

Energy-Efficient Cooperative Protocol for Wireless Networks

Sagiraju Srinadh Raju¹, K.Ramesh²

¹ Post Graduate Student, Kakinada Institute of Engineering and Technology

²Assistant Professor, Kakinada Institute of Engineering and Technology

Summary— In cooperative networks, transmitting and receiving nodes recruit neighboring nodes to help in communication. We model a cooperative transmission link in wireless networks and then it forms a cluster at transmitter end and a receiver end. In this paper, we propose a cooperative communication protocol for formation of these clusters for cooperative transmission of data. This paper proposes a new reliable and energy efficient cooperative protocol, and we analyzed the 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, such as one non-cooperative named as disjoint-paths and one another cooperative scheme named as Cooperation along Non-cooperative path (CAN). The reduction in error rate and the energy savings translate into increased lifetime of cooperative sensor networks.

Keywords— clustering, cooperative networks energy-efficient protocols, cooperative transmission, Routing, sensor networks.

1. INTRODUCTION

In wireless ad-hoc networks, nodes spend most of their energy in transmitting data, but in many applications these nodes are small and have limited energy supply such as in wireless sensor networks. Much work has been done in this area to take down the total required transmit power going from a source node to a destination node by choosing a transmission scheme that have need the minimum amount of transmit power. One such group of techniques is specifying to as cooperation routing which is based on cooperation among neighboring nodes. Examine variant cooperative routing algorithms using the wireless broadcast advantage and relays.

In multi-hop wireless sensor networks, the information from the source to the destination is relayed by intermediate nodes. Traditionally, the routing protocols choose a path – a sequence of

nodes between the source and the destination -and then forward packets along the path. To encounter the link level packet loss and to avoid significant end-to-end throughput reduction, networks use link-level retransmissions. However, due to correlation of errors in retransmitted packets exclusively in wireless networks, retransmission is often quite worthless. It could also be quite faulty, leading to significant misuse of network capacity and energy, and considerably enlarging the end-to-end delay. Thus, in numerous occasions, such as real-time traffic for example, link-level retransmission may not be the right way for increasing the end-to-end transmission reliability.

In cooperative communication, clustering could be used to group nodes which are positioned close to each other. The massive classification of the nodes in wireless sensor network accommodate an effective scenario for node clustering. All nodes in a cluster cooperate to transmit and receive packets to/from other cooperative clusters. Compared with other schemes, the cluster-based approach reduces the complication of resource management of the cooperation among the cluster's nodes. Fig. 1 shows an example of cooperative transmissions from the source to the destination through multiple clusters, where packets are relayed from a cluster to a cluster. Our model of cooperative transmission for a single hop is further illustrated in Fig. 2. 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 depresses the likelihood of a packet being received in error. We consider that some system 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.

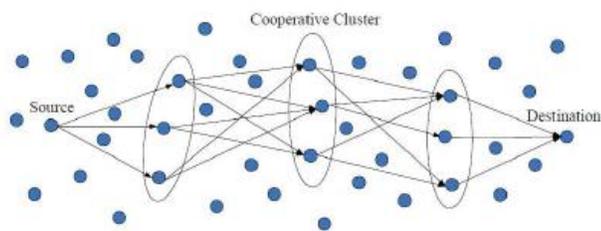


Fig.1: Example of cooperative clusters in a wireless network

Our model of cooperative transmission for a single hop is further shown in Fig. 2. 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.

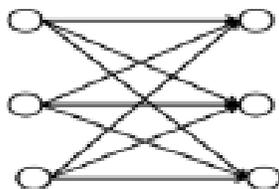


Fig 2: Proposed cooperative model

In Existing Techniques, Two energy-efficient approximation algorithms are suggested 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. Existing methods focus on MAC layer design for networks with cooperative transmission. 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 the multiple-input–multiple-output (MIMO) systems, each node is provided with multiple antennas. Information is transmitted from the sender node by multiple antennas and received by multiple antennas at the receiver node. The close concurrence

of the antennas at the transmitting nodes and of the antennas at the receiving nodes makes synchronization easier to setup. 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.

In this paper we propose 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 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.

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.

2. RELATED WORK

Two energy-efficient approximation algorithms are shown 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.

2.1 CAN Protocol

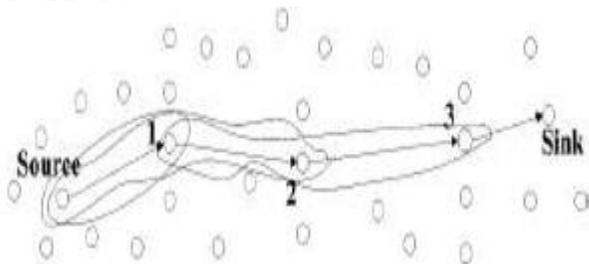


Fig 3(a): CAN Protocol

In CAN protocol, instead of sending once per hop, the sender node transmit data packets to all the nodes along the path. In the Fig.3 (a).The number of nodes between source and destination is $m=3$ 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. Likewise, the source node send packets to all hops in the path, then all hops sends to the receiver node.

Each hop in this protocol consists of cooperative transmission with the last m nodes on the path in order to send the packet to the next node, as illustrated in Fig. 3(b).



Fig 3(b): CAN Reception model

2.2 Disjoint Path Scheme

In the disjoint-paths scheme, nodes form a number of disjoint paths from source to sink. The same message is routed individually along the different paths with no coordination between the nodes on the different paths.

2.3 One-Path Scheme:

In the one-path scheme, the “one-node-thick” path is finding and established first. Then, each node on the path transmits with power equal to the sum of transmission powers of all the 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-path and One- Path schemes. In

summary, in this paper we originate a new protocol to facilitate cooperative transmission that decreases the energy consumption and increases the transmission reliability in comparison to the other two schemes. The operation of our protocol is fully distributed in all its phases. We derive an analytical model to calculate the performance of our protocol in terms of the end-to-end robustness to data loss, and the energy consumption.

3. DESIGN COOPERATIVE PROTOCOL

It consists of two phases: 1. Routing Phase, 2. Recruit & Transmit Phase. The routing phase of the protocol, which is responsible for determining an initial route from the source node to the sink node, could be carried out using one of the many previously published routing protocols. 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 in the next phase. The next node on the routing path becomes the cluster head of the receiving cluster. The receiving cluster is formed by the cluster head recruiting neighbor nodes through replacement of short control packets. Then, the sending cluster head synchronizes its nodes, at which time the nodes transmit the data packet to all nodes of the receiving cluster.

3.1 Routing Phase

The routing phase of the protocol, which determines the initial path from source to sink. In example, upon receiving the packet from node 5, node 2 sends a confirm packet to the nodes in its sending cluster (nodes 1 and 3) to synchronize their transmission of the data packet. The packet contains the waiting-time-to-send and the transmission power level. 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 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

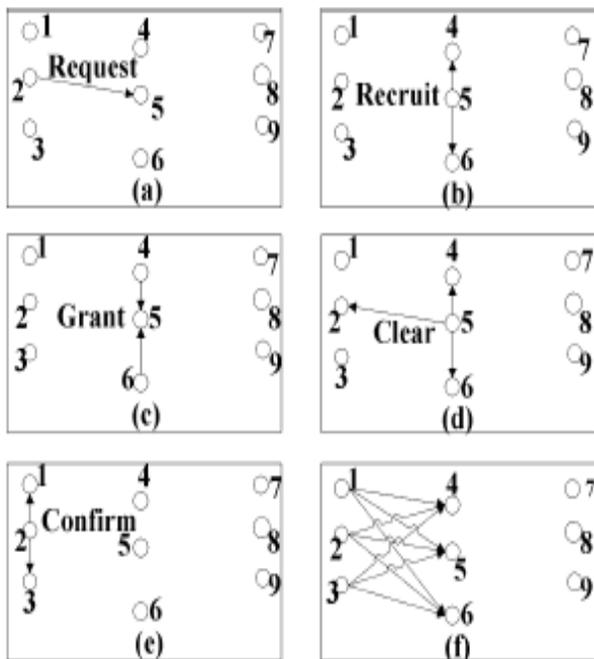


Fig. 4: Example of the recruiting phase operation. (a) Request-to-recruit (RR) packet. (b) Recruit (REC) packet. (c) Grant (GR) packet. (d) Clear (CL) packet. (e) Confirm (CF) packet. (f) Transmission of the data packet.

3.2 Operation of the “recruit-and-transmit” phase

In this phase, the nodes on the initial path become cluster heads, which recruit additional adjacent nodes from their neighborhood. Recruiting is done dynamically and per packet as the packet covers 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.

The example in Fig. 4–(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. 4(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 transmits to its neighbors a recruit (REC) packet [Fig. 4(b)]. The REC packet contains: the id of the previous node (2),

the id of the next node (8), and the maximum time to reply, denoted as T . Each node that acquires 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 determines 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 responds to the REC packet with a grant (GR) packet that contains the computed sum [Fig. 4(c)] after a random back off time drawn uniformly from $(0, T)$. The GR packets advise 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 grants, 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. 4(d)] that contains the ids of the selected cooperating nodes (4 and 6 in our example). 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. 4(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 [Figure. 4(f)].

3.3 Calculation of the Cost of Links

The cost of a link from node to node j , C_i, j , is calculated by node i as: $C_i, j = [(e_i, j) \theta] / [R_i / R_{avg}]$, 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 .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. The protocol-selectable parameter determines 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.

3.4 Error Calculation of Cooperative Model

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. With known signal-to-noise ratio (SNR) at the receiver of SNR, the probability of an error at the receiver is given by $P(\text{error}) = f(\text{SNR}, m) = (1 + (\text{SNR}/2))^{-m}$. In our model, we assumed that the power attenuation due to distance is carried out by $d^{-\gamma_i, j}$, where $d_{i, j}$ is the distance between node to node , and γ is the attenuation exponent. In particular, let P_n 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 to node, the SNR measured at the receiver j is computed as $\text{SNR} = [(P_t/d_{i, j}) / P_n]$. In other words, to achieve a certain value of SNR, the transmitter needs to transmit with the power of $P_t = [\text{SNR} \cdot d_{i, j} \cdot P_n]$. The bit error probability is then terminated by (4). We also assume that for a packet to be successfully received, all the bits in the packet must be successfully received.

4. FAILURE PROBABILITY

We figure out 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 and the one-path scheme.

4.1 Cooperative Transmission Protocol

Let the nodes in the cluster be allocated from 0 to m-1. We denote the transmission pattern of nodes in a sending cluster by a binary representation $b_{m-1} \dots b_1, b_2$

... according to which node 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 express the reception pattern of nodes in a receiving cluster by a binary representation $b_{m-1} \dots b_1, b_2$ according to which node 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^J_j be the probability that nodes with binary representation $I = u_{m-1} \dots u_1, u_2$ transmit a packet of length L bits to nodes with binary representation $J = b_{m-1} \dots b_1, b_2$ across a single hop, and let SNR_j be the SNR of the received signal at node j. Then

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

$$g^J_j = \prod_{i=0}^{m-1} [(1 - b_j)(1 - (1 - \text{BER})^L) + b_j(1 - \text{BER})^L].$$

Let vector $V(i)$ be the binary representation of integer. We define: $g^{v(0)}_{v(0)} = 1, g^J_{v(0)} = 1, J \neq v(0)$. Let A_{jk} be the probability that a packet reaches the kth hop to nodes with binary representation J, given that at least one copy reaches hop k-1, then

$$A_{jk} = \sum_{I=1}^{2^n-1} g^J_{v(I)} A_{v(I)k-1}$$

where

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

Now, let B^h_{CwR} be the probability of failure of a packet to reach any node by the hth hop.

$$B^h_{CwR} = \sum_{k=1}^h A_{v(0)k}$$

4.2 One-Path Scheme

The analysis in this case is similar to the disjoint-paths case, but with one path only and each node transmitting with power of, where

$$\sum_{j=1}^m P_t(j)$$

is the transmission power of the j th node. Let $P_t(j)$ is the probability of failure of a packet to reach the h th node of the one-path scheme, then

$$B_{\text{noC}}^h = 1 - [1 - (f((m \cdot P_t) / (P_n d_r \beta^n)))]^{1/h}$$

4.3 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_h CAN be the probability of failure of a packet of length L bits to reach the node at the h th hop

$$B_{\text{CAN}}^h = \tilde{\Pr}(X_h=0)$$

$$= \sum_{I=1}^m \Pr(X_h=0 \mid X_{h-1}=u_0, \dots, X_{h-n}=u_{n-1})$$

Where $n = \min(m, h)$. The first term in (7), the probability that a packet is not received at the h th hop given that the last n nodes transmit with binary representation $I = u_{n-1} \dots u_1, u_0$.

5. ENERGY CONSUMPTION

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 one-path scheme. To make the comparison of energy consumption of any two schemes meaningful, the failure probability, as defined in Section 4, 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. 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 CAN protocol, 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 .

6. SIMULATION RESULTS

For sample we took the result of number of nodes shown in Fig. 5, 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 design the capacity versus 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 th 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 absorbed in other parallel transmissions. The largest capacity is accomplished at $m=2$, for a transmission range =50 m, at $m=3$ for a transmission range m , and at for a transmission range 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. In Fig. 6, we plot the ratios $Cr1$, $Cr2$, and $Cr3$ for $p\gamma=0.01$, and for $p\gamma=0.1$, 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 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.

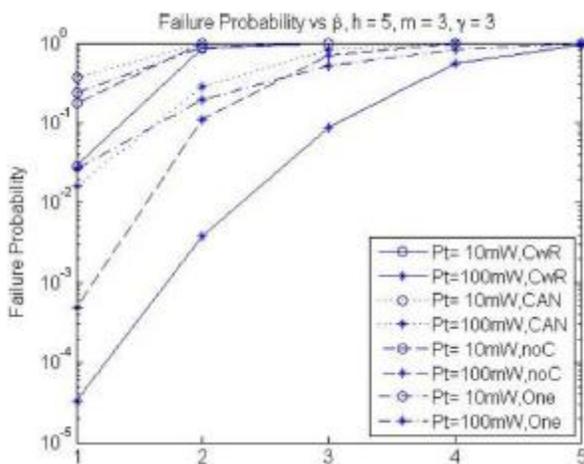
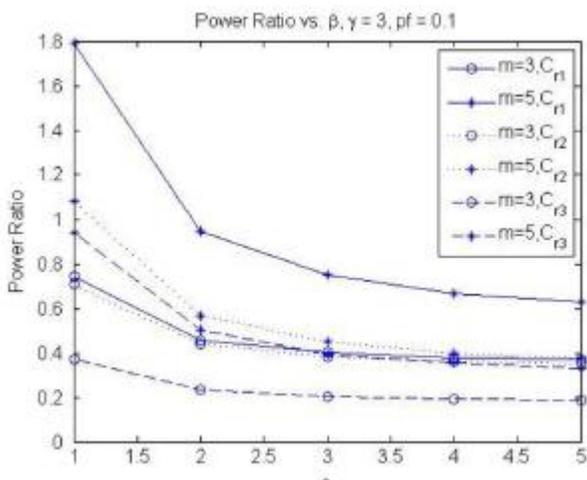
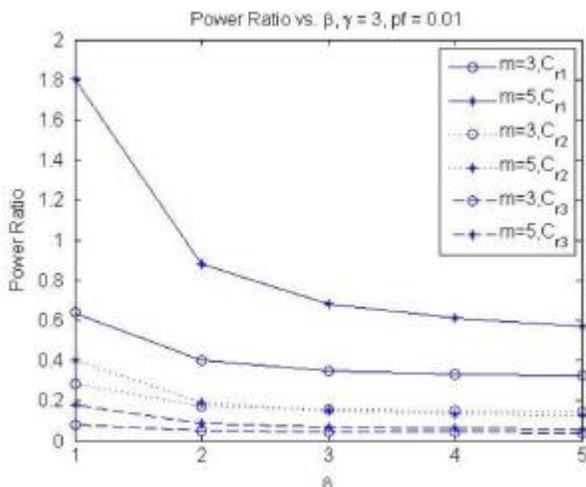


Fig.5: Failure probability ratio

Fig 6.: Power cost ratio β , $\gamma = 3$.

7. CONCLUSION

We estimated the performance of cooperative transmission, where nodes in a sending cluster are synchronized to correspondence 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 is 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. Same way, compared to the CAN protocol and to the one-path scheme, this reduction amounts to a factor of up to 10 000.

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%. 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. Overall, the study determines 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 smaller delay under high-load conditions, as compared to the CAN protocol and the disjoint-paths scheme.

REFERENCES

- [1] Pei Liu, Zhifeng Tao, Sathya Narayanan and Thanasis Korakis, "CoopMAC: A Cooperative MAC for Wireless LANS," *IEEE Journal on Selected Areas in Communication*, vol. 25, no. 2, pp. 340 - 354, Feb 2007.
- [2] H. Zhu and G. Cao, "rDCF: A Relay-Enabled Medium Access Control Protocol for Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1201-1214, Sept. 2006.
- [3] Mohamed Elhawary and Zygmunt J. Haas, Fellow, "Energy-Efficient Protocol for Cooperative Networks" *IEEE/ACM Trans, Networking*, vol. 19, no. 2, Apr. 2011

[4] 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.

[5] Hanguan Shan, Member, IEEE, Ho Ting Cheng, and Weihua Zhuang, "Cross-Layer Cooperative MAC Protocol in Distributed Wireless Networks", *IEEE TRAN. ON WIRELESS COMM (ACCEPTED)*

[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] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 3062–3080, Dec 2004.

[8] 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.

[9] S. Yi, B. Azimi-Sadjadi, S. Kalyanaraman, and V. Subramanian, "Error control code combining techniques in cluster-based cooperative wireless networks," in *IEEE International Conference on Communications (ICC)*, 2005.

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

[11] A. Stefanov and E. Erkip, "Cooperative information transmission in wireless networks," in *Asian-European ITW 2002*, Jun 2002.

[12] T. E. Hunter and A. Nosratinia, "Diversity through coded cooperation," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 283–289, Feb 2006.

[13] 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.

[14] Lu Jin, Zhongwei Zhang, Hong Zhou. "Implementing and Evaluating An Adaptive Secure Routing Protocol for Mobile Ad Hoc Network". *Wireless Telecommunications Symposium*, California, April 27-29, 2006.

[15] Jing Nie, Xin He, Zheng Zhou, Chenglin Zhao, Feng Lu, Danjing Xie. "An Adaptive Fuzzy Logic Based Secure Routing Protocol in IPv6 Ad Hoc Networks". *Processing of Wireless Telecommunications Symposium*, Pomona, California, April 28-30, 2005

[16] Jin, Lu and Zhang, Zhongwei and Zhou. "Performance Comparison of the AODV, SAODV and FLSL Routing Protocols in Mobile Ad Hoc Networks". Jin et al. *Proceedings of the 5th Workshop on the Internet Telecommunications and Signal Processing*. Type: conference_proceedings. Pages: 6-11. (2006).

[17] H. Shen and S. Kalyanaraman, "Asynchronous cooperative MIMO communication," in *Proc. IEEE WiOpt*, Limassol, Cyprus, Apr. 2007, pp. 1–9.

[18] K. Sundaresan, R. Sivakumar, M. Ingram, and T. Chang, "A fairmedium access control protocol for ad-hoc networks with MIMOlinks," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, vol. 4, pp.2559–2570.



Mr.Sagiraju Srinadhraju, Postgraduate student, Kakinada Institute of Engineering and Technology, Kakinada, E-Mail: ssraju.cse.siet@gmail.com

Mr.K.Ramesh has been working as an Assistant Professor in CSE Department, Kakinada Institute of Engineering and Technology, Kakinada, E-mail: kramesh861@yahoo.com