Robust Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks

Prasun Seal*, Mainak Chatterjee†, and Tuhina Samanta*
*Indian Institute of Engineering Science and Technology, Shibpur, India
Email:prasun1926@gmail.com, t_samanta@it.becs.ac.in
†University of Central Florida,Orlando, FL
Email: mainak@eecs.ucf.edu

***Abstract*—Energy efficient routing protocol design in the presence of noise and interference is a challenging task in wireless sensor networks (WSNs). In this paper, we present a robust, energy efficient multi-path routing protocol that uses the residual energy and Signal-to-Noise Ratio (SNR) to predict the best next hop node during the route establishment. The new protocol uses multi-path to transmit the data. It splits the message into number of segments of equal size and transmits via multi-path without excessive delay. We studied the performance metrics and compared with other protocols. Simulation results shown that our protocol saves more energy, has lower end to end delay and jitter, and higher packet delivery ratio than other protocols.**

***Keywords*-Wireless Sensor Networks; Multipath Routing; Energy Efficiency; End-to-end delay.**

I. Introduction

Recent developments of low-power wireless sensor embedded computing and middle-ware have made it possible to develop low-cost, low-power, and multi-functional microsensors [5]. Due to the ubiquitous and flexible nature of wireless sensor networks (WSNs), they have been utilized in many applications such as health care, traffic control, and military as a connectivity infrastructure and a distributed data generation and dissemination network. However, the challenge with the vast use of the sensor network is limited power backup and non-replaceable battery of the functioning sensors in WSNs [7]. Apart from energy, reliable data transfer is a crucial factor, where originality of information is affected by fading and interference.

Wireless Sensor Networks (WSNs) are the special types of ad-hoc networks but has restrictions because of limited energy, computing capacity, limited wireless links bandwidth, and less immunity to fault during each hop.

Multi-path routing has the ability to overcome the drawbacks of a single path routing by providing reliability in data transmission, congestion control, quality of service (QoS) and fault-tolerance. Such use of multi-path routes when traffic is diffused over the network overcomes shortcomings like uneven energy dilation and congestion bottlenecks. It can even exploit temporal and spatial fluctuations of wireless links and use favorable conditions in an opportunistic manner [2]. Motivated by the notion of multi-path routing, we try to answer the question: *is it possible to design a multi-path routing protocol guided by SNR and link cost to deliver an energy and delay efficient routing ?*

To answer this question, we present a robust, energy efficient multi-path routing protocol for WSNs, which can achieve reliable data routing and recovery from node failure by efficient use of multiple paths. We mainly focus on establishing multipath using node disjoint path selections without flooding the network in an energy and delay efficient manner. Simultaneously, we study how to split the data into a number of equal sized packets and transmit via multi-path. Lower complexity algorithms which are suitable for WSN to retrieve packets in sink. The proposed algorithms has the following features :

- The new protocol broadcasts a control message to select the neighbor and to predict the next hop for reliable packet transmission.
- During path construction phase based on neighbor knowledge, the nodes use the information of residual energy, and signal-to-noise ratio (SNR) of the other sensor nodes.
- Once the best possible paths are chosen, a source node starts sending data packets.
- The routing protocol splits the messages into equal segments and simultaneously transmits via multiple paths. This increases resilience to path failures and increases the probability of receiving packets without excessive delay.

The rest of the paper is organized as follows. Section II discusses some related works. The protocol details are described in Section III. Section IV presents the experimental setup and comparative studies of our results with other existing algorithms. Finally, Section V concludes the paper with possible future directives.

II. Related Works

Multi-path routing is studied in past elaborately and is still a challenging area of research in application specific routing in WSNs. Energy efficient multi-path routing procedures with multiple sink nodes for WSN have been proposed in [2] and [8]. The authors proposed a multiple node-disjoint paths between the sink and source nodes by using distributed, scalable and localized multi-path search algorithm in the first work. The later work focused on residual energy and buffer size during routing. However, none tried to reduce jitter, and packet delivery delay with interference avoidance between the neighbor nodes. We tried to address these issues in our present work.

In [1] authors proposed a systematic performance study of three routing protocols, namely Ad Hoc On demand Distance Vector $(AODV)$, Dynamic Source Routing Protocol (DSR), and Destination Sequenced Distance Vector

978-1-4673-9692-9/15 $31.00 © 2015 IEEE
DOI 10.1109/iNIS.2015.11

$(DSDV)$. Bellman-Ford routing algorithm is used to develop a modified energy efficient AODV routing protocol. Authors in [7] proposed a trusted and energy efficient routing protocol $(TERP)$, and applied the trust concept to DSDV protocol to increase the network security level suppressing malicious nodes. Encrypted data is transmitted through the network based on the level of security adopted by each of the nodes, hence enhancing the security of information.

In [3] and [4] authors proposed a cross layer quality of service-aware scheduling for wireless sensor network $(CQoSS)$ in an energy efficient way reducing delay and increased reliability. Authors in [3] divide the traffic with various priorities and use a priority queuing protocol to prioritize each node's packet transmission. [4] proposes a multi objective QoS routing $(MQoSR)$, which uses a heuristic approach to select neighbor and uses geographical routing mechanism. Addressing multiple cost functions simultaneously to develop a routing strategy is rare in the literature. The cost function used in our proposed method tries to optimize residual energy, remaining buffer size, link quality (SNR), end-to-end delay, packet delivery ratio and jitter in an energy efficient way.

III. Protocol Description

In this section, we present a detail description of our protocol and a formal representation of the proposed algorithm. The protocol stands upon three consecutive phases: neighbour selection, path establishment and data transmission via multi-path. After random deployment, sensor nodes are active and participate in the protocol. Node starts exchanging their information by broadcasting $HELLO$ messages to select neighbour nodes. After receiving the request messages $RREQ$ from the desired node, source uses a cost function to select the best K paths based on different traffic requirement from the sink to source N number of available paths. To reduce delay and increase resilience to path failures, the transmission message is split up by the protocol into equal size segments. Then transmission takes place via multiple paths simultaneously.

A. Assumptions

Identical sensor nodes are deployed randomly and the nodes have same transmission range. The network is densely deployed and each node has unique ID. Transmission power (P_t) of each node lies within the range $P_{min} \leq P_t \leq P_{max}$ [6].

B. Best Link Cost Function

During path discovery phase, each node calculates the link cost to select its neighbor as a next hop. Let, x and y be the nodes $\{x, y\} \subseteq N$ (total number of node). We use a cost function to choose next hop mentioned in equation (1), which is a modified version from [8].

$$Next\_hop = max_{y \subseteq N}[E_{re,y} \cup L_y \cup I_{xy}] \tag{1}$$

where, $E_{re,y}$ denotes the current residual energy of node y, because during data transmission node y consumes its energy and when it reacts as a node x then it also spends some energy for data transmission. L_y indicates the available buffer size of node y and I_{xy} denotes the interference between two node (calculated using the SNR value of the link) x and y. However, if we consider a path (P), which consists of $M(M \subset P)$ number of nodes (source to destination), the total cost for this path can be calculated as,

$$C_{total,P} = \sum_{i=1}^{M-1} l_{(xy)i} \tag{2}$$

where $l_{(xy)i}$ denotes individual link cost between two nodes $(x, y) \subseteq M$ and the $C_{total,P}$ indicates the total cost for the path P. For the algorithms we use the following definitions.

TABLE I
Algorithms Definitions

Abbreviation	Descriptions
n	Total number of sensors
s_i	$\{s_i.id,\ s_i.x,\ s_i.y\}$
$d(a, b)$	Euclidean distance between two points a and b
TH	Threshold
T_range	Transmission Range
$dist, Val$	Distance, value (Variables)
$n\_count, s_i.source$	Neighbour count, Source node (Variables)
$s_i.nid$	Source node's neighbour ID (Variables)
n_id	Neighbour node ID (Variables)
$flag, temp, Sum$	Value taking Variables
H_Msg	Hello message
R_EN	Residual Energy of a sensor $s_i.id$
BUF_C	Buffer capacity of a sensor $s_i.id$
SNR	Signal to noise ratio of a sensor $s_i.id$
N_i	Neighbour nodes of a sensor
N_Table	Neighbour table
$N\_Table.s\_p$	Field of N_Table, for every sensor $s_i.id \in S$, used to check whether $s_i.id$ already used in other path or not

Algorithm 1: Algorithm Neighbor Selection

input : $S = [s_1, s_2, ..., s_n]$
output: neighbour table of sensors
begin
  for each $s_i \in S$ do **begin**
    for each $s_j \in S$ do **begin**
      dist= $d(s_i, s_j)$
      if $dist < T\_range$ **then**
        begin
          val= SEND_HELLO(H_MSG)
          if $val > TH$ **then**
            begin
              $s_i \leftarrow s_j$ is a neighbor
              $s_j \leftarrow s_i$ is a neighbor

C. Neighbor Selection

Neighbor selection is the primary phase of path discovery. At very beginning each sensor node present in the network

Algorithm 2: HELLO Message

input : H_Msg
output: return link value of a sensor node
begin
  Sum $= R\_EN\_s_i.id + BUF\_C\_s_i.id + SNR\_s_i.id$
  return (sum)

Algorithm 3: Path Selection

input : $s_i.id, s_i.source \in S$
output: Sink to Source path established
begin
  if $N\_Table.s\_p \neq \emptyset$ **then**
    n_count $\leftarrow s_i.idN_i.count$
    for each $s_i.id \in S$ do **begin**
      if $s_i.nid \in N_i == s_i.source \in S$ **then**
        Send $\rightarrow RREQ$
        Break
      if $s_i.nid \in N_i == n\_count$ **then**
        $N\_Table.s_p = \emptyset$
  else
    while $s_i.id \neq s_i.source$ **do**
      n_id = Best_link $(s_i.id \in S)$
      Send $\rightarrow RREQ$
      $s_i.id = n\_id$

Algorithm 4: Best link

input : $s_i.id \in S$
output: Provide best link quality sensor $s_i.id \in S$
begin
  n_count $\leftarrow s_i.idN_i.count$
  for each $s_i.id \in S$ do **begin**
    flag $\rightarrow s_i.id.flag$
    if *flag* $== 0$ **then**
      val $= s_i.idlinkquality$
      if *temp*$<$ *val* **then**
        temp=val
        n_id $\leftarrow s_i.id$
  return (n_id)

selects their neighbors by checking link cost level. That means x first broadcasts a $HELLO$ message to its neighbors (within the transmission range of x). Message construction is given in Algorithm 2. After receiving a message the neighbor nodes first check the link cost level of the node, which broadcasts the message. If the link cost level is greater than certain threshold it selects the node as a neighbor and updates its neighbor table. This procedure is done by each sensor node until every node constructs its neighbor table. In this phase, each sensor node maintains and updates its neighbor table. A formal description of neighbor selection is shown in Algorithm 1. Table 1 shows the variables used in algorithm.

D. Path Discovery

Proposed path discovery is a node disjoint multi-path creation method. If a node is selected as a member of a path then it can not be a member of another path. Maximum utilization of network resources and most fault-tolerant node disjoint paths are usually performed in our proposed work. That means if a node fails to route due to less energy or less buffer capacity, the path containing this node is affected, deciding on the other nodes with minimum impact to the diversity of the routes.

In this mechanism when a source node is willing to establish a path to destination, it first sends a $RREQ$ message to its entire neighbor, after that the neighbor node sends the $RREQ$ message to its neighbor. Finally, the $RREQ$ will reach the destination by traversing multi-node hop. The cost of routing will be high in such mechanism, like when a node sends a $RREQ$ message to its neighbor the cost will be at least,

$$\sum_{i=0}^{n} T_{econ} \quad \propto \quad d_{xy} \tag{3}$$

where T_{econ} denotes total energy consumption per transmission with respect to distance d_{xy}. When a node receives $RREQ$ message from its neighbor, the cost (with respect to energy and buffer) will be as shown below,

$$\sum_{i=0}^{n} R_{econ} + B_i \tag{4}$$

where, R_{econ} denotes energy consumption per buffer (B_i). However, our proposed method of best path selection is a modified strategy as is described next.

1) Best Path Selection: Best path selection is performed in such a way to minimize the cost functions mentioned in Equations (3) and (4). After neighbor selection, each node has the information about their neighbor. To select the best path, we use 15 byte long $RREQ$ packet. At first, the sink checks its neighbor table if the destination id is present or not. If the id is present then it checks the link quality and sends the $RREQ$ and the path is selected. If not then the sink node set a $NULL$ value in a field of its neighbor table so that it does not need to check further neighbors id in secondary path selection and send an $RREQ$ message most performed next hop. The next hop of the sink is selected based on best link cost function by checking its neighbor table. Same as sink, the next hop also check its neighbor table and set the value $NULL$ or 1 to the field in the neighbor table. If the value is 1, send the $RREQ$ and select a path. Otherwise, it selects its best performed next hop based on best link cost function and sends the $RREQ$ message in the direction of the source node. This process will go on until the $RREQ$ message reaches the source node. Algorithms 3 and 4 depicts the procedure formally.

Let x be a node in the network, $x \subseteq N$, where N is the total number of nodes present in the network, x first sends a

$RREQ$ message to node y by initializing the values in $RREQ$ packet.

(i) $h_{count} = 0$; h_{count} denotes the hop count.
(ii) $L_{path} = 0$; L_{path} denotes the number of loaded nodes in the path.
(iii) $D_{path} = 0$; D_{path} is the end-to-end path delay.
(iv) $I_{path} = 0$; I_{path} is the end-to-end path quality.
(v) E_{path} = energy value of node x.

After receiving the $RREQ$ message by the node y, it stubbed its information before sending to a neighbor. Then,

(i) $h_{count} = h_{count} + 1$;
(ii) $L_{path} = L_{path} + 1$; when $L_{path} > L_{level}$, no change in L_{path};
(iii) $D_{path} = D_{path} + d_{si}$; where d_{si} denotes delay from node s_i;
(iv) $I_{path} = I_{path} + Int_{si}$; where Int_{si} denotes SNR value chosen between 10 to 30 as per the threshold calculated from [7], [2] ;
(v) $E_{path} = E_{path} + e_y$; when e_y is energy value of node y;

Next, $RREQ$ will send maximum value of best link neighbor computing D_{path} and I_{path} to choose next hop.

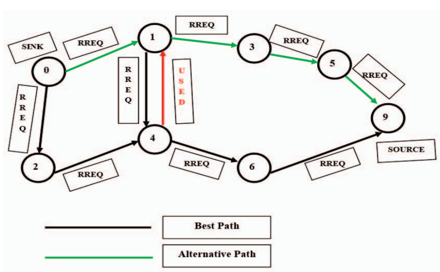

Fig. 1. Example of Path Discovery

Figure 1 gives an example of our proposed method. Let 0 is a sink node, establishing a path to 9. Now, the sink already established the best path to source node 9 using the best link cost function. To establish an alternative path, sink node sends a $RREQ$ message to node 1 which is the next performed next hop. Now, node 1 first check which of its neighbor have highest link cost value. So, it gets the node id 4 and sends a $RREQ$ message to node 4. But the node 4 is already selected as member of a path. So, it sends a $USED$ message to node 1. After getting the $USED$ message node 1 immediate search its neighbor table and computes the next preferred neighbor (node 3) and send $RREQ$ message to node 3. Node 3 accepts the $RREQ$ and again sends $RREQ$ message to its most preferred hop (node 5). This procedure will continue until a path is establishes to source. However, in our protocol there is no need to send response delivery ($RREP$) messages because we select paths from sink to source. So the source node gets the all information about the paths during $RREQ$ message transmission; as a result no more energy consumption will happen during path selection.

E. Data Transmission using Multipath

Before starting the discussion about data transmission using multi-path, we perform two procedures to get the proper output from the new protocol. When the source node receives $RREQ$ messages it estimates all paths to sink using some path parameters. To calculate each path cost function some parameters are taken into account like energy, load and delay. So, the P_{cost} can be calculated as,

$$P_{cost} = \frac{e_{level}}{E_{path}} + \frac{l_{level}}{L_{path}} + \frac{delay_{level}}{D_{path}} \tag{5}$$

where, e_{level}, l_{level}, and $delay_{level}$ denote the node energy, load and delay threshold value, which is going to participate in data transaction. If a node energy value is less than threshold energy level, then it is unable to take part in data transmission. Then the source calculates each path value, which can give optimal cost. The source then evaluates the optimal paths for data transmission. After calculating P_{cost} of all available paths, they are sorted based on cost like $P_{cost1} > P_{cost2} > P_{cost3} \ldots$ so on. During path selection, path with lowest cost is selected as the best path, then the second lowest and so on. We select K number of paths from N available paths to transmit data via multi-path. That means the lowest cost path, which is selected as a best path is denoted as K and secondary path is K_1 and so on from the N available paths. Now the data transmission procedure will be done via K available selected multi-paths.

Before starting discussion about sending data via K multi-path, classes of traffic pattern need to be discussed.

1) Traffic classes and Number of Routing Paths: Mainly two types of traffics are distinguished in our work, real-time and non-realtime traffic. This real time traffic needs reliability, speedy delivery or some times both. To distinguish the traffic classes separately in packet header, two descriptive bits are added. Where two consecutive zero (00) denotes non real time traffic. 01 denotes packet with delay tolerant reliability traffic, 10 denotes data packet with reliability having low delay and 11 denotes reliability with speedy delivery.

We assume that sensor nodes have the ability to pick the lowest cost path value (K_1) and the highest cost path value K_n from out of the K paths. $K_1 + L = R_{path}$ number of paths are picked up by the protocol to transmit all kinds of real time traffic and

$$K - (K_1 + L) = NR_{path} \tag{6}$$

path for non real time traffic, where

$$K = R_{path} + NR_{path} \tag{7}$$

To make proper understanding about path distribution a short example is discussed: Let, an array is sorted in ascending order. Now, if we need to calculate a lower middle order values then the value can be calculated as:

$$MID = ((K_1 + K_n)/2) \tag{8}$$

$$L_{mid} = ((MID + K_n)/2) \tag{9}$$

The values from L_{mid} to K_n are the paths denoted as NR_{path} to transmit non real time traffic, where

$$L_{mid} < NR_{path} \leq K_n \tag{10}$$

The K_1 to L_{mid} value paths are used for real time traffic. But what will be happened for those traffic which need delay

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Max. number of nodes (N)	500
MAC type	IEEE 802.11/ DCF
Traffic type	Constant bit rate
Agent	UDP
Queue length	50 packets
Connection Rate	4 pkts/sec
Tx power	0.2818 W
Transmission range	250 m
Initial energy	200J

tolerant reliability ? To answer this question we allocate those paths whose values are from MID to L_{mid}. Such type of path denoted as R_{delay}, where

$$MID < R_{delay} \leq L_{mid} \tag{11}$$

The rest of the value paths are used for those real time traffic which need reliability with low delay or speedy delivery, denoted as

$$K_1 \leq R_{path} \leq MID \tag{12}$$

Continuous use of the same path during data transmission is not a good option. So, we divide the message into N segments $(S_0, S_1, S_2, \ldots, S_{N-1})$, and with the original message overhead part of $M + 1(N > M)$ error correction codes $(C_0, C_1, C_2, C_3, \ldots C_M)$ are added. Simultaneously this error correction code and data segments are of equal size, should be multiple of 8 [2].

IV. EXPERIMENTAL RESULTS

We run simulations for various WSNs routing topologies, which are studied in this paper. "GNU" licensed C Compiler (GCC) is used to implement our protocol, and simulation results are compared with the protocols Dynamic Source Routing (DSR) and Ad Hoc On demand Distance Vector Routing $(AODV)$. N $(100, 200, 300, 400, 500)$ numbers of randomly deployed sensor nodes in $2125m \times 2125m$ area with 10 CBR sources are used for our experimental simulation. All nodes have a radio transmission range set to $250m$. The sink node is situated at the upper right corner of the simulation field, and the source nodes are situated on the left bottom corner. Table II shows the simulation parameters.

We investigated the proposed protocol $(PROTO)$ performance in a multi-hop network topology that indicates it uses the best available multi-paths, which are selected based on the link quality to send data from the source to the destination. Performance metrics used in the evaluation are the average energy consumption, average delivery ratio, average delay variance $(jitter)$ and average end to end delay. We fixed the packet arrival rate to 4 packets/second and used fixed data packet length of 512 bytes long.

A. Average End to End Delay (EED)

$$EED = \sum_{i=0}^{n} \frac{P_{receivedi} - P_{transmiti}}{n} \tag{13}$$

Here, n denotes total number of delivered data packet and $P_{received}$ represents the time when a sink receiving a data packet, $P_{transmit}$ is the transmission time when data packets is going to be transmitted by each source node. Hence, average time to send a data packet from source to sink is represented by Equation (13).

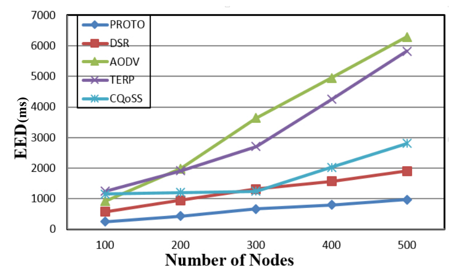

Fig. 2. Average End to End Delay

Figure 2 shows the comparison on average end to end delay. The average end to end delay for the new protocol gives the better output than other tested protocols, when the network size is large. The new protocol output shows that, it is more stable when compared with $TERP$, DSR, $AODV$ and $CQoSS$ [7], [1], [3]. $AODV$ and $TERP$ perform better when the number of nodes is low but DSR, $CQoSS$ and the proposed protocol $(PROTO)$ performs better than $AODV$ when the number of nodes increase. The new protocol output is almost less than $0.65s$ (300 nodes) and $1s$ (500 nodes) than DSR, which is about 49%better than DSR. Simultaneously, the new protocol performs $46\%(300$ nodes$)$ and $65\%(500$ nodes$)$ better than CQoSS. However, when the number of node increases, end to end delay for the new protocol is slightly increased.

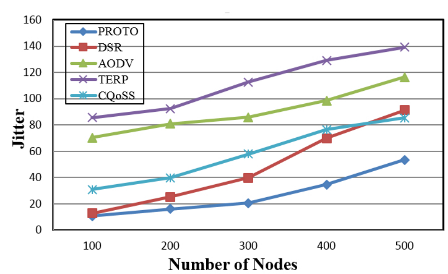

Fig. 3. Delay Variance

B. Delay Variance

The delay variance is measured by calculating the min value of maximum and minimum delays for a packet at each path from source to destination.

Figure 3 shows the compared protocols' delay variance, where the new protocol performs better than $TERP$, $AODV$, DSR and $CQoSS$ [7], [1], [3]. When the number of node increases and intervals of the packet transmission are not high enough, the jitter value will also increase. The output result shows that the delay variance will decrease when the number of node is low for $AODV$, but when the number of node increases, DSR and the new protocol performs better. The new protocol shows 35%(300 nodes) and 37%(500 nodes) better result than $CQoSS$.

C. Average Packet Delivery Ratio (PDR)

Average packet delivery ratio stands for number of packets transmitted by the source node divided by number of transmitted packets successfully delivered.

$$PDR = \frac{\sum_{i=1}^{n} P_{success}}{\sum_{i=1}^{n} P_{trans}} \times 100 \quad (14)$$

where $P_{success}$ denotes successfully delivered packet and P_{trans} indicates number of packet transmission. Figure 4 shows the result for packet delivery ratio. When the number of node is less or equal to 100, $AODV$ [1] performs good. However, with the increased node number, the delivery ratio decreases continually for the other protocols, specially $AODV$, because of bulky traffic. Selection of all possible paths to transmit data from source to the destination is based on link quality. Hence, when the number of nodes increases, the new protocol and DSR performs well.

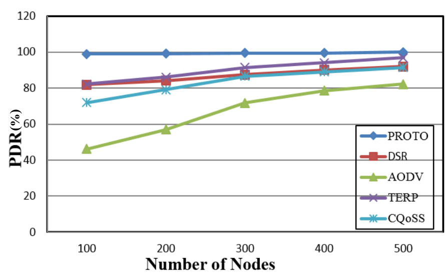

Fig. 4. Average Packet Delivery Ratio

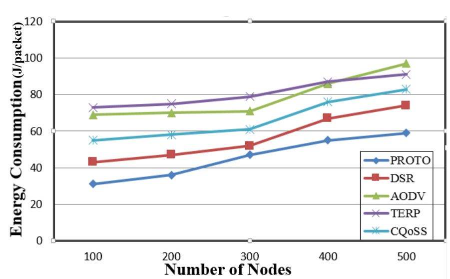

Fig. 5. Energy Consumption per Delivered Packet

D. Energy Consumption per Delivered Packet

The average energy consumption is the energy consumption by the nodes, which are participating during data message transmission from source to sink. Equation (15) represents energy consumption per packet delivery.

$$\frac{\sum_{i=0}^{n} E_{con}}{\sum_{i=0}^{n} P_{trans}} \quad (15)$$

Where, E_{con} denotes energy expended by each node. Figure 5 shows the results for the energy consumption. During routing discovery $TERP$, $AODV$, DSR [7], [1] both flood the $RREQ$ message, for that reason the energy consumption is too high. When the network node count is less than 300, $AODV$ performs well. When the number of node in the network is high, then the new protocol consumes less energy and the behavior is more stable with a little impact compared to AODV and DSR. The new protocol consumes energy less than 49%(300 nodes) to 60%(500 nodes) than DSR. The simulation result shows that the new protocol $(PROTO)$ consumes less energy than $CQoSS$ [3] as well.

V. Conclusion and Future Works

In this paper we present an energy efficient, delay aware multi-path routing protocol for wireless sensor networks. Recovery from node failure is achieved by using multi-path technique. The prime metrics chosen to design the routing protocol are signal-to-noise ratio (SNR) and jitter. Simulation results show improvement of our proposed technique compared to some other existing works. Possible future directive could be studying other performance metrics such as throughput, control overhead, and multi-destination scheduling for routing.

References

[1] A. A. S Ihbeel, H. I. Sigiuk and A. A Alhnsh", *Simulation Based Evaluation of MANET Routing Protocols for Static WSN*, IEEE 2nd International Conference on Innovative Computing Technology, pp: 63-68, 2012.

[2] B. Yahya and J. Ben-othman, *REER: Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks*, Proceedings of IEEE Global Telecommunications Conference (GLOBECOMM), pp. 1 - 7, November 2009.

[3] H. Alwan and A. Agarwal, *A Cross-Layer-Based Routing with QoS-Aware Scheduling for Wireless Sensor Networks*, ACS International Conference on Computer Systems and Applications, pp. 1 8, May 2013.

[4] H. Alwan and A. Agarwal, *Multi-objective QoS routing for wireless sensor networks*, International Conference on Computing, Networking and Communications (ICNC), pp. 28 - 31, January 2013.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, *Wireless sensornetworks: a survey*, Computer Networks, Vol. 38, No. 4, pp. 393422, March 2002.

[6] J. Shin, M. Chin, C. Kim, *Optimal Transmission Range for Topology Management in Wireless Sensor Networks*, Information Networking, Vol. 3961, pp. 177-185, 2006.

[7] M. Almasri, K. M. Elleithy,A. Bushang andR. Alshinina, *TERP: A Trusted and Energy Efficient Routing Protocol for Wireless Sensor Networks(WSNs)*, IEEE 17th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), pp. 207 214, October 2013.

[8] Ye Ming Lu and V. W. S. Wong, *An energy efficient multipath routing protocol for wireless sensor networks*, International Journal of Communication System, Vol. 20, pp. 747-766, 2007.