# Reliable and Energy Efficient Protocol for Cooperative Wireless Sensor Networks

R.Parimala Thangam
M.E Student - CSE
Dr. Sivanthi Aditanar College of Engineering
Tiruchendur-628215,TN,India.

G.Wiselin Jiji
Professor & Head - CSE
Dr. Sivanthi Aditanar College of Engineering
Tiruchendur-628215,TN,India.

## ABSTRACT

In cooperative networks, each node in the routing path recruits the neighboring nodes to transmit and receive the data to assist in communication. It forms a cluster at transmitting and receiving end and then form a transmission link between these two clusters. This paper proposes a new reliable and energy efficient cooperative protocol to establish a cluster at the receiver end, before receiving the data. The end-to-end robustness of the protocol to data-packet loss, along with the tradeoff between energy consumption and error rate is analyzing here. The analysis results are used to compare the end-to-end robustness and energy saving of our new protocol with other two schemes such as one non-cooperative and one another cooperative scheme named as CAN-I. The reduction in error rate and the energy savings translate into increased lifetime of cooperative sensor networks.

**General Terms** – Clustering, cooperative networks energy-efficient protocols, cooperative transmission, Routing, sensor networks.

## Keywords

CN, FLSL

## 1. INTRODUCTION

In WIRELESS SENSOR NETWORKS, the nodes often spend most of their energy on communication. In many applications, the nodes are small and have limited and non-replenishable energy supplies. The recent technology which allows reliable transmission and energy efficiency is Cooperative communication. In which multiple nodes can transmit and receive the data packets simultaneously. In existing techniques, multiple transmissions has takes place at each hop, but this new protocol allows simultaneous transmissions. The nodes in the path from source to destination will be considered as a cluster head. 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. For example, In Figure.1 the width of each intermediate hop is 3. 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 Figure. 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.

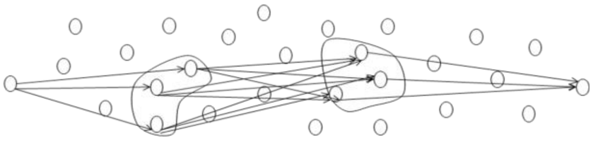


**Figure 1. Proposed cooperative transmission protocol**

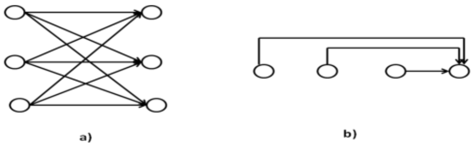


**Figure 2. (a) proposed cooperative and (b) the CAN reception models.**

Our cooperative transmission protocol consists of two phases. In the *routing phase*, the initial path between the source and the sink nodes is discovered by FLSL algorithm. In the next 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 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.

We compare our cooperative transmission protocol with another cooperative protocol, called Cooperation Along Non-cooperative path (CAN) [2], and with another non-cooperative scheme: "*one-path*" . The equivalent of the "one-node-thick" path is called in [2] 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. 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 Figure. 2(b). In the one-path scheme, the "one-node-thick" path is discovered first. The overall operations depicted in figure.3.

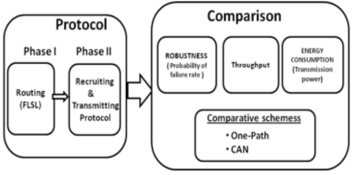

Protocol
Phase I
Phase II
Routing (FLSL)
Recruiting & Transmitting Protocol
Comparison
ROBUSTNESS (Probability of failure rate)
Throughput
ENERGY CONSUMPTION (Transmission power)
Comparative schemess
• One-Path
• CAN

**Figure 3. block diagram**

## 2. RELATED WORK

Cooperative communication is a technology of creating spatial diversity and robustness against the channel variations due to fading in wireless network. It allows nodes to cooperate in their transmissions in order to improve the overall performance of the network. CoopMAC [1] and rDCF [2] are the representations of the cooperative MAC. In CoopMAC protocol high data rate nodes assist low data rate nodes in their transmission by forwarding their traffic. The rDCF is an approach similar to CoopMAC, which enables packet relaying by requiring each station to broadcast the rate information between nodes explicitly.

The problem of energy-efficient routing in wireless networks that support cooperative transmission was formulated in [4].Here 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. An optimal grouping strategy for efficient helper node selection, and advised a greedy algorithm for MAC protocol refinement is given in [5]. This approach can effectively exploit beneficial cooperation, thereby improving system performance. Coded cooperation for transmission between two sending nodes and one receiving node is proposed in [12]. In each time slot, only one of the sending nodes transmits a data block that contains N1 bits from its own coded bits and N2 bits from its partner. The receiver then combines the received bits from the two senders by code combining.

In cooperative networks, the transmitting nodes uses idle nearby nodes as cooperative one to provide spatial diversity. But most of previous research considers the transmission between two senders and one receiver [6]–[8], [11], [12] or multiple nodes between source and destination [10].In [9] multiple receiving nodes form the receiving cluster and the sending node transmits packets to the receiving cluster. Each cluster member relays its signal copy to the destination. The destination node uses code combining techniques to decode the original information bits. It uses code combining in the receiving group of cooperative MIMO system. 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.

Before transmitting the data a special secure routing protocol, which is security conscious, is needed for wireless networks. In [15] the implementation of a new security conscious routing protocol, FSLS, is described. This protocol is used in the routing phase. From [16] FLSL protocol could reliably select the data transmission route with high security level, and self-adaptively and dynamically adjust the route updating without delay. Comparing with AODV and SAODV routing protocols, FLSL spends reasonable and affordable time on security-level algorithm and route selection to improve the reliably and security of WSNs.

The model in [13] utilizes multiple nodes to forward the data, but only one node can transmit at any time. 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. Lots of researches on cooperative communication have been done, but few of them consider the energy consumption for cooperative communication.

## 3. PROTOCOL DESIGN

It consist of two phases: 1.Routing Phase, 2.Recruit & Transmit Phase. The routing phase of the protocol, which is responsible for discovering an initial route from the source node to the sink node, could be implemented 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 exchange 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 discovers the initial path from source to sink. For the purpose of performance evaluation, we chose to implement this phase using the Adaptive Fuzzy Logic Based Security Level Routing Protocol (FLSL).

### *3.1.1 FLSL Protocol*

The FLSL protocol is developed based on SAODV protocol, and security level algorithm has been used to assess the reliability nodes and determine the most secure route among a few possible routes. In FLSL, A new attribute, Security Level, is introduced in the format of protocol control messages and routing table to denote the reliability and dependability of certain mobile host or route. Meantime, because FLSL protocol enables the destination node to accept multi-Route Request message, the security level is also used by source node and destination node to determine the most secure route. In wireless environment, the security level of a mobile node is affected by many conditions.

1. Secret key length ($l$). Longer the secret key is, stronger to defend serious brute force attack.
2. Changing frequency of secret key ($f$ ). If mobile host's secret key is changeable, the difficulty of decryption must be increased and security level of mobile hosts also get enhanced.
3. Amount of active neighbor hosts ($n$). More active neighbor hosts existing will increase the percentage of potential attackers existing.

Apparently, the security level of a single node has a relation with these three factors as follows:

$$S \propto l \times f \times (1/\mathrm{n}) \qquad (1)$$

The Security-Level of a route is decided by the node which has the lowest Security-Level in that route. In another word, the route with the highest Security-Level is comparably most secure. More precisely, if we define the source node as $S$ and the destination node as $D$ and assume that there are totally $n$

possible routes, i.e. *R*1,*R*2, . . . ,*Rn*, from the source *S* to the destination *D* . In the route *Ri*, there are intermediate nodes $n_i^1, n_i^2, \ldots, n_i^{j,} \ldots, n_i^m$, totally *m* possible nodes to forward the packets from the source to the destination. If the current Security-Level of the *jth* node in the *ith* route is *Si j* , the Security Level of the *ith* route is defined as:

$$SL_i = \min_{j\,\text{€}\,(1,\ldots,m)}(Si\,j), \qquad (2)$$

The most desired route *Rk* is the maximum value of all those route [14], i.e.:

$$SL_k = \max_{i\text{€}\{1,2,\ldots,n\}}(SLi) = \max_{i\text{€}\{1,2,\ldots,n\}}\big(\min_{j\text{€}\{1,2,\ldots,n\}}(Si\,j)\big) \qquad (3)$$

Therefore, the FLSL protocol is capable of determining a more secure route among possible routes by comparing the security level while the security level of each individual node is evaluated. The procedure of route discovery is described in Algorithm 1.

```
Source node S calculates SLS and generates RREQ
Source node S broadcasts RREQ to all of its neighbors
while Neighbor node i is not destination node D do
    Authenticate and verify the RREQ
    Calculate node i's security level SLi
    if SLi < SLq then
        Update the security level in the RREQ packet by overwriting
        the SLq in RREQ with SLi
    end if
    Broadcast the RREQ to node i's neighbor nodes
end while
for all RREQ messages received by destination node D do
    if There is available route to source node S then
        if SLq > SLRT then
            Update routing table using the latest data in RREQ
        else
            Drop the RREQ
        end if
    else
        Create entry in routing table using the latest data in RREQ
    end if
    Increase sequence number by 1
    Create a RREP
    Unicast RREP back to source node S
end for
for all RREP messages received by source node S do
    Update routing table using the latest data in RREP
end for
```

**Algorithm 1. FLSL Route Discovery**

## 3.2 Recruit and transmit phase

The example in Figure. 4(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 [Figure. 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 [Figure. 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 [Figure. 3(c)] after a random back off 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.

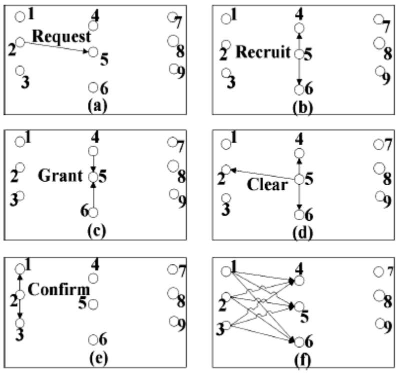


**Figure 4. Example of the recruiting phase operation. (a) Request-to-recruit (RR)packet. (b) Recruit (REC) packet. (c) Grant (GR) packet. (d) Clear (CL) packet.(e) Confirm (CF) packet. (f) Transmission of the data packet.**

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 [Figure. 3(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 [Figure. 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 Cost of the links

The cost of a link from node to node j,$C_{i,j}$, is calculated by node 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 , 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 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.

## 3.4 Network 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

$$P(error)=f(SNR,m)=(1+(SNR/2))^{-m} \qquad (4)$$

In our model, we assumed that the power attenuation due to distance is governed 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 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 $P_t$ = [SNR. $d^{\gamma}_{i,j}$ * $P_n$ ] .The bit error probability is then determined 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 RATE

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 and the one-path scheme.

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

$$BER=f(\ SNR_{j,} \sum_{i=0}^{m-1} u_i)$$

$$g^I{}_J=\prod_{j=0}^{m-1}[(1-b_j)(1-(1-BER)^L) +b_j (1-BER)^L$$

Let vector V(i) be the binary representation of integer . We define: $g^{v(0)}{}_{v(0)}$= 1, $g^J{}_{v(0)}$= 1, J≠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

$$A_{JK} =\sum_{I=1}^{2^m-1} g^J{}_{v(I)}A_{v(I)k-1}$$

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

$$B^h{}_{CwR} =\sum_{k-1}^{h} A_{v(0)k} \qquad (5)$$

## 4.2 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 , where $\sum_{j-1}^{m} P_t(j)$ is the transmission power of the jth node. Let $P_t$(j) be the probability of failure of a packet to reach the hth node of the one-path scheme, then

$$B^h{}_{noC} =1-[1-(f([( m.P_t)/( P_n d_{\gamma}ß^{\gamma})],1))^{Lh} ] \qquad (6)$$

## 4.3 CAN

Let $X_i$=0 represent the event that a packet is not received at the th hop along the non-cooperative path, while $X_i$=0 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

$$B^h{}_{CAN} =Pr(X_h=0)$$

$$= \sum_{I=1}^{m} Pr(X_h=0\ X_{h-1}=u_0,\ldots, X_{h-n}=u_{n-1}) \quad * \quad Pr(X_{h-1}= u_0\ \ldots, X_{h-n}=u_{n-1}) \qquad (7)$$

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 $I=u_{n-1} \ldots u_1,u_0$ .

# 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 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. 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 $m^2.P_t$ , 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 $m.P_t$, and we assume that the power consumption for the one-path protocol is $P_t$.

# 6. THROUGHPUT

We analyze and compare the capacity of a single flow for these 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 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 the transmission time of the ACK packet is very small compared to the data packet, so we ignore it.

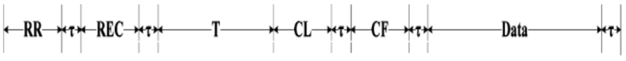


**Figure 5. cycle of the “recruiting-and-transmission” phase.**

One cycle of the control packets and the data packet transmissions in our cooperative transmission protocol is shown in Figure 5. In this Figure, Γ 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

$$Cycle_{CwR=} (4\ L_{c/W})+(L_{d/W})+5\Gamma+T$$

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 on e cycle duration is then

$$Cycle_{CwR=}(4\ L_{c/W})+(L_{d/W})+(N_b\ L_c)/_W+(N_b+5)\Gamma$$

The capacity upper bound of one hop in our cooperative transmission protocol is then defined as:

$$Tr_{CwR=}\ L_{d/}\ Cycle_{CwR}.$$

The upper bound of the capacity of one flow between the source and the sink in our cooperative transmission protocol is then

$$PT_{CwR} = (Tr_{CwR/3})(1- B^h_{CwR}) \qquad (8)$$

Next, we compute the upper bound of the one-path 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, $Cycle_{noC}$ , is

$$Cycle_{One=}[(2\ L_c+ L_d)/W]+3\Gamma \qquad (9)$$

The capacity upper bound for one hop and one flow in the one-path scheme is calculated by $Tr_{noC}$ and $PT_{noC}$ . Similarly, the upper bound of the capacity of one flow between the source and the sink in the one-path scheme is

$$PT_{One} = (Tr_{CwR/3})(1- B^h_{One}) \qquad (10)$$

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

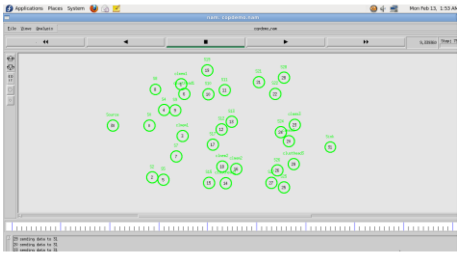
$$Cycle_{CAN=}(3\ L_c+ L_d)/_W+4\Gamma$$

The upper bound of the capacity of one flow between the source and the sink of the CAN protocol is

$$PT_{CAN} = (Tr_{CAN}/_{(m+2)})(1- B^h_{CAN}) \qquad (11)$$

## 7. SIMULATION RESULTS

In the experimental set up, we have 32 nodes and placed in random manner ; among these 31st and 32nd nodes are consider as the source and the sink nodes, and the nodes are placed in a 400× 400 m area, and the transmission range per node is 100 m. The number of cooperative nodes is 3.



29 sending data to 31
30 sending data to 31
31 sending data to 31

**Figure 6. cluster formation of our cooperative transmission protocol**

Thus the cluster heads in the routing path will recruit the neighbors based on the energy consumption. So that two nodes with the smallest reported cost are selected to form a receiving cluster. Then the packets will transfer from all nodes in the sending cluster into all nodes in the receiving cluster.Figure.6 shows the formation of clusters. It demonstrates the cluster head and cooperative nodes in the cluster with their ids.

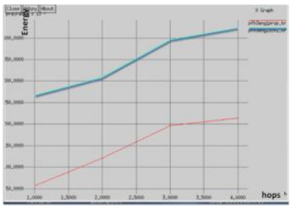


**Figure 7. Failure probability ratios**

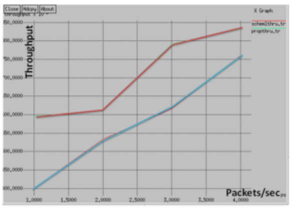


**Figure 8. Throughput**

We design the failure probability ratios using four different intermediate hop values h=1, 2, 3, 4. 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. Fig. 7 shows, our cooperative transmission protocol has lower failure probability compared to one path scheme.

We plot the throughput ratios in Fig. 8. Our cooperative transmission protocol has higher capacity compared to the existing scheme. Our cooperative transmission protocol has larger capacity depending on the value of m.

The total energy consumption measures the sum of the energy of all packet transmissions (control and data packets). The energy consumption ratios shows the total energy consumption of our cooperative transmission protocol to the total energy consumption of the CAN protocol. Figure.9 shows that our cooperative transmission protocol has lower energy consumption compared to the CAN protocol.

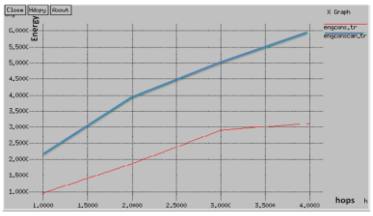


**Figure 9.Energy consumption ratio**

## 8. CONCLUSION

This paper introduces a new protocol to facilitate cooperative transmission. In the *routing phase*, the initial secure path could be discovered between the source and the sink nodes by means of FLSL Algorithm and In "*recruiting-and-transmitting*" phase, the cluster on the initial path recruit additional nodes from their neighborhood, hence form a cluster. The simulation results show that our proposed cooperative transmission protocol reduces the Energy consumption and Failure rate. Also it improves the throughput with the comparative scheme. The reduction in failure rate and energy consumption translates into increased lifetime of cooperative sensor networks.

## 9. REFERENCES

[1] Pei Liu, Zhifeng Tao, Sathya Narayannan 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] Hangguan 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 diversityparti: 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).