the non-cooperative path, while $X_i = 1$ is the complementary event. Let B_{CAN}^h be the probability of failure of a packet of length L bits to reach the node at the h th hop

$$\begin{aligned} B_{\text{CAN}}^h &= \Pr(X_h = 0) \\ &= \sum_{I=1}^{2^n-1} [\Pr(X_h = 0 | X_{h-1} = u_0, \dots, X_{h-n} = u_{n-1}) \\ &\quad \times \Pr(X_{h-1} = u_0, \dots, X_{h-n} = u_{n-1})] \end{aligned} \quad (5)$$

where $n = \min(m, h)$. The first term in (5), the probability that a packet is not received at the h th hop given that the last n nodes transmit with binary representation $I = u_{n-1} \dots u_1 u_0$, can be computed by

$$\begin{aligned} \Pr(X_h = 0 | X_{h-1} = u_0, \dots, X_{h-n} = u_{n-1}) \\ = 1 - \left(1 - f \left(\frac{P_t}{P_\eta d^\gamma \beta^\gamma} \sum_{j=1}^n \frac{u_{j-1}}{j^\gamma}, \sum_{i=0}^{n-1} u_i \right) \right)^L. \end{aligned} \quad (6)$$

The second term in (5) is the probability that the packet is received at the hops $h-1$ to $h-n$ with binary representation $I = u_{n-1} \dots u_1 u_0$ and is defined recursively as

$$\begin{aligned} \Pr(X_{h-1} = u_0, \dots, X_{h-n} = u_{n-1}) \\ = \Pr(X_{h-n} = u_{n-1}) \\ \cdot \Pr(X_{h-n+1} = u_{n-2} | X_{h-n} = u_{n-1}) \dots \\ \cdot \Pr(X_{h-1} = u_0 | X_{h-2} = u_1, \dots, X_{h-n} = u_{n-1}). \end{aligned} \quad (7)$$

Next, the definition in (6) is generalized for use in (7) as

$$\begin{aligned} \Pr(X_i = x_i | X_{i-1} = x_{i-1}, \dots, X_{i-j} = x_{i-j}) \\ = \sum_{I=0}^{2^{k-j}-1} \left[\Pr \left(X_i = x_i \mid \begin{array}{l} X_{i-1} = x_{i-1}, \dots, X_{i-j} = x_{i-j}, \\ X_{i-j-1} = u_0, \dots, X_{i-k} = u_{k-j-1} \end{array} \right), \right. \\ \left. \Pr(X_{i-j-1} = u_0, \dots, X_{i-k} = u_{k-j-1}) \right] \end{aligned}$$

where $k = \min(i, m)$ and $I = u_{k-j-1} \dots u_1 u_0$.

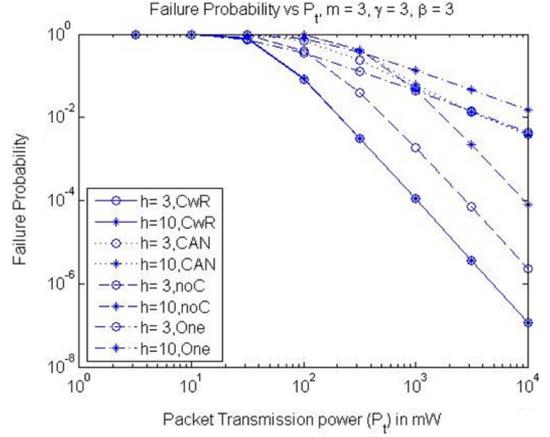


Fig. 5. Failure probability versus P_t , for $m = 3$, $\beta = 3$, and $\gamma = 3$.

Next we plot (2)–(5) for different values of the following parameters: path loss exponent, γ , the number of hops in a route, h , the transmission power measured at nominal distance, P_t , the number of cooperative nodes, m ; and β . We assume that $P_\eta = 10^{-7}$ mW, which corresponds to thermal noise across a 10-k Ω resistance in room temperature and a 1-MHz bandwidth [23]. We also assume that $d = 20$. The failure probability of our cooperative protocol (“CwR”) is plotted in solid lines, the CAN (“CAN”) failure probability is plotted in dotted lines, the disjoint-paths (“noC”) failure probability is plotted in dashed lines, and the one-path (“One”) failure probability is plotted in dashed-dotted lines.

In Fig. 5, we plot the failure probability versus the transmission power P_t , for $h = 3$ and $h = 10$, and for $m = 3$, $\gamma = 3.0$, and $\beta = 3$. The failure probability of our cooperative transmission protocol is lower compared to the three other protocols for both values of h . Note that the difference in the failure probability for the two cases $h = 3$ and $h = 10$ is smaller for the two cooperative transmission protocols compared to the two non-cooperative schemes. The reason is that in the disjoint-paths and the one-path schemes, when a packet is lost on a path, the whole path becomes useless, and the number of copies of the packet gets smaller as the number of hops increase. With cooperative transmission, when a packet is received in error at one receiver node, this decreases the receiving transmission power in the next hop and hence the success probability in the next hop. However, this does not necessarily decrease the number of propagating copies of the packet in the following hops. Once the number of copies increases in the current hop, the success probability increases in the next hop. This suggests that the performance of cooperative transmission is more scalable with the number of hops.

In Fig. 6, we plot the failure probability versus transmission power for different values of m , where $h = 10$, $\gamma = 3.0$, and $\beta = 3$. The failure probability of our protocol is lower compared to the other schemes. Note that as long as the transmission power is large enough, the failure probability decreases with an increase in the number of cooperating nodes or in the number of disjoint-paths. This is so, as there is larger probability that at least one copy of the packet reaches the destination.

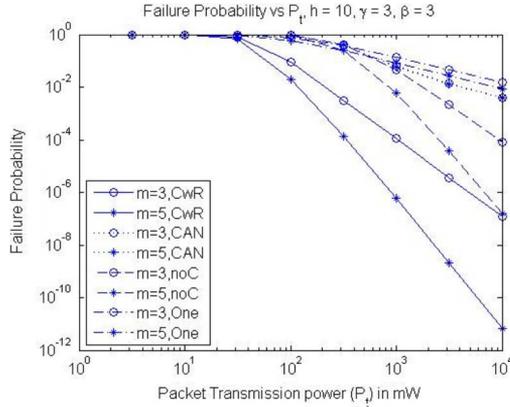


Fig. 6. Failure probability versus P_t ($h = 10$, $\beta = 3$, $\gamma = 3$).

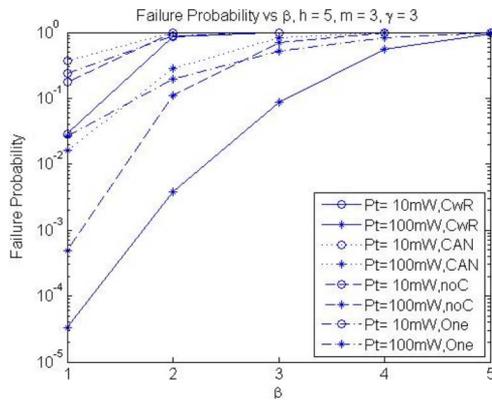


Fig. 7. Failure probability versus β ($h = 5$, $m = 3$, $\gamma = 3$).

Next, we plot in Fig. 7 the failure probability versus β for different transmission power levels, where $m = 3$ and $h = 5$. The failure probability increases as β increases since, as β increases, the received power decreases and the probability failure increases. The failure probability of our protocol is lower compared to the other protocols—by a factor of up to 100 compared to the disjoint-paths scheme, and by a factor of up to 10 000 compared to the CAN protocol and the one-path scheme. The reason for the relatively poor performance of CAN is that the received signal at the receiver node in the CAN protocol consists mainly of the closest sending node's transmitted signal, as the other cooperating nodes are much further away. If a node fails to receive a packet, this has a large effect on the probability of success in the next hop when this node becomes the closest node to the receiver node.

In Fig. 8, we plot the failure probability versus m , for $h = 7, 10$, where we fix the total transmission power $mP_t = 1$ W. The failure probability of our cooperative protocol decreases as we increase the number of nodes until $m = 5$; then, the failure probability increases. Thus, there is an optimal value of m , which minimizes the failure probability. When m increases, there are more receiving nodes, and the probability of receiving more copies of the packet increases. On the other hand, increasing m decreases the transmission power per node P_t as the total transmission power is fixed. This results in smaller received power level at each receiver node and hence smaller probability

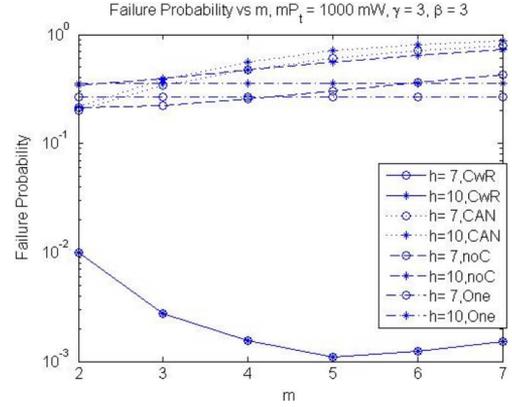


Fig. 8. Failure probability versus m ($mP_t = 1$ W, $\lambda = 3$, $\beta = 3$).

of correctly receiving the packet. Similar behavior, but with different optimal values of m , is observed for different values of γ or the total transmission power. Due to space limitations, further results are omitted.

V. ENERGY CONSUMPTION ANALYSIS

In this section, we analyze the one-hop energy consumption of the transmissions of the control and data packets between two cooperative clusters of nodes, each with m cooperating nodes. We compare the energy consumption of our cooperative protocol to the CAN protocol and the disjoint-paths scheme.

To make the comparison of energy consumption of any two schemes meaningful, the failure probability, as defined in Section IV, needs to be kept equal for the compared schemes. To this end, we assume that the probability of bit error is a function of the SNR of the received signal as derived in the Appendix and as discussed in Section III-D. We label this failure probability as p_f .

For every value of the failure probability p_f , we calculate the needed transmission power of a single node P_t from (2)–(5). We assume that the power consumption for the cooperative protocol is $m^2 \cdot P_t$, as we need m transmissions per hop, with each transmission being of the type m -to-1. For the disjoint-paths and the CAN protocols, we assume that the power consumption is $m \cdot P_t$, and we assume that the power consumption for the one-path protocol is P_t .

Finally, we define the ratios C_{r1} , C_{r2} , and C_{r3} as the ratio of the transmission power needed for our protocol to the transmission power of the disjoint-paths, the CAN protocol, and the one-path protocol, respectively, to achieve the same probability of success ($1 - p_f$). When $C_{ri} < 1$, there is energy saving of our cooperative scheme.

In Fig. 9, we plot the ratios C_{r1} , C_{r2} , and C_{r3} for $p_f = 0.01$ and for $p_f = 0.1$, when h is set to 10 and γ to 3. We vary β and m . In the CAN protocol, the distances between the cooperating nodes and the receiver node are larger than the corresponding distances in our protocol, hence this increases the energy consumption. Consequently, there is an energy saving for our protocol compared to the CAN protocol for all the values of β .

When m is small, regardless of β , there is energy saving for our cooperative protocol over the disjoint-paths scheme. When m is larger than 3, a value of $\beta \geq 2$ achieves energy savings.

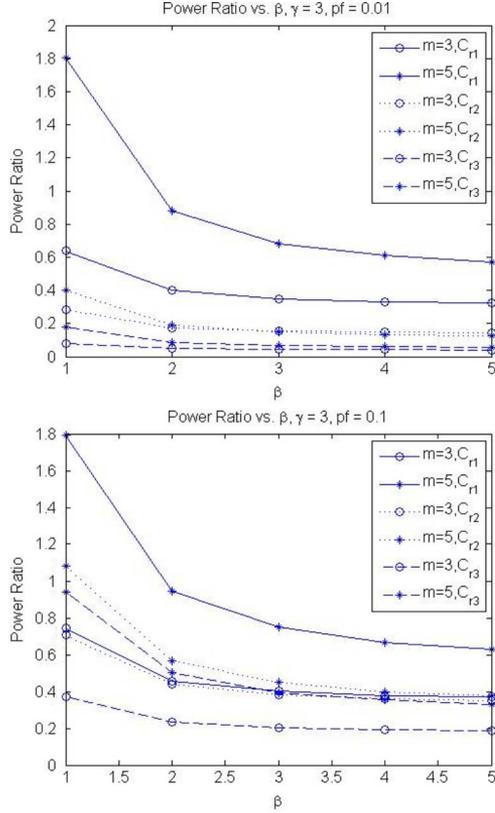


Fig. 9. Power cost ratio versus β , $\gamma = 3$.

Consequently, when the distance between the sending and the receiving clusters is small, one should use a small number of cooperative nodes, such as $m = 3$. When this distance is large, one should use larger m . Our cooperative protocol can save up to 60% in energy over the disjoint-paths scheme and up to 80% in energy over the CAN protocol for large values of β . The amount of savings increases as the failure probability decreases and as β increases.

VI. UPPER BOUND ON CAPACITY

We analyze and compare the capacity upper bounds of a single flow for the three protocols. To compute these bounds, we assume low-load operation, during which a node is in idle state when it receives a packet to transmit. First, we compute the capacity upper bound of one hop on the path, and then we extend the bound to the whole path.

To determine the capacity upper bound for one hop, we divide the number of data bits in the data packet transmitted in one hop by the minimum delay needed to complete this transmission. We assume that each node can either transmit or receive at any time, but not both. Packet transmission on a hop means that no transmission can occur on the previous and the next hops. Thus, the distribution of the number of attempts needed for the data packet to reach the sink is geometric. We assume that the transmission time of the ACK packet is very small compared to the data packet, so we ignore it.

One cycle of the control packets and the data packet transmissions in our cooperative transmission protocol is shown in

Fig. 10. In this figure, τ is the maximum propagation delay between a pair of nodes, where one node is in the sending cluster and the other node is in the receiving cluster. Here, RR, REC, T, CL, CF, and Data stand for the RR packet transmission time, the REC packet transmission time, the maximum waiting time T to collect the GR packets, the CL packet transmission time, the CF packet transmission time, and the data packet transmission time, respectively.

Let W be the channel data rate in bits per second, L_d be the length of data packets in bits, and L_c be the length of control packets in bits. The duration of one cycle of transmission over one hop in our cooperative transmission protocol is

$$\text{Cycle}_{\text{CWR}} = \frac{4L_c}{W} + \frac{L_d}{W} + 5\tau + T.$$

Assume that the maximum waiting time T is equal to the sum of GR packets' transmission times of a number of GRs equal to the average number of neighbor nodes N_b in the network. The one cycle duration is then

$$\text{Cycle}_{\text{CWR}} = \frac{4L_c}{W} + \frac{L_d}{W} + \frac{N_b L_c}{W} + (N_b + 5)\tau.$$

The capacity upper bound of one hop in our cooperative transmission protocol is then defined as: $\text{Tr}_{\text{CWR}} = \frac{L_d}{\text{Cycle}_{\text{CWR}}}$.

Assume that the probability that a packet does not reach the sink, B_{CWR}^h , is calculated as in (2). This results in the average number of attempts being equal to $1/(1 - B_{\text{CWR}}^h)$. The upper bound of the capacity of one flow between the source and the sink in our cooperative transmission protocol is then

$$\text{PT}_{\text{CWR}} = \frac{\text{Tr}_{\text{CWR}}}{3}(1 - B_{\text{CWR}}^h). \quad (8)$$

Next, we compute the upper bound of the disjoint-paths scheme capacity. The two control packets, RTS and CTS, are followed by the data packet transmission. The total cycle duration for one hop of the disjoint-paths scheme, $\text{Cycle}_{\text{noC}}$, is

$$\text{Cycle}_{\text{noC}} = \frac{2L_c + L_d}{W} + 3\tau.$$

The capacity upper bound for one hop in the disjoint-paths scheme is

$$\text{Tr}_{\text{noC}} = \frac{L_d}{\text{Cycle}_{\text{noC}}}.$$

Assume that the probability that a packet does not reach the sink, B_{noC}^h , is calculated as in (3). This results in the average number of attempts being equal to $1/(1 - B_{\text{noC}}^h)$. The upper bound of the capacity of one flow between the source and the sink of the disjoint-paths scheme is

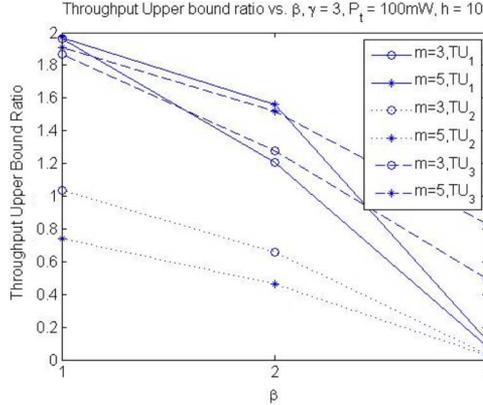
$$\text{PT}_{\text{noC}} = \frac{\text{Tr}_{\text{noC}}}{3}(1 - B_{\text{noC}}^h). \quad (9)$$

Similarly, the upper bound of the capacity of one flow between the source and the sink in the one-path scheme is

$$\text{PT}_{\text{One}} = \frac{\text{Tr}_{\text{noC}}}{3}(1 - B_{\text{One}}^h). \quad (10)$$



Fig. 10. cycle of the “recruiting-and-transmission” phase.

Fig. 11. Capacity upper bounds ratios TU_1 , TU_2 , and TU_3 ($h = 10$, $P_t = 100$ mW).

Next, we compute the CAN capacity upper bound. The two control packets, RTS and CTS, are followed by a broadcast of a control packet for coordination and then the data packet transmission. The total cycle duration for one hop of the CAN protocol, $Cycle_{CAN}$, is

$$Cycle_{CAN} = \frac{3L_c + L_d}{W} + 4\tau.$$

The capacity upper bound for one hop of the CAN protocol is: $Tr_{CAN} = \frac{L_d}{Cycle_{CAN}}$.

Assume now that the probability that a packet does not reach the sink, B_{CAN}^h , is defined as in (5). This results in the number of attempts being equal to $1/(1 - B_{CAN}^h)$. Packet transmission on a hop means that no transmission can occur on the previous m hops and the next hop. The upper bound of the capacity of one flow between the source and the sink of the CAN protocol is

$$PT_{CAN} = \frac{Tr_{CAN}}{m+2}(1 - B_{CAN}^h). \quad (11)$$

Next, we plot the ratios of the upper bounds $TU_1 = PT_{noC}/PT_{CwR}$, $TU_2 = PT_{CAN}/PT_{CwR}$ and $TU_3 = PT_{One}/PT_{CwR}$, where PT_{CwR} , PT_{noC} , PT_{One} , and PT_{CAN} are defined in (8)–(11), respectively, for $L_c = 160$, $L_d = 1024$, $N_b = 6$, $W = 10^6$ b/s, $\gamma = 3$, $P_\eta = 10^{-7}$ [mW], $P_t = 100$ [mW], $d = 20$, and $h = 5$, where we vary m at each value of β . In Fig. 11, the results show that the capacity upper bound of our cooperative transmission protocol stays above the CAN protocol’s upper bound except for small m and for $\beta = 1$. Also, our protocol’s upper bound is larger than the disjoint-paths’ and the one-path’s upper bounds when $\beta \geq 3.0$.

The low upper bound of the CAN protocol results from the fact that the protocol allows only one node to receive in every m successive hops. The CAN protocol was designed to save energy and not to optimize capacity or bandwidth [1]. The increase in the capacity of the disjoint-paths and the one-path schemes over the cooperative transmission at small values of β is due to the

extra number of control packets needed in our protocol, which leads to larger delay. As β increases, the relative overhead of the control packets in our cooperative transmission protocol is reduced, and hence the increase in the delivery rate. This leads to a larger upper bound of our protocol’s capacity compared to the upper bounds of the other protocols. Increasing the number of cooperative nodes is beneficial for large values of β . For small β , the overhead of the control packets is dominant in the capacity upper bound formula. Small number of cooperating nodes reduces this overhead. As β increases, the term corresponding to the data packet successful transmission rate dominates the capacity upper bound formula and larger m increases the capacity bound.

VII. SIMULATION RESULTS

We use simulation to evaluate the performance of our protocol by comparing it to the CAN protocol. We also compare it to the disjoint-paths and the one-path schemes, in which the IEEE 802.11 protocol [24] is used as the MAC-layer protocol. We use the JiST [25] simulation package to compare the three protocols in the multiple-hop scenario.

We run two sets of experiments. In the first set, nodes are positioned on a grid to compare our simulation results to our analytical results. In the second set of experiments, nodes are randomly placed for a more realistic scenario. Unless otherwise stated, we assume that the channel bandwidth is 1 Mb/s, the length of the data packets is 1 kB, γ is 3, the maximum waiting time T is 1.5 ms, and the maximum retry time R_t is 50 ms. We use $\theta = 1$. Each of our simulation results represents an average of 10 random runs, and each simulation run represents a real time of 100 s. Here, we do not ignore interference between different paths as we had done in the analytical results.

In the first set of experiments, we have 7×21 nodes placed in a grid format of 7 rows and 21 columns. The distance between a node and its closest neighbor in the grid is $d = 20$ m. We establish one route of 5 hops between a source node in the first column and the middle row and a sink node in the same row. The sink’s column varies depending on the parameter β . In our protocol, the initial path is set as the middle row, and clusters are formed from nodes in the same column.

In the CAN protocol and the one-path scheme, the cooperative path is set as the middle row. In the disjoint-paths scheme, paths are formed from nodes in the same row, except that the source and the sink nodes are in the middle row. Data packets arrive at the source node following the Poisson distribution with a rate that corresponds to the maximum load that the network can carry.

The metrics that we use for evaluation in the grid placement are: the capacity ratio, the failure probability ratio, and the energy consumption ratio. The capacity measures the total capacity of all the nodes in the network. The capacity ratios, TR_1 , TR_2 , and TR_3 , are the ratios of the capacity of the disjoint-paths scheme, the CAN protocol, and the one-path

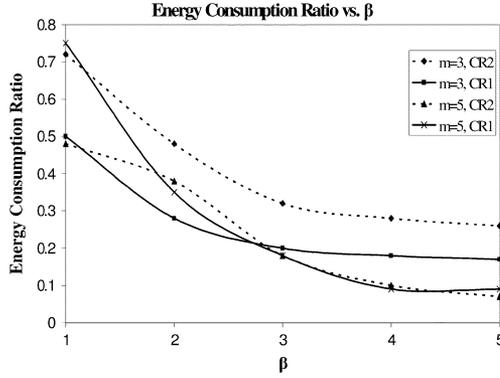


Fig. 12. Energy consumption ratio, $h = 5$, $P_t = 10$ mW.

scheme, respectively, to our cooperative protocol capacity. The failure probability that a data packet does not reach the sink is calculated as the ratio of the number of data packets that do not reach the sink node to the number of data packets that are transmitted by the source node. The failure probability ratios, PR_1, PR_2 , and PR_3 , are the ratios of the failure probability of our cooperative transmission protocol to the failure probability of the disjoint-paths scheme, the CAN protocol, and the one-path scheme, respectively. We assume that each packet transmission (either control or data packet) between a pair of nodes consumes the energy of $SNR(P_\eta + P_{int})d^3L/W$, where SNR is the required *signal-to-noise ratio* to achieve a particular probability of a received bit being error [see (1)], P_η is the noise power, P_{int} is the interference power, d is the distance between the pair of nodes, L is the number of bits in the packet, and W is the channel rate. The total energy consumption measures the sum of the energy of all packet transmissions (control and data packets). The energy consumption ratios, CR_1 and CR_2 , are the ratios of the total energy consumption of our cooperative transmission protocol to the total energy consumption of the disjoint-paths scheme and the CAN protocol, respectively. The results of the grid placement experiments are shown next. The ratios CR_1, TR_1 , and PR_1 are plotted in solid lines; the ratios CR_2, TR_2 , and PR_2 are plotted in dotted lines; and the ratios TR_3 and PR_3 are plotted in dash-dotted lines.

As the energy consumption ratio CR_2 in Fig. 12 is less than 1, our cooperative transmission protocol has lower energy consumption compared to the CAN protocol. When we have a large number of cooperative nodes, $m = 7$, our cooperative transmission protocol has larger energy consumption than the disjoint-paths scheme when β is 1. The reason for that is the high overhead of control packets when m is large and β is small. The results that we get here match the analytical results in Fig. 9, except that here there are also energy savings at $m = 5$, where $\beta = 1$. The energy consumption increases as the interference level increases. The analytical equations assume that the interference power is constant at all time. However, in the simulations of our cooperative transmission protocol, all the transmissions are synchronized and their powers are combined at the receiving nodes, with no other interference present.

We plot the throughput ratios in Fig. 13. Our cooperative transmission protocol has higher capacity compared to the CAN

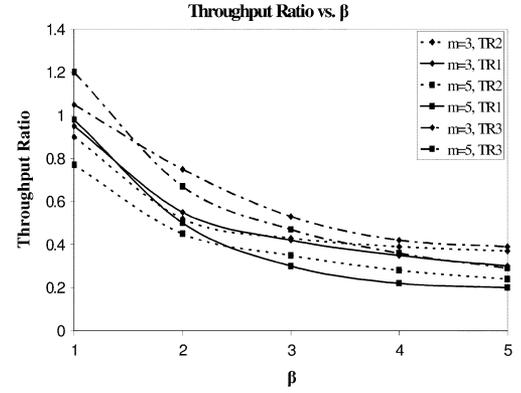


Fig. 13. Throughput ratio, $h = 5$, $P_t = 10$ mW.

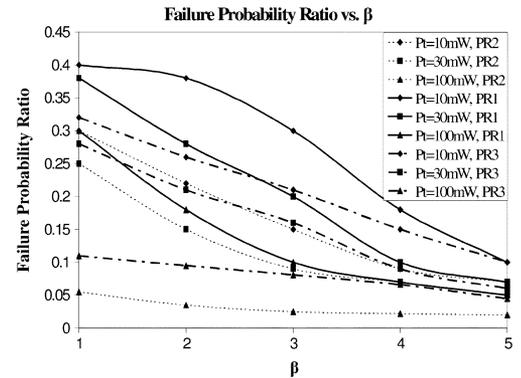


Fig. 14. Failure probability ratio, $h = 5$.

protocol and the disjoint-paths scheme, as indicated by the capacity ratios being smaller than 1. The one-path scheme has larger capacity when β is 1. Our cooperative transmission protocol has two to five times larger capacity depending on the value of m , and our protocol benefits more when β is large.

The reason for our protocol's poor performance at small values of β compared to the non-cooperative schemes is the large overhead of control packets when the number of cooperative nodes is large. These extra control packet transmissions affect the performance for small values of β . However, for large values of β , the cooperation among a large number of nodes improves the performance, and the transmission can be successful over larger distances between the sending and the receiving clusters. The results in Fig. 1 match the analytical results in Fig. 11 for large values of β , but not for small values of β . The reason for the discrepancy is that the analytical evaluation assumes low-load operation, but the simulation results are obtained for the maximum load that the network can carry. In the simulation of the disjoint-paths scheme, contention among packets prevents some parallel transmissions, thus increasing the delay. However, the analytical evaluation assumes that all transmissions in the same hop can be done at the same time.

In Fig. 14, we plot the failure probability ratios using three different values of transmission power: 10, 30, and 100 mW. Our cooperative transmission protocol has lower failure probability compared to the CAN protocol, the disjoint-paths, and the one-path schemes (the ratios are less than 1 for all values of β in

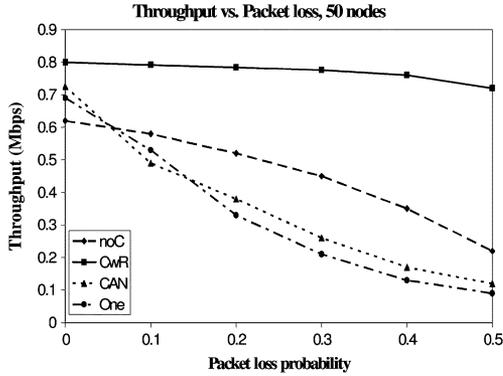


Fig. 15. Effect of packet loss on throughput.

the figure). The success probability of our cooperative transmission protocol can be as much as 50 times better than the CAN protocol and up to 20 times better than the disjoint-paths and the one-path schemes. These results match the analytical results in Section III.

In the second set of experiments, we have 50 nodes that are randomly placed in a $400 \times 400 \text{ m}^2$ area, and the transmission range per node is 150 m, unless otherwise stated. The number of cooperative nodes m is 3, unless otherwise stated. The source and the sink nodes are picked randomly, and each node can be a source or sink for more than one flow.

The parameters we vary are the data packet arrival rate, the number of cooperative nodes, the transmission range, and the probability of packet loss (control and data). The data packets are generated according to Poisson distribution. A 0.2 probability of packet loss means that only 80% of the nodes within transmission range of the sender node will receive the packet (control and data) sent by the sender node. The metrics that we use for evaluation are: the capacity, the average packet delay, and the energy consumption. The capacity measures the total capacity of all the nodes in the network. Lost packets are ignored in the calculation of the average packet delay, which measures the delay from the reception of the data packet at the sender node of the current hop until the reception of the data packet at the receiver node of the same hop. The energy consumption measures the sum of the energy used for all control and data packet transmissions. Our protocol curves (CwR) are plotted in solid lines, the CAN protocol curves (CAN) are plotted in dotted lines, the disjoint-paths curves (NoC) are plotted in dashed lines, and the one-path curves (One) are plotted in dashed-dotted lines.

Fig. 15 shows the capacity versus the data packet loss probability. Each point represents the maximum load that can be pushed through the network. The capacity of our cooperative protocol decreases as the probability of loss increases, but the effect is small compared to the decreasing rate in the capacity of CAN, the one-path, and the disjoint-paths schemes. This suggests that our cooperative protocol is resilient to packet loss.

In the CAN protocol, the probability of the success of packet reception relies heavily on the closest sender node within the cooperating nodes. Failure to receive a packet results in a large reduction in the success probability on the next hop. CAN does better than disjoint paths in energy savings, but is worse in capacity, as transmission from the furthest node in the sending

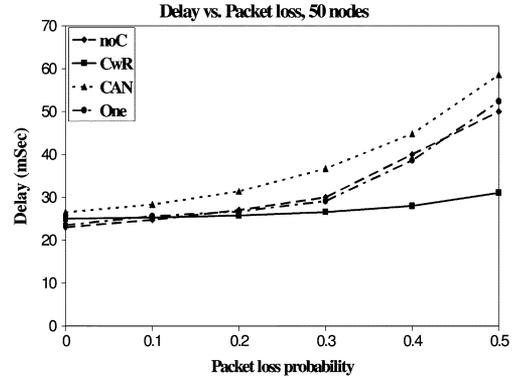


Fig. 16. Effect of packet loss on delay.

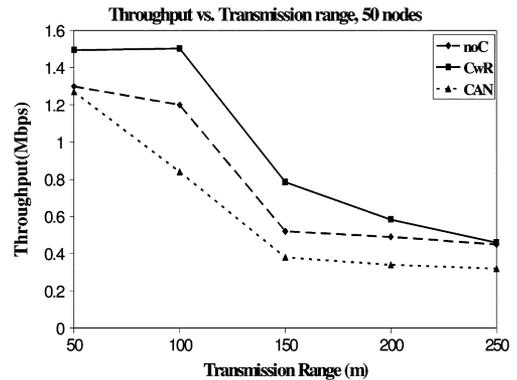


Fig. 17. Effect of transmission power on throughput.

cluster to the receiver node blocks a large number of nodes in the network from transmitting. Also, a data packet loss in the disjoint-paths and the one-path schemes results in a retransmission attempt of the packet, which increases the delay of this packet and the delay of the other packets in contention with this packet. This is the reason for the large delay of the disjoint-paths and the one-path schemes in Fig. 16, which depicts the effect of the packet loss probability on the average packet delay. In the CAN protocol, a node is committed to cooperate in m successive hops. This node cannot transmit other packets that arrive during this commitment. This is the reason for the large delay of the CAN protocol. When the packet loss probability increases, the average delay in our cooperative transmission protocol increases, but at a slower rate than the CAN protocol, the one-path scheme, and the disjoint-paths scheme. Note that the average delay of the two non-cooperative schemes is smaller than the delay of our protocol when there is no packet loss. This is due to the delay in waiting for the grant packets and the extra control packets in our cooperative protocol.

Next, in Figs. 17 and 18, we compare our cooperative transmission protocol to the CAN protocol and the disjoint-paths scheme at different power transmission levels. We vary the power transmission level, so that it corresponds to transmission ranges that vary between 50 and 250 m, and we fix the packet loss probability at 0.2. Each point represents the maximum load that can be pushed through the network. The effect of the transmission range on the capacity is shown in Fig. 17. The results show that our cooperative protocol has larger capacity, until the

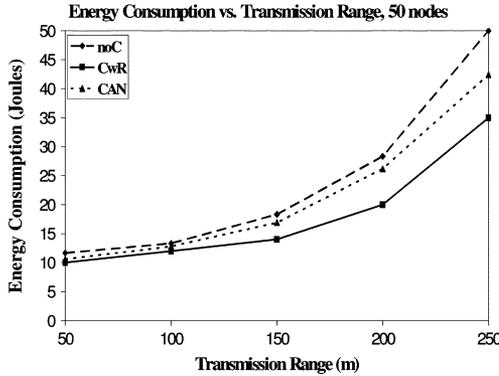


Fig. 18. Effect of transmission power on energy consumption.

increase in power transmission level transforms the network from multihop to a 1- or 2-hop network. As the transmission range increases, the average number of hops in a path from source to sink decreases. When the number of hops decreases, the difference in the failure probability between our protocol and the disjoint-paths scheme decreases. This leads to a smaller difference in the difference in the capacity between the two schemes. When the network becomes a 1- or 2-hop network, there is no difference in the capacity between the two schemes.

The capacity of the CAN protocol degrades with an increase in the transmission range, as a transmission on one hop blocks a large number of nodes from transmitting other packets, and hence the network can only carry lower load. In the CAN protocol, failure to receive a packet results in a large reduction in the success probability on the next hop. The disjoint-paths scheme has larger energy consumption, as demonstrated in Fig. 18, which shows the effect of the transmission range on the total energy consumption. Here, we sum the energy consumption for all packets transmitted (control and data packets). Our cooperative transmission protocol saves between 6% and 20% of the energy consumption compared to the CAN protocol and between 10% and 40% of the energy consumption compared to the disjoint-paths scheme. As the transmission range increases, the contention increases and the noise power increases. This increases the energy consumption. The elevated contention increases the retransmission of control and data packets, which, in turn, increases the total energy consumption.

In Fig. 19, we plot the capacity of all protocols versus the load in the network. At a small load, the delay needed to recruit the cooperative nodes affects the capacity of our cooperative protocol. As the load increases, the interference increases, and the number of concurrent transmissions on the different paths diminishes. The one-path scheme is not resilient to packet loss, and the maximum carried load is affected by the packet loss rate. The maximum load of the CAN protocol is limited by the fact that transmission of a packet blocks several nodes from transmission. Also, the CAN protocol is affected more by packet loss. Consequently, due to the above limitations, the capacity of the cooperative protocol is higher than that of the other three schemes.

Next, in Fig. 20, we study the effect of the number of cooperative nodes on the performance of our cooperative protocol. We fix the packet loss probability at 0.2. We plot the capacity versus

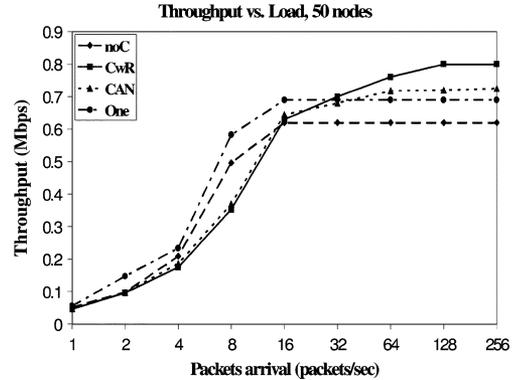


Fig. 19. Throughput versus load.

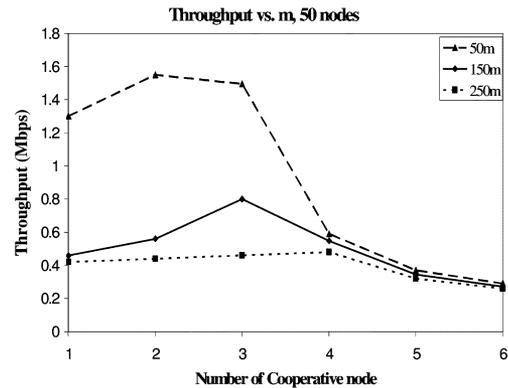


Fig. 20. Effect of the number of cooperative nodes.

the number of cooperative nodes for three different transmission ranges: 50, 150, and 200 m. Each point in the figure represents the maximum load that can be pushed through the network. There is a tradeoff between the delay of recruiting the cooperative neighbors and the robustness to packet loss. At small m , the delay is small, but the effect of packet loss is more significant on the performance of our cooperative transmission protocol. Losing one copy of the data packet out of two copies when $m = 2$ has a more pronounced effect on the probability of success to reach the sink, as compared with losing one copy out of five copies when $m = 5$. At large m , the delay is larger. However, as there are many nodes that cooperate in one transmission hop, the network is more resilient to transmission errors. Furthermore, none of these nodes can be involved in other parallel transmissions. The largest capacity is achieved at $m = 2$ for a transmission range = 50 m, at $m = 3$ for a transmission range = 150 m, and at $m = 4$ for a transmission range = 250 m. At these points in the figure, the balance of this tradeoff between the delay and resilience to packet loss is reached and the capacity is maximized.

VIII. SUMMARY AND CONCLUDING REMARKS

In this paper, we evaluated the performance of cooperative transmission, where nodes in a sending cluster are synchronized to communicate a packet to nodes in a receiving cluster. In our communication model, the power of the received signal at each node of the receiving cluster is a sum of the powers of the transmitted independent signals of the nodes in the sending

cluster. The increased power of the received signal, vis-à-vis the traditional single-node-to-single-node communication, leads to overall saving in network energy and to end-to-end robustness to data loss.

We proposed an energy-efficient cooperative protocol, and we analyzed the robustness of the protocol to data packet loss. When the nodes are placed on a grid and as compared to the disjoint-paths scheme, we showed that our cooperative protocol reduces the probability of failure to deliver a packet to destination by a factor of up to 100, depending on the values of considered parameters. Similarly, compared to the CAN protocol and to the one-path scheme, this reduction amounts to a factor of up to 10 000. Our study also analyzed the capacity upper bound of our protocol, showing improvement over the corresponding values of the other three protocols.

The total energy consumption was analytically computed, illustrating substantial energy savings. For example, when nodes are positioned on a grid, the energy savings of our cooperative protocol over the CAN protocol is up to 80%.

The size of the clusters, m , should be relatively small, when the inter-cluster distance β is small, with the optimal value of m increasing with β . For scenarios that are not covered by our theoretical analysis, we used simulation to evaluate and compare the protocols. For random placement of nodes, the simulation results show that our cooperative transmission protocol saves up to 20% of energy compared to the CAN protocol and up to 40% of energy compared with the disjoint-paths and the one-path scheme.

Overall, the study demonstrates that the energy savings of our protocol, relative to the other schemes, do not substantially decrease even when the data packet loss approaches 50%. Our protocol also supports larger capacity and lower delay under high-load conditions, as compared to the CAN protocol, the one-path scheme, and the disjoint-paths scheme.

APPENDIX

COOPERATIVE COMMUNICATION MODEL

Our cooperative communication model assumes m transmitters located in the transmitting cluster and a single receiver located in the receiving cluster. In this sense, the model is similar to the MISO case. The following analysis has been inspired by [21].

We start from representation of the transmitted symbols. We label as x_i the symbol transmitted by the i th transmitter. For *quadrature phase-shift key* (QPSK) modulation, a symbol consists of two bits, and the four possible complex values of the symbol could be represented as: $x = \pm\sqrt{\varepsilon_b} \pm j\sqrt{\varepsilon_b}$.

We model the received signal y as a superposition of the m transmitted signals: $y = \sum_{i=1}^m h_i x_i + n$, where h_i is the complex fading between the i th transmitter and the receiver, and n represents the uncorrelated AWGN with zero mean and variance of σ^2 per signal dimension. As in MIMO, we assume that the fading is mutually independent; i.e., $\forall i \neq j$, h_i is independent of h_j .

We now define the SNR as: $\text{SNR} = E[|y|^2]/E[|n|^2]$, where y and n are as defined previously. For our case, if all the signals

were to be transmitted with the same power, $\forall i, E[|x_i|^2] = E[|x|^2]$, then the SNR at the receiver from the m transmitters would be

$$\begin{aligned} \text{SNR}_m(h_1, h_2, \dots, h_m) &= E \left[\left| x \sum_{i=1}^m h_i \right|^2 \right] / E[|n|^2] \\ &= \text{SNR}_1(1) \cdot E \left[\left| \sum_{i=1}^m h_i \right|^2 \right] \end{aligned}$$

where $\text{SNR}_1(1)$ is the unfaded received SNR of a link with a single transmitter; i.e., $\text{SNR}_1(1) = \text{SNR}_{m=1}(h_1 = 1) = \frac{E[|x|^2]}{E[|n|^2]} = \frac{\varepsilon_b}{\sigma^2}$. (Note that for QPSK: $E[|x|^2] = 2\varepsilon_b$ and $E[|n|^2] = 2\sigma^2$.)

To allow the transmitted signals to be of different power, we multiply the transmitted signals by complex numbers α_i for $i = 1, 2, \dots, m$, where $\sum_{i=1}^m |\alpha_i|^2 = 1$ to preserve the total transmission power. Then, the received signal would be $y = x \cdot \sum_{i=1}^m h_i \alpha_i + n$, and the corresponding SNR would be

$$\begin{aligned} \text{SNR}_m(h_1, h_2, \dots, h_m) &= E \left[\left| x \sum_{i=1}^m h_i \alpha_i \right|^2 \right] / E[|n|^2] \\ &= \text{SNR}_1(1) \cdot \left| \sum_{i=1}^m h_i \alpha_i \right|^2. \end{aligned}$$

Using the Cauchy–Schwartz inequality, the maximum $\text{SNR}_m(h_1, h_2, \dots, h_m)$ is achieved with $\alpha_i = \frac{h_i^*}{\sqrt{\sum_{j=1}^m |h_j|^2}}$.

This is the equivalent of *maximum ratio combining* in the single-input–multiple-output (SIMO) case. With the above condition, the maximized SNR is

$$\text{SNR}_m(h_1, h_2, \dots, h_m) = \text{SNR}_1(1) \cdot \sum_{i=1}^m |h_i|^2.$$

Using the Maximum Likelihood receiver, the probability of QPSK error² is [27]

$$\begin{aligned} P(\text{error}|h_1, h_2, \dots, h_m) &\leq \exp \left(-\frac{\text{SNR}_m(h_1, h_2, \dots, h_m)}{2} \right) \\ &= \exp \left(-\frac{\text{SNR}_1(1) \sum_{i=1}^m |h_i|^2}{2} \right). \end{aligned}$$

Although this expression provides an upper bound on the error probability, this bound is tight for medium-to-high values of SNR [27], which are values of interest in this paper.

Next, we label $\rho_i = |h_i|$, $i = 1, 2, \dots, m$. We note that the ρ_i 's are *Rayleigh*-distributed and mutually independent; i.e., $\forall i \neq j$, ρ_i is independent of ρ_j . Consequently, the joint *probability density function* is a product of the individual probability density functions: $f_{\rho}(\rho_1, \rho_2, \dots, \rho_m) = \prod_{i=1}^m 2\rho_i \cdot \exp(-\rho_i^2)$. We can now

²Note that $Q(x) < \exp(-x^2/2)$

remove the condition from $P(\text{error}|h_1, h_2, \dots, h_m)$ by averaging over $f_{\rho}(\rho_1, \rho_2, \dots, \rho_m)$, which yields

$$\begin{aligned} P(\text{error}) &= E(P(\text{error}|h_1, h_2, \dots, h_m)) \\ &\leq \int_{\rho_1=0}^{\infty} \cdots \int_{\rho_m=0}^{\infty} \exp\left(-\frac{\text{SNR}_1(1) \sum_{i=1}^m \rho_i^2}{2}\right) \\ &\quad \cdot \prod_{i=1}^m 2\rho_i \cdot \exp(-\rho_i^2) \cdot d\rho_1 \cdots d\rho_m \\ &= \prod_{i=1}^m \int_{\rho_i=0}^{\infty} 2\rho_i \cdot \exp(-\rho_i^2 (1 + \text{SNR}_m(1)/2)) \cdot d\rho_i \\ &= \prod_{i=1}^m \frac{1}{(1 + \text{SNR}_1(1)/2)} = \frac{1}{(1 + \text{SNR}_1(1)/2)^m}. \end{aligned}$$

Again, the upper bound is quite tight. Thus, we use as our cooperative communication model the following error probability function³: $P(\text{error}) = (1 + \text{SNR}_1(1)/2)^{-m}$.

REFERENCES

- [1] A. Khandani, J. Abounadi, E. Modiano, and L. Zheng, "Cooperative routing in static wireless networks," *IEEE Trans. Commun.*, vol. 55, no. 11, pp. 2185–2192, Nov. 2007.
- [2] N. Shankar, C. Chun-Ting, and M. Ghosh, "Cooperative communication MAC (CMAC)—A new MAC protocol for next generation wireless LANs," in *Proc. IEEE Int. Conf. Wireless Netw., Commun., Mobile Comput.*, Maui, HI, Jul. 2005, vol. 1, pp. 1–6.
- [3] T. Korakis, S. Narayanan, A. Bagri, and S. Panwar, "Implementing a cooperative MAC protocol for wireless LANs," in *Proc. IEEE ICC*, Istanbul, Turkey, Jun. 2006, vol. 10, pp. 4805–4810.
- [4] C. Chou, J. Yang, and D. Wang, "Cooperative MAC protocol with automatic relay selection in distributed wireless networks," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, White Plains, NY, Mar. 2007, pp. 526–531.
- [5] J. Mirkovic, G. Orfanos, H. Reumerman, and D. Denteneer, "A MAC protocol for MIMO based IEEE 802.11 wireless local area networks," in *Proc. IEEE WCNC*, Hong Kong, Mar. 2007, pp. 2131–2136.
- [6] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—Part I: System description," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1927–1938, Nov. 2003.
- [7] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—Part II: Implementation aspects and performance analysis," *IEEE Trans. Commun.*, vol. 51, no. 11, pp. 1939–1948, Nov. 2003.
- [8] J. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec. 2004.
- [9] A. Stefanov and E. Erkip, "Cooperative information transmission in wireless networks," in *Proc. Asian-Eur. Workshop Inf. Theory*, Breisach, Germany, Jun. 2002, pp. 90–93.
- [10] T. Hunter and A. Nosratinia, "Diversity through coded cooperation," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 283–289, Feb. 2006.
- [11] J. Laneman and G. Wornell, "Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2415–2425, Oct. 2003.
- [12] P. Herhold, E. Zimmermann, and G. Fettweis, "Cooperative multi-hop transmission in wireless networks," *Comput. Netw.*, vol. 49, no. 3, pp. 299–324, Oct. 2005.
- [13] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 74–80, Oct. 2004.
- [14] H. Shen and S. Kalyanaraman, "Asynchronous cooperative MIMO communication," in *Proc. IEEE WiOpt*, Limassol, Cyprus, Apr. 2007, pp. 1–9.
- [15] D. Hoang and R. Iltis, "An efficient MAC protocol for MIMO-OFDM ad hoc networks," in *Proc. IEEE Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Oct. 2006, pp. 814–818.
- [16] K. Sundaresan, R. Sivakumar, M. Ingram, and T. Chang, "A fair medium access control protocol for ad-hoc networks with MIMO links," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, vol. 4, pp. 2559–2570.
- [17] J. E. Kleider, G. Maalouli, and X. Ma, "Timing synchronization in distributed mobile MISO Rayleigh fading channels," in *Proc. IEEE MILCOM*, Orlando, FL, Oct. 2007, pp. 1–7.
- [18] Y. Yuan, M. Chen, and T. Kwon, "A novel cluster-based cooperative MIMO scheme for multi-hop wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2006, no. 2, pp. 38–38, Apr. 2006.
- [19] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in *Proc. IEEE Int. Conf. Syst. Sci.*, Jan. 2000.
- [20] C. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. IEEE WMCSA*, New Orleans, LA, Feb. 1999, pp. 90–100.
- [21] I. Hen, "MIMO architecture for wireless communication," *Intel Technol. J.*, vol. 10, no. 2, pp. 157–165, May 2006.
- [22] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2001.
- [23] J. Proakis, *Digital Communications*. New York: McGraw-Hill, 1995.
- [24] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE 802.11 Working Group, 1997.
- [25] "JIST," Cornell University, Ithaca, NY, 2005 [Online]. Available: <http://jist.ece.cornell.edu/>
- [26] Q. Dong, S. Banerjee, M. Adler, and A. Misra, "Minimum energy reliable paths using unreliable wireless links," in *Proc. ACM MobiHoc*, Urbana-Champaign, IL, May 2005, pp. 449–459.
- [27] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
- [28] J. G. Proakis and M. Salehi, *Communication Systems Engineering*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [29] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos, P. V. Krishnamurthy, and O. Ercetin, "A cross-layer framework for exploiting virtual MISO links in mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 579–594, Jun. 2007.



Mohamed Elhawary received the B.Sc. and M.Sc. degrees from the University of Alexandria, Alexandria, Egypt, in 2000 and in 2003, respectively, and the M.Sc. degree from Cornell University, Ithaca, NY, in 2008, all in computer science.

He joined Google, Mountain View, CA, in October 2009. His research interests are in the areas of wireless communication. He has been studying MAC-layer protocols in cooperative networks and multichannel wireless sensor networks.



Zygmunt J. Haas (F'07) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1979, 1985, and 1988, respectively.

He then joined the Network Research Department, AT&T Bell Laboratories, Holmdel, NJ. There, he pursued research on mobility management, wireless networks, wireless communications, fast protocols, optical networks, and optical switching. In August 1995, he joined the faculty of the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, where he is now a Professor. He is an author of numerous technical papers and holds 18 patents. His interests include mobile and wireless communication and networks, biologically inspired systems, and performance evaluation of large and complex systems.

Dr. Haas has organized several workshops, delivered numerous tutorials at major IEEE and ACM conferences, and served as an Editor for several journals and magazines, including the IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, *IEEE Communications Magazine*, and *Wireless Networks*. He has been a Guest Editor of several IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS issues and served as a Chair of the IEEE Technical Committee on Personal Communications. For more information, see <http://wnl.ece.cornell.edu>.

³Note that we used the approximation that $P_b(\text{error}) \approx 1/2 P_S(\text{error})$ [28].