

## Energy Aware and Link Quality Based Routing in Wireless Sensor Networks under TinyOS-2.x

Dhaval Patel,<sup>1</sup> Bijal Chawla,<sup>2</sup> Chandresh Parekh<sup>3</sup>

<sup>12</sup>PG Student, Department of Wireless Mobile Computing, Gujarat Technological University, India

<sup>3</sup>Head, Department of Electronics and Communication, Government Engineering College, Gujarat, India

**Abstract:** In WSN the size of sensor node is too small and in future it will become smaller and energy is one of the constraints in sensor node in WSN. In sensor node energy required to process data, transmitting data and receiving data. Communication requires more energy than processing data. Tiny OS is embedded operating system in sensor node. In Tiny OS 2.x there is routing protocol called collection tree protocol. Collection tree protocol finds the route according to link quality. But residual energy is key factor in sensor node. The proposed routing protocol improves routing mechanism in collection Tree protocol by energy based routing. The proposed algorithm (ELR) was compared with CTP protocol in terms of packet reception ratio (PRR) and number of nodes alive ratio of the network. This work was simulated in Castalia 3.2 and TOSSIM simulator. The results show that our algorithm performs better than CTP in terms of load distribution and increased lifetime.

**Keywords:** Collection Tree Protocol, Energy Aware Routing Protocol, Energy Measurement, Tinos, TOSSIM, Wireless Sensor Networks

### I. INTRODUCTION

Wireless sensor network is consisting of potentially large number of tiny sensor node communicating with each other for performing some task. The task is defined by some application like environmental monitoring, habitat monitoring, and health monitoring and home automation. The main purpose of sensor node is to sense data, process data and sends the data to sink node. Sink node is the node which is gathering information from sensor nodes. For doing this task sensor node requires energy. This energy is provided by battery. The battery has limited battery power. If the battery is gone down then sensor will not work longer. The nodes are mostly unattended hence the battery power is more precious<sup>[1]</sup>.

WSN nodes require one operating system to work. Tiny OS is one of the operating system for WSN node. Tiny OS is a free and open source component-based event driven operating system, which addresses these issues of the WSN nodes. Tiny OS is implemented using nes C (Network Embedded System C) language, which supports the event based concurrent model of Tiny OS<sup>[2][6]</sup>. Tiny OS provides components for packet communication, sensing, scheduling, routing and medium accessing etc. The routing protocols supported by this operating system is Collection Tree Protocol (CTP), it estimates the quality of the link based on either one of the link estimator protocols such as LEEP or four-bit link estimator.

In CTP protocol decides the parent node merely based on the link quality and thus a node with good link will always be selected as the parent node. It is apparent that a node with good wireless link will involve in more communication and be drained out quickly. These nodes will be exhausted soon and the network will be disconnected. Thus the balancing the traffic among the nodes is necessary and this can be archived by considering the residual energy of the node as one the metric in the routing strategy. In this paper we proposed a protocol (ELR), will choose the parent by considering link quality and residual energy of the node so, we can improve the lifetime of network.

The proposed algorithm (ELR) is implemented in TinyOS-2.X and tested with TOSSIM simulator, which can run the actual Tiny OS code without any real nodes.

The rest of the paper is organized as follows: Section 2 gives introduction about Collection Tree protocol, Section 3 is survey of existing routing protocol, Section 4 describes proposed protocol and algorithm, Section 5 is performance analysis of proposed protocol (ELR), Section 6 gives the conclusion.

### II. COLLECTION TREE PROTOCOL

CTP is routing protocol implemented in TinyOS-2.x. CTP uses routing messages (also called beacons) for tree construction and maintenance, and data messages to report application data to the sink. The standard implementation of CTP described in and evaluated in consists of three main logical software components: the Routing Engine (RE), the Forwarding Engine (FE), and the Link Estimator (LE). In the following, we will focus on the main role taken over by these three components.

**Routing Engine**<sup>[10]</sup>. The Routing Engine, an instance of which runs on each node, takes care of sending and receiving beacons as well as creating and updating the routing table. This table holds a list of neighbors from which the node can select its parent in the routing tree. The table is filled using the information extracted from the beacons. Along with the identifier of the neighboring nodes, the routing table holds further information, like a metric indicating the "quality" of a node as a potential parent.

In the case of CTP, this metric is the ETX (Expected Transmissions), which is communicated by a node to its neighbors through beacons exchange. A node having an ETX equal to  $n$  is expected to be able to deliver a data packet to the sink with a total of  $n$  transmissions, on average. The ETX of a node is defined as the “ETX of its parent plus the ETX of its link to its parent”. More precisely, a node first computes, for each of its neighbors, the link quality of the current node-neighbor link. This metric, to which we refer to as the 1-hop ETX, or ETX1hop, is computed by the LE. For each of its neighbors the node then sums up the 1-hop ETX with the ETX the corresponding neighbors had declared in their routing beacons. The result of this sum is the metric which we call the multi-hop ETX, or ETXmhop. Since the ETXmhop of a neighbor quantifies the expected number of transmissions required to deliver a packet to a sink using that neighbor as a relay, the node clearly selects the neighbor corresponding to the lowest ETXmhop as its parent. The value of this ETXmhop is then included by the node in its own beacons so as to enable lower level nodes to compute their own ETXmhop. Clearly, the ETXmhop of a sink node is always 0.

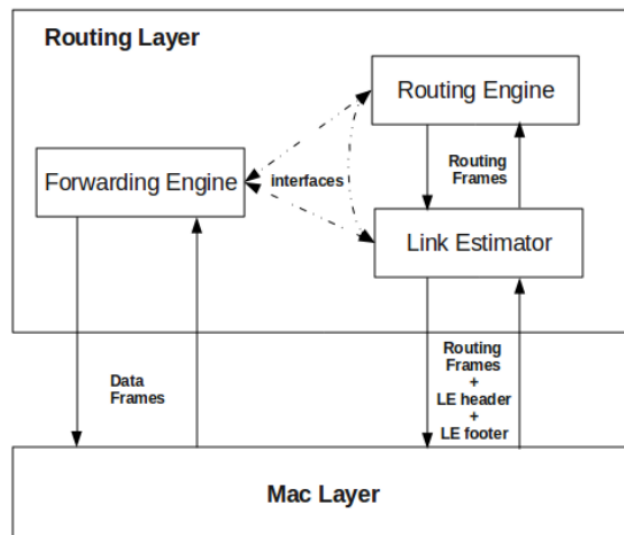


Fig. 1. Message flow and modules interaction [10]

**Forwarding Engine** <sup>[10]</sup>. The Forwarding Engine, as the name says, takes care of forwarding data packets which may either come from the application layer of the same node or from neighboring nodes. The FE is also responsible of detecting and repairing routing loops as well as suppressing duplicate packets.

**Link Estimator** <sup>[10]</sup>. The Link Estimator takes care of determining the inbound and outbound quality of 1-hop communication links. The LE computes the 1-hop ETX by collecting statistics over the number of beacons received and the number of successfully transmitted data packets. From these statistics, the LE computes the inbound metric as the ratio between the total number of beacons sent by the neighbor over the fraction of received beacons. Similarly, the outbound metric represents the expected number of transmission attempts required by the node to successfully deliver a data packet to its neighbor.

### III. RELATED WORK

There are lots of researches done in improving collection tree protocol but only some of the protocols are addressing real life deployment of wireless sensor network. In literatures we found that routing decision is taken based on residual energy of only parent node. CTP is a pre-active routing protocol so energy value is not bound to only parent node but need to consider all the nodes in between sink node and source node. In our work we use min-max battery cost routing <sup>[8]</sup> to select route to sink node. And we used ELQR protocol as based paper for our research, ELQR protocol select parent based on 1-hop energy metric, we modify ELQR by multi-hop energy metric parameter because CTP is pre-active routing protocol so routing decision must be based on all hops in route.

#### 3.1. RLQ: Resource Aware and Link Quality Based Routing Metric[3]

RLQ is routing protocol, for wireless sensor network. In RLQ routing decision is based on the link cost which is the sum of normalized energy cost for the transmitter and receiver. The energy cost depends on the energy consumption for transmission and reception, residual energy and the link quality. The weighing factors  $x$  and  $y$  are used to change the routing decision. If both are zero, the routing decision is based on the minimum hop count. If  $x=1$  and  $y=0$ , the minimum total energy consumption path is the shortest path. If both  $x$  and  $y$  are 1, routing metric is based on both link quality and the residual energy of the node. This algorithm was tested in the test bed consisting of 21 Tmote sky nodes and the performance of RLQ is tested with shortest path algorithm, and LQI in terms of PRR, throughput and lifetime of the network.

### 3.2. LQER(Link Quality Estimation based Routing Protocol)<sup>[4]</sup>

The LQER protocol is a routing protocol with the objective of reliable data delivery in an energy efficient manner. LQER protocol makes the path selection based on the historical states of link quality after minimum hop field is established. In this paper, a dynamic window concept (m; k) is used to record the link historical information and which integrates the approach of minimum hop field. Here 'm' is the number of successful transmission over the link for the window size of 'k' previous transmissions.<sup>[6]</sup>

It uses Minimum Hop field establishment algorithm to find the minimum hop neighbors using flooding approach and each node will have the list of forwarding neighbors. Each node chooses its next hop candidate from this list, which are having best m/k value.

The LQER protocol is simulated in WSNsim for Mica2 platform and the performance parameters considered are energy efficiency and the packet success rate. The authors claim that the proposed algorithm performs better in terms of energy efficiency, packet loss rate and scalability.

### 3.3. Efficient and Multi-path protocol for Wireless Sensor Network [5]

<sup>[6]</sup>This protocol is designed for ensuring the QoS requirement of multimedia transmission such as throughput, delay etc. It is based on directed diffusion, which is scalable and uses single energy efficient path for data transmission. In multipath routing, disjoint multiple paths are chosen which are based on the link quality and the delay incurred in the path. The path cost for the selection of forwarding candidate is based on cumulative path\_ETX and path\_Delay. To get high throughput and low delay paths, the path cost is defined as

$$\text{Path\_Cost} = \text{Path\_ETX}^a * \text{Path\_Dealy}^b$$

<sup>[6]</sup>The ETX based link quality is estimated from SNR value of the received packets and is calculated according to [12] and is given by  $\text{ETX} = 1 / (\text{df} * \text{dr})$  where df and dr are the forward and reverse packet delivery ratio of the particular link. It is simulated in NS2 and provides high throughput and less delay, which are the requirement for the high quality video transmission

### 3.4. Energy And Link Quality Based Routing Tree (ELQR)<sup>[6]</sup>

In this paper, a protocol, takes the routing decision based on the link quality and the residual energy of the nodes. Each node will make one of its neighbours as its parent if the neighbor node has sufficient energy to forward the packet and the quality of the link to that node is good. A threshold is said for both parameters and the energy threshold is set well above 2V such that the lifetime of the node will be increased by reducing its load. The link quality increases with hop count and the node with less value will be chosen as the forwarding candidate. In this work, the threshold for the link quality is changed throughout the lifetime of the network to make the network alive by selecting the longer routes with more energy. Hence the routes with more energy and adequate wireless link quality will be chosen for the forwarding operation without exhausting the nodes in the optimal path. This algorithm is simulated on TOSSIM and it is tested on the test bed comprising of IRIS motes.

## IV. ENERGY AWARE AND LINK QUALITY BASED ROUTING PROTOCOL (ELR)

This section outlines a routing protocol based on energy aware and link quality. ELR uses the link quality of wireless links and residual energy during the routing selection process to increase the system's reliability and assure QoS. It also provides mechanism to provide load balancing and avoid premature death of nodes/networks. We can improve current CTP by broadcasting new parameter called energy and maintaining this parameter to routing table for creation of tree. The sink node is data gathering node and it broadcasts control packet periodically. It includes link quality parameter called ETX (Expected transmission) and residual energy.

ETX value is measured by link estimator engine. ETX of sink node is 0, because sink node does not have any parent node. Initially the energy of node is 100%. We take energy value in percentage because it is simple to carry percentage than joules value. Joule is unit of energy. To store and carry percentage value required only 2 bytes. This two parameter ETX and Residual energy is added in control packet. Sink node broadcast the control packet containing ETX value and energy value. Figure 2 shows the procedure followed by the sink node for broadcasting a control packet. Sink node creates routing frame with ETX value, Energy value, etc. and broadcast control packet to its neighbor nodes.

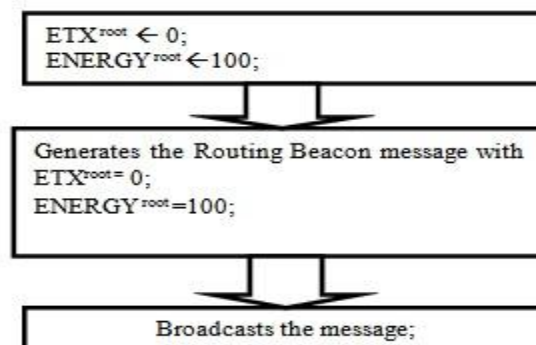


Fig2.Control Packer Broadcast

Figure 3 shows upon reception of control packet Sensor node (receiver) node extract control packet and get value of ETX and energy change their routing table entry to maintain tree. Sensor node(receiver) check the entry of sender node in routing table if node is present in routing table then only update the entry if node is not present in routing table then sensor node add entry to its routing table. Then after the sensor node will broadcast control packet to its neighbor nodes by obtaining new value of energy and ETX. Here energy is set to minimum energy of all nodes in route. This is similar to min-max battery cost routing<sup>[8]</sup>. Sensor node create control packet which contains information of ETX value and least energy value of all nodes in between to sink node.

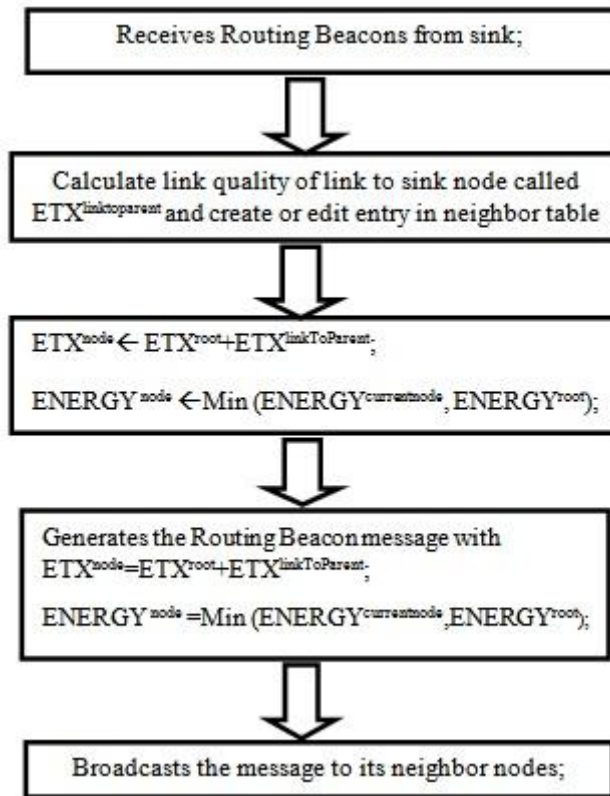


Fig.3. Control Packet Broadcast by sensor node

The limitations of node hardware and the quality variation of wireless links are great challenges to providing high service availability, especially in WSN applications, where it is necessary to create mechanisms that can identify and mitigate or solve the energy hole problem. Energy hole can be caused by congestion or overuse of route leading to the premature death of nodes. Routing solutions must be employed load balancing mechanisms that are able to divert traffic and thus increase QoS (less packet loss and delay rate) and reduce energy consumption. We create an algorithm to select parent based on two metric namely ETX and energy.

The best ETX route is selected by following algorithm:

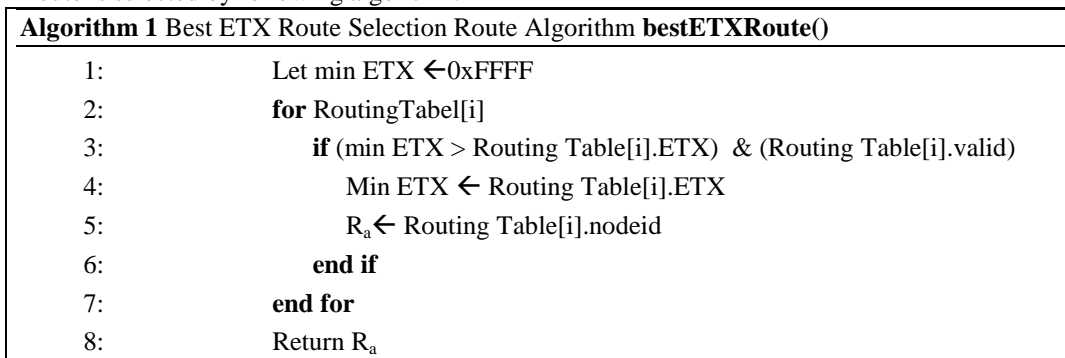


Fig.4. Algorithm for best ETX Route

In CTP route is selected based on only ETX value but we use ETX and residual energy of nodes to select best route. Algorithm 1 is used in CTP for finding best ETX path. We have used Algorithm 1, 2 and 3 for route selection.

The best Energy route is selected by following algorithm:

```

Algorithm 2 Best Energy Route Selection Route Algorithm bestEnergyRoute()
1:          Let maxEnergy ← 0
2:          for RoutingTabel[i]
3:              if (maxenergy < Routing Table[i].energy) & (Routing Table[i].valid)
4:                  maxEnergy ← Routing Table[i].energy
5:                  Rb ← Routing Table[i].nodeid
6:              end if
7:          end for
8:          Return Rb
    
```

Fig.5. Algorithm for best Energy Route

The proposed algorithm (ELR) will select route based on link quality and residual energy of nodes. Algorithm 3 is proposed ELR protocol which select route to sink node.

In our algorithm, routing table has addition entry of energy which indicates least energy in route and control packet also carries this information. The algorithm searches for the node with the highest least energy node route among all the routes with minimum ETX, then that node will be taken as parent node. We used two threshold parameters  $E_{th}$  and  $ETX_{diff_{th}}$ .  $E_{th}$  is an energy threshold and  $ETX_{diff_{th}}$  is ETX difference threshold. After selection of best ETX route and best energy route, algorithm find difference ETX. If difference ETX ( $ETX_{diff}$ ) is less then  $ETX_{diff_{th}}$  then best energy route will be selected. Otherwise further check for best ETX route has efficient energy to complete cycle of task. If best ETX route has energy greater then energy thresh hold, then best Energy route is selected otherwise best ETX route is set to invalid and it will be removed from selection of parent node and repeat cycle till algorithm does select a best route.

```

Algorithm 3 Selection Route Algorithm of ELR Protocol
1:          Let  $ETX_{diff_{th}}$  ← Expected Transmission Threshold
2:          Let  $E_{th}$  ← Energy Threshold
3:          Let Ra ← bestETXRoute()
4:          Let Rb ← bestEnergyRoute()
5:           $ETX_{diff}$  ← Rb.ETX – Ra.ETX
6:          if Ra=Rb then
7:              SelectedRoute ← Ra
8:          else if  $ETX_{diff} \leq ETX_{diff_{th}}$  then
9:              SelectedRoute ← Rb
10:         else
11:             if Ra. energy ≤  $E_{th}$  then
12:                 Ra. valid ← False
13:                 Repeat step 3
14:             end if
15:         end if
    
```

Fig.6 Algorithm for ELR

### V. IMPLEMENTATION OF ELR ALGORITHM

We implemented this algorithm in Tinyos-2.x in module ~\tos\lib\net\ctp. The energy consumption by sensor node is very challenging task because there is no tool to calculate energy consumption. We calculate remaining energy of nodes by software method. The energy used by sensor node is calculated runtime by tracking the time spent by components such as microcontroller, radio, sensor and memory. Each component consumes current. Energy consumption is calculated as  $Energy = Current * Voltage * Time$

Current and voltage depends on which mote we are using. In our implementation we take mote users manual<sup>[9]</sup> to calculate energy. In this manual we get value of current required by different modules (microcontroller, radio, sensor and memory). Figure 6<sup>[9]</sup> will show the system specification of all component of mote. Figure 7 shows the current required by components with duty cycle. Mote operates on 3V and radio in transmission mode take 12mA. If radio active for 1 ms then energy consumption is

$$\begin{aligned}
 \text{Energy} &= (12 * 10^{-3}) * (3) * (1 * 10^{-3}) \text{ joule} \\
 &= 36 \mu\text{joule}.
 \end{aligned}$$

So like this we calculate energy spent by all components and deduct spent energy from initial energy to get remaining energy of node. For 2AA type battery capacity is 2000 joules initially.

SYSTEM SPECIFICATIONS		
Currents		Example Duty Cycle
<b>Processor</b>		
Current (full operation)	8 mA	1
Current sleep	8 μA	99
<b>Radio</b>		
Current in receive	8 mA	0.75
Current transmit	12 mA	0.25
Current sleep	2 μA	99
<b>Logger Memory</b>		
Write	15 mA	0
Read	4 mA	0
Sleep	2 μA	100
<b>Sensor Board</b>		
Current (full operation)	5 mA	1
Current sleep	5 μA	99

Fig. 7. System Specification.<sup>[9]</sup>

We add new parameter entry called energy in routing frame as well as in routing table. Figure 8 shows the routing frame with new parameter added energy.

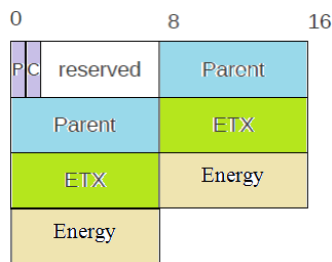


Fig. 8. Routing Frame.

The CtpRoutingEngine.nc is modified to implement ELR algorithm. We create new method Consumed Energy () to calculate consumed energy, which returns residual energy of mote and this is called periodically by the routing engine. The energy value is sent with the control packet and the nodes which receives control packet, will update their routing table with the updated energy value. The routing decision is modified according to ELR algorithm in the Update Route task.

The energy Threshold is set to 10%. When nodes have battery capacity less than 10% it will not work as intermediate node but it is able to send its own packet till death. The ETXdifTh is set to 10. We can also change this threshold value because it is about link quality.

## VI. PERFORMANCE ANALYSIS

We use TOSSIM simulator for test proposed protocol under Linux. TOSSIM is simulator for tiny OS developed by university of California at Berkeley, USA, Which can run the actual tinyOS code without any motes<sup>[7][6]</sup>. The hardware components are simulated using software at packet level and code developed for this simulator can be directly fused into the motes with minimum changes.

The simulation parameter used for creating the network topology is given below. The nodes are placed in uniform topology.

Parameter	Value
No. of Nodes	100
Topology	Uniform, Random
Size	500*500
Simulation Time in Seconds	1000,2000,4000,6000,8000,10000
Radio Model	Indoor
Initial Battery Capacity	100mA-hour

Table 1. Simulation Parameters



**6.1. Simulation Results**

The figure 9 shows the PRR value for two protocols with the simulation time from 1000 to 10000 seconds. X-axis shows the simulation time and Y-axis shows the packet reception ratio. The PRR value of proposed scheme is compared to the CTP protocol. The PRR value of the CTP protocol is almost same as ELR initially, but decreases as time advances. This is due to high link quality nodes are dead as time passed. But in ELR protocol load is distributed among all nodes, so we can get high packet reception ratio than CTP.

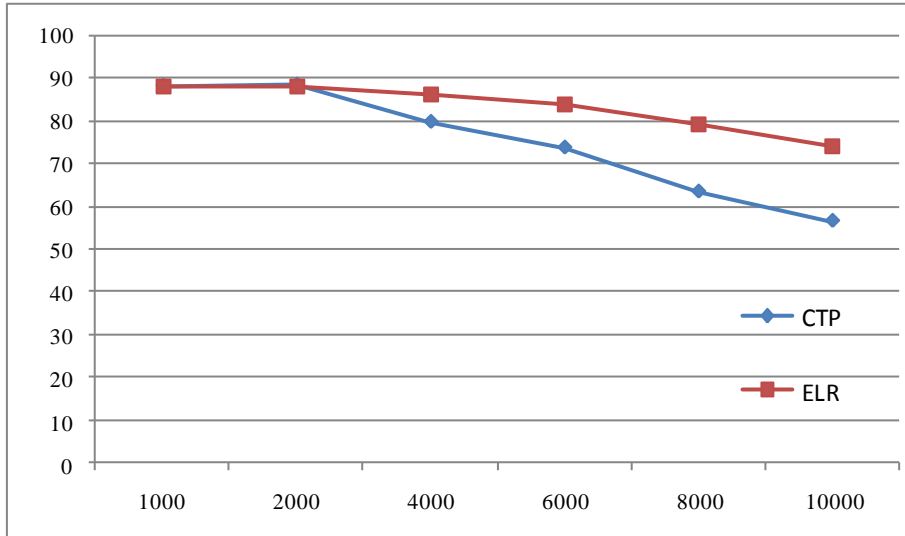


Fig. 9. Packet Reception Ratio

The Number of node alive over the simulation time is measured and shown in the figure 10. X-axis shows the simulation time and Y-axis shows the number of nodes alive. The time at which first node dies, is also calculated for CTP and ELR which are 2498 and 4596 seconds respectively. This result shows that network life is increased due to load balancing in ELR protocol.

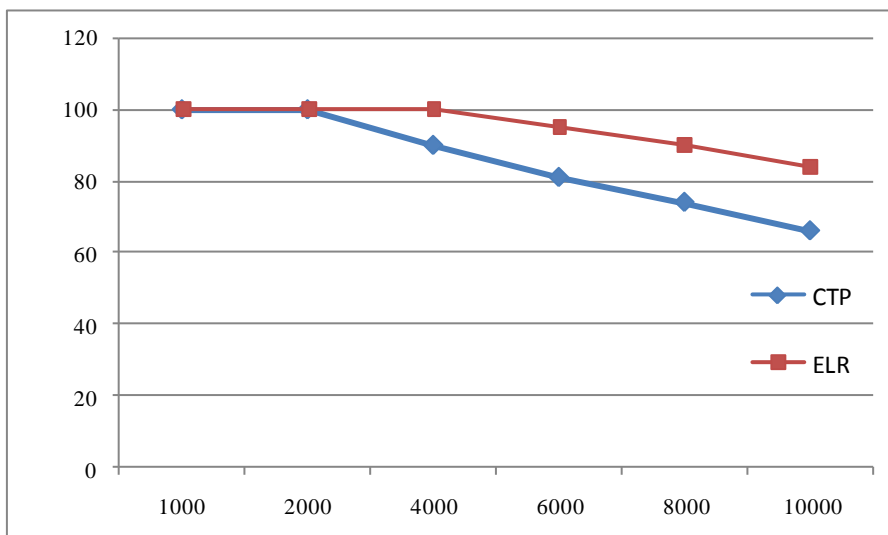


Fig. 10. Number of nodes alive

Figure 11 shows control packet overhead ratio value for two protocols for the simulation times from 1000 to 10000 seconds. X-axis shows the simulation time and Y-axis shows the control packet overhead ratio. In ELR control packet overhead is greater than CTP. This is due to receiving more recent updated energy value.

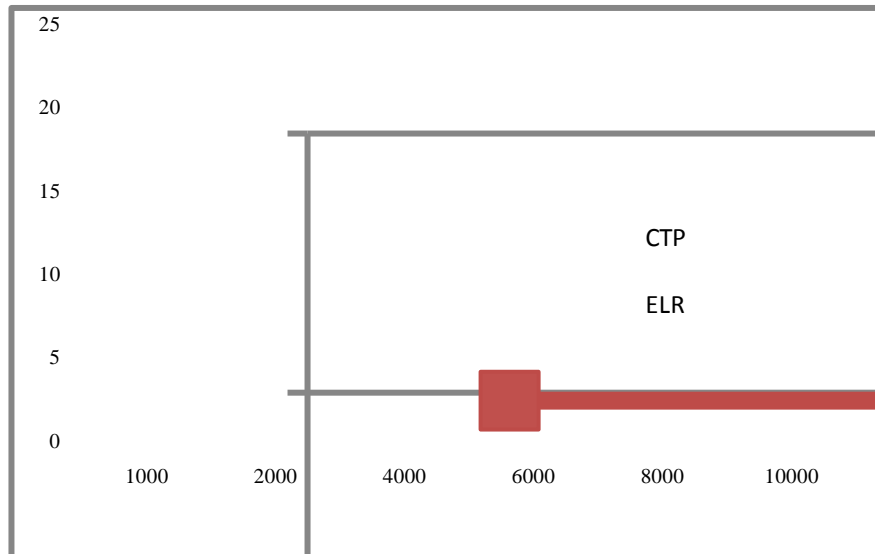


Fig. 11. Control Packet Overhead Ratio

Figure 12 shows the data packet retransmission ratio. X-axis shows the simulation time, Y-axis shows the retransmission ratio. The data packet retransmission increases as time is increased. The data packet retransmission ratio of CTP is almost same as ELR initially, but increases as time advances. This is because the node dies as time passes. Due to increased retransmission ratio delay increased in reception. Quality of service decreases if delay is increased.

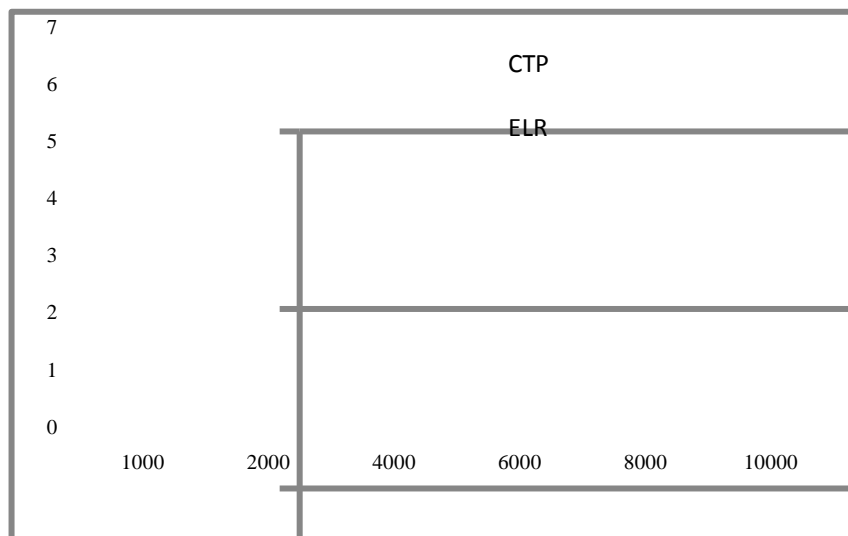


Fig. 12. Data Packet Retransmission Ratio

## VII. CONCLUSION

Energy is an important resource in WSN. To increase network lifetime, the traffic load should be distributed among the forwarding nodes in such a way that they could be alive for a long time. In this paper, we proposed a protocol, which take the routing decision based on the link quality and the energy of nodes. Every node contains energy entry in routing table which indicates least energy node in route. We got this idea from min-max battery cost routing. Each node will make one of its neighbors as its parent if the parent has route to sink having efficient energy and good link. We found in our result that ELR protocol increases network lifetime by load balancing among forwarding nodes. The result shows that PRR is increased, Data packet retransmission is decreased and more number of nodes are alive than CTP with some overhead of control packet. The limitation of ELR and CTP is that PRR decreases if nodes are mobile.



### REFERENCES

- [1] I.Akyldiz, W. Su. Y.Sankarasubramanian and E. Cayirci(2000), "A Survey on Sensor Networks", IEEE Communication Magazine Vol. 40, No. 8, pp. 103 -114.
- [2] P. Levis, S. Madden, D. Gay, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer and D. Culler (2005), "TinyOS: an operating system for sensor networks", Ambient Intelligence, Springer-verlag, pp. 115–148.
- [3] V. Cagri Gungor, Chellury Sastry, Zhen Song and Ryan Integlia(2007), "Resource-Aware and Link Quality Based Routing Metric for Wireless Sensor and Actor Networks", IEEE International Conference on Communications, ICC '07, pp. 3364-3369.
- [4] Chen J., Lin R., Li Y., Sun Y. (2008), "LQER: A Link Quality Estimation based Routing for Wireless Sensor Networks", Sensors. vol. 8, issue. 2, pp. 1025-1038.
- [5] Shuang Li, Raghu Kisore Neelisetti, Cong Liu, and Alvin Lim (2010), "Efficient Multi-path protocol for Wireless Sensor Networks", Int. Jr. of Wireless & Mobile Networks, vol.2, No.1, pp. 110-130
- [6] K. Pavai, D. Sridharan, S.A.V. Satya Murty A. Sivagami, "Energy and Link Quality Based Routing for Data Gathering Tree in Wireless Sensor Networks Under TINYOS -2.X," International Journal of Wireless & Mobile Networks, vol. 2, no. 2, pp. 47-60, 2010.
- [7] Philip Levis, Nelson Lee, Matt Welsh, and David Culler (2003), "TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications." In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003) pp. 126-137
- [8] C-K. Toh, Hiroshi cob, and David A. Scott, "Performance Evaluation of Battery-life-Aware Routing Schemes for Wireless Ad Hoc Network" 2001 IEEE
- [9] Mote Processor radio &Mote Interface boards user manual. Document part number: 7430-0021-09 Rev A [online:] [http://www.cs.wmich.edu/gupta/teaching/cs5950/fall2011/mica%20pinouts%20from%20memic%20mpr-mib\\_series\\_users\\_manual%207430-0021-09\\_a-t.pdf](http://www.cs.wmich.edu/gupta/teaching/cs5950/fall2011/mica%20pinouts%20from%20memic%20mpr-mib_series_users_manual%207430-0021-09_a-t.pdf)
- [10] Collection Tree Protocol for Castalia for wireless sensor network simulator [Link:] [https://ctp-castalia.googlecode.com/files/colesanti\\_santinis\\_2011\\_ctp-castalia.pdf](https://ctp-castalia.googlecode.com/files/colesanti_santinis_2011_ctp-castalia.pdf)