Indian Journal of Science and Technology, *Vol 9(5), DOI: 10.17485/ijst/2016/v9i5/84916, February 2016*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Establishing Path Quality Management in Wireless Sensor Networks through Cluster Head Determination

G. Sirisha*, R. Bulli Babu and K. Raghava Rao

Department of Electronics and Computer Engineering, K L University, Vaddeswaram, Guntur District - 522502, Andhra Pradesh, India; gubbalasireesha6@gmail.com, babuklu123@kluniversity.in, raghavarao@kluniversity.in

Abstract

Because of its huge scale and restrictions on scope of communication, a Wireless Sensor Network (WSN) generally depends on multi-hop transmissions to transmit a data packet along a succession of storage nodes. It is of key significance to estimate the forwarding nature of multi-hop routes and such data might be used in developing dynamic and economical routing methodologies. **Objective:** To develop a scheme which eases the determination of Cluster Heads (CH) in a WSN, which inturn would enhances the quality of path management. **Analysis:** Existing measurements for the most part, concentrate on evaluating the connection link performance among the sensor nodes while ignoring the forwarding abilities inside them. Recent studies demonstrate that the nature of forwarding inside every wireless sensor node is equally essential component that adds to the path quality in data transmission. **Methodology:** We propose a methodology to reduce the overhead by developing an efficient scheme for selecting a Cluster Head (CH) in each cluster to represent all the other nodes in the cluster. This CH will be responsible for representing the entire Cluster viz., To determine the routing path ,To detect the malicious nodes (Attacker Node) in each cluster, etc., which eventually eases the task of determining the forwarding path, in other words, enhancing the forwarding path quality. **Findings/Improvements:** A stand-alone application had been developed for this proposed system in Eclispe IDE and the results we obtained has proved that the proposed methodology ,when applied can achieve the required forwarding path quality management in WSN by evaluating both the connection link performance and also the forwarding abilities inside a sensor node.

Keywords: Clustering, Multi Hop Routes, Path Quality, Quality of Forwarding, Wireless Sensor Networks

1. Introduction

A Wireless Sensor Network (WSN) is regularly planned to compass in an expansive field for information collection. Data transmission is generally accomplished with multi-hop transmission technique along a succession of wireless sensor nodes. An efficient scheme need to be incorporated into the WSN because there are a number of constraints on resources like energy, bandwidth, storage capacity of the node, computational speed, etc. While designing a routing protocol for a large scale WSN, these constraints are to be dealt with utmost priority as they themselves define the nature and performance of a WSN[1]. Numerous multi-hop routing path determining protocols have been proposed for WSN information collection and they for the most part adapt a unique path analysis measurements to choose "good" paths for transmitting data packets.

There have been numerous assess measurements proposed to measure the forwarding nature of a multi-hop routing path, such as Estimated Transmission Count (ETX)[2], ETF[3]. Existing measurements principally concentrate on assessing the quality of packet delivery on path links among the sensor nodes. The limit of forwarding capacity along a path is evaluated by the total of the forwarding characteristics of the considerable number of connection links on the path. Those link based measurements while mirror the link quantification of the path, in

*Author for correspondence

any case, ignore the forwarding abilities inside the wireless sensor nodes, consequently bringing about a fragmented estimation of the path quality. Utilizing these fragmented path markers will prompt problematic routing choices and corrupted routing performance evaluation.

Similar sub optimal effect has been observed in many large scale WSNs[4,5]. Amid the field test there had been packet drops on a few sensor nodes. They are because of an assortment of reasons, for example, forwarding line flood under high traffic, programming bugs in the protocol execution, and so forth[6,7]. Those sensor nodes, on the other hand, still react with Acknowledgements (ACKs) at the radio equipment, on the receiver side. The matter of concern is that with current path pointers, the incompetence of packet forwarding inside the individual sensor nodes can't be shared among the system, yet there is not a metric to evaluate the packet forwarding quality at every sensor node. As an outcome, the path estimation not always genuinely quantizes the path quality and the data transmission capability is extremely degraded.

The packet drops on the malicious nodes present inherent lack of quality in data transmission[8]. Truly, indeed, even a solitary connection itself can barely accomplish full unwavering quality. ETX over a connection measures the required number of transmissions for effectively conveying a packet, however transmitting the packet at the required number of times does not ensure it will be effectively gotten at the collector end. In real world frameworks, most number of retransmissions are generally determined to a link to avoid sending a packet on a "faulty" link endlessly, that debilitates the limited resources of WSN. The packet ultimately will be dropped by the transmitter after a number of transmission retries[10]. The system is along these lines rendered to be problematic because of both sensor node's lack of path quality and link faultiness. ETX of a path is evaluated as the summation of ETX qualities over all connection links constituting the path. Utilizing this computed ETX for path determination minimizes the transmission cost and accomplishes a high throughput[10]. On the other hand, ETX presumes that end-to-end transmission is dependable which, in any case, is not achieved in practice as we have observed from the above.

For data transmission inside of the network of innate inconsistency, a metric that better measures the information profitability is the measure of effective data transmission to the destination, i.e., information yield[5]. Information yield over the genuine number of data transmissions, measures both transmission cost and additionally accomplished throughput. Existing path ETX cannot measure such a parameter. Therefore, ETX cannot capture path reliability. Routing based on path ETX can never give ideal data transmission path in terms of data yield per transmission.

The contributions are as follows: To begin with, we uncover the restrictions of existing connection link based pointers like ETX in assessing the forwarding quality of the path and routing determination in view of ETX may prompt seriously corrupted data yield.

We propose to go for Path Quality Measurement through Cluster Head (CH) determination which in turn will look after the forwarding capabilities of both the Connection Link and also the Sensor Node[11,12].

Finally, we implement this proposed scheme and based on evaluation of the results using our proposed approach, Throughput is greatly increased while maintaining Low Latency.

The reaming part of this paper is classified as follows. In Section 2 we present our proposed methodology. In Section 3 we explain our system design. In section 4 we present the simulation work we've done. With section 5 we conclude our work. Acknowledgements and References are given in sections 6 and 7 respectively.

2. Proposed Scheme

In this work, we explore the faultiness in both connection links and storage nodes. In our proposed system, each region have separate channel in a network. Thus the network region have been moved into multiple nodes. When multiple nodes are moving on that present region, one node has to be determined as a Cluster Head (CH) in that region[13]. That region is denoted as Channel node, thus the Channel node was notified by CH. Cluster head generates a message to Certificate Authority (CA) regarding channel node. Channel node information can be handled by CA. We determine the CH, which appraises the chances for a packet to pass both a link and a node. After incorporating this modification through the CH, The connection link apart from considering the transmission cost at the transmitter side, additionally considers the data transmission proportion at the receiver end. The CA measures the quality of forwarding inside each node, and it assumes a vital part in separating the malicious nodes. Taking into account the forwarding quality of connection link/Storage

node, we calculate the Forwarding Quality over a path, which measures the path quality and it takes into account both transmission cost as well as, end-to-end data transmission proportion. Consequently utilizing such a metric can incredibly enhance the throughput while having a low latency.

Our proposed scheme adapts path quality management through CH determination and from then it is the responsibility of the CH to establish path quality. The selected CH deals with the forwarding capabilities of both the Connection Link and also the Sensor Node[14]. By this mechanism, we're able to estimate both the transmission cost and the data delivery proportion along the forwarding path.

3. System Design

In this section we'll be dealing with system architecture its corresponding use cases and the system modules through which we implement our proposed scheme.

3.1 System Architecture

Figure 1 shows the architecture of our proposed scheme

The system architecture consists of the following entities:

3.1.1 Certificate Authority (CA)

This entity acts a central power house controlling the CHs in the network. It is this entity that facilitates issuing of public/private keys, certificates and maintaining the CHs, which in turn controls the individual wireless sensor nodes. This CA determines the CHs from the group of clusters. It selects a CH from each and every cluster and from then on that CH will be looking after the entire cluster, ranging from data communication, identifying the malicious nodes in the cluster, determining the routing path that should be taken in order to optimize the path quality, etc.

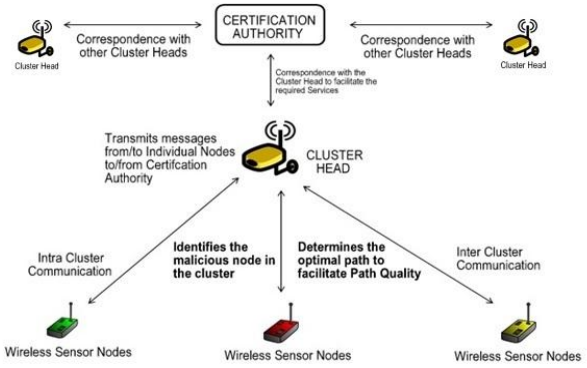

Figure 1. System Architecture.

3.1.2 Cluster Head (CH)

Cluster Head will be selected by the CA for every cluster and it is usually based on a predetermined metric like maximum bandwidth, maximum distance, etc. After CA determines the CH, the entire Cluster is the responsibility of the CH. Required services of the network are facilitated by the CH by acting as a mediator for CA and the sensor nodes. It is this CH which detects the malicious nodes in the cluster and also determines the optimal routing path for data transmission to achieve the path quality.

3.1.3 Individual Nodes

These are the individual nodes in the network. These nodes must be registered with the CA to access the particular network and then later they are clustered as per their qualities and behaviour in the network. It is one of these individual nodes in a cluster that is selected as a CH by the certificate authority.

3.2 Modules

In this sub-section we'll look into the system modules. They are as follows:

3.2.1 User Interface Design

Figure 2 depicts the user interface design. To connect to server user must be registered with the CA. After registration he must log in to the server and can establish the network connection via server. This user account will be used for analysing the data rates in the network.

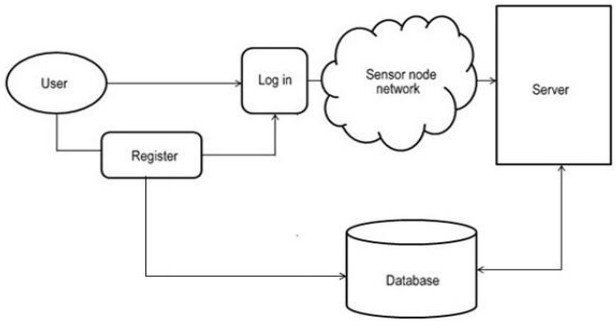

Figure 2. User Interface Module.

3.2.2 Cluster Head Selection

Figure 3 depicts the CH selection. As already stated earlier, all clusters will be in an ad hoc network (WSN). The CH is selected by the CA through a predefined metric. Once the CH is selected it is solely responsible for the entire cluster. It monitors the entire cluster to detect malicious nodes and to determine an optimal routing path ultimately striving for path quality.

3.2.3 Faulty Node Isolation

Figure 4 represents the module for isolating the faulty node in the clusters. After the CH selection is done by the CA, now it is the responsibility of the CH to identify and isolate the faulty node if any, in the network.

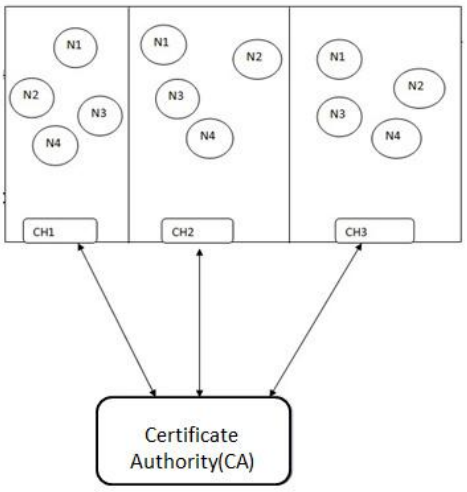

Figure 3. Cluster Head Selection.

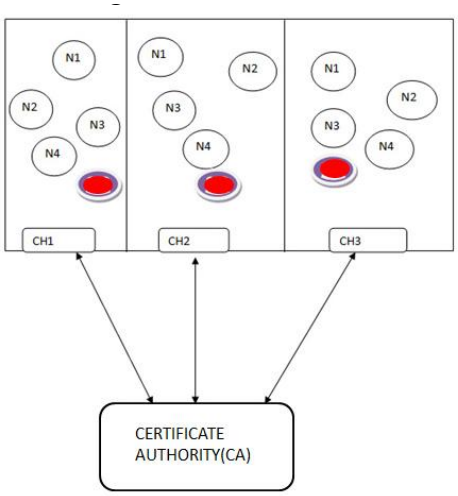

Figure 4. Identifying the faulty node in the cluster.

CH is privileged to be the most trustworthy node in the entire cluster. Each node in the cluster is monitored by its trustee neighbours, called "Verifier Nodes (VN)". A VN is a node that has smaller or equal threshold distance (Td) as that of the node, say node 'n'. It is to be noted that each of the node knows each other's Td when they are in the same cluster. Node 'n' (each node in the network) can play two roles based on the data transmission through it: it can either be a *repeater node* (relays the data from the previous node to the next node) or a *source node* (generates data and transmit to its next node). In both the cases the VNs monitors the node and if there are any abnormalities, notify the CH, which requests the trust parameters from the verifier nodes and recalculates these values for the node 'n'. The CH notifies the new values to the VNs. The VNs update the new values in their routing list. After updating nodes start cooperating with the node 'n', if the Td value of 'n' is lesser than or equal to the neighbours. If the Td value becomes greater than that of the neighbours, that node is faulty one and should be immideately reported to CA by the CH so that it wont transmit data via this faulty node to ensure quality path for forwarding.

3.2.4 Communication between CH and CA

There exist a number of clusters in the network which work under the supervision of one CA. Each CH is notified of all the other CH's in the network by CA. Optimal path quality can be established in the network through communication among the CHs and the CA. Here this communication can be of two types. They are as follows:

3.2.4.1 Intra Cluster Communication

When nodes within the same cluster try to communicate with one another, we call such a communication as intra cluster communication. Here the sender node sends the information to their cluster head, which in turn will forward it to the CA. CA verifies the information whether it is valid or not and then resends it to the CH if it valid, and CH forwards it to the intended destination node in the cluster. If the data that it being transmitted is corrupted, then the node that transmitted will be considered as a faulty node. If the data transmission is a mere relaying one rather than the node generated one, it simply forwards to the next node, although module-3 is implemented here to check whether it is faulty or not to ensure the quality

forwarding of the data. Figure 5 represents module for intra cluster communication.

3.2.4.2 *Inter Cluster Communication*

When two nodes belonging to different clusters decided to communicate with each other such a communication is termed as inter cluster communication. It is represented in Figure 6.

Consider a situation where two nodes N1 and N2 of two different clusters C1 and C2 with cluster heads CH1 and CH2 tries to communicate with one another. Then data transmission has to be done among N1, CH1 and N2, CH2. First N1 sends its message to CH1 which relays it to CA. CA verifies for any corrupted data and if it is free from corrupted data, then it relays it to CH2, which in turn relays to N2.

3.3 Use Case

Use case diagram for system is represented in Figure 7. It consists of three actors, sensor node, CH and CA. Each of these actors has specific tasks. They are:

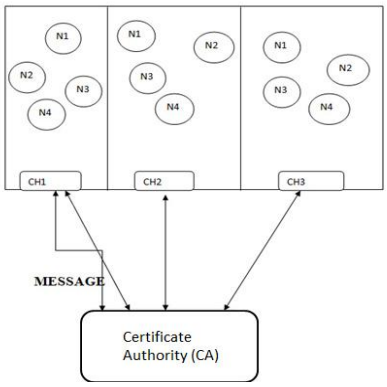

Figure 5. Intra cluster communication.

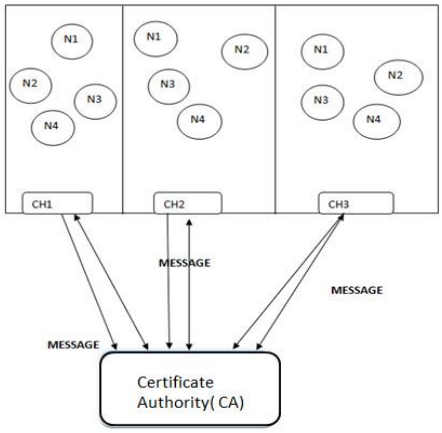

Figure 6. Inter cluster communication.

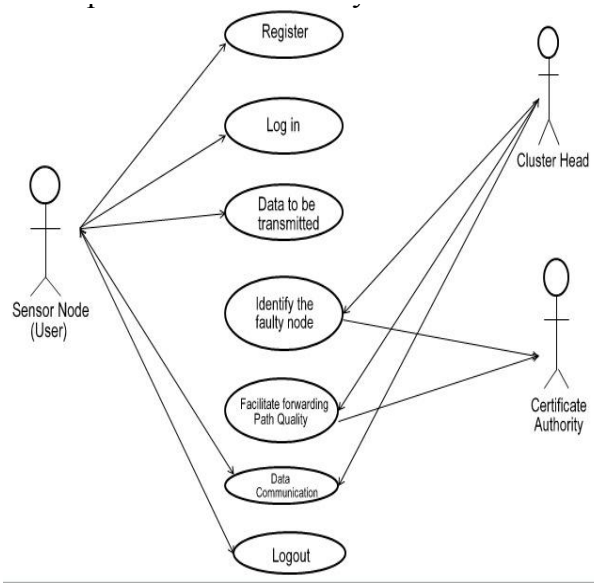

Figure 7. System Use Case.

3.3.1 *Sensor Node*

It is the individual sensor node in the network which is placed in its corresponding cluster. User should first register in the network with his credentials. Then he log-ins to the server and access the services of the network. After CH determines the faulty nodes and a quality path for forwarding, User can transmit the data over the network and communicate with other users.

3.3.2 *Cluster Head*

It takes the whole responsibility of the cluster it is assigned to. It verifies the cluster for any of the faulty nodes and if it founds any it reports to the CA. It verifies the data that is transmitted over the cluster among the individual cluster nodes. It also facilitates intra cluster and inter cluster methodologies for data communication in the network. Considering all the parameters, it determines the quality path for data forwarding.

3.3.3 *Certificate Authority*

It is the central power house. It looks after the entire network. It determines CHs for every cluster based on the predetermined parameters in the network. Each CH will serve the CA and their interaction will define the quality of the network. It has the data base of all the registered users and it decides the fate of data obtained from CHs. If it senses the data is corrupted it notifies the corresponding CH to identify the respective node as faulty.

4. Implementation

We implement Path Quality Management system in eclipse IDE through Java networking. We develop our system as a standalone application in the IDE. User Interface is designed in java applet and AWT techniques and database is designed in MySQL. The system server is the default RMI server (Remote Method Invocation). Using RMI we can establish client/server relationship over the remote machines. As we've developed our application as a standalone application, and require both client (sensor nodes in the network) and server we are going for this RMI technique as this will preserve the object model of the system and also both the client and server can save transmission time by catching up the exceptions and mistakes quite quickly.

Figures 8.1 and 8.2 represents the system registration page with the data base. Only after successful registration any user (sensor node) can access the server and can communicate with other nodes.

Figures 9.1 and 9.2 represents the user login process. After entering the valid credentials, i.e., username and password, user can login to the system, connect to the server for data communication and data access.

After logging in to the system, user must first establish a server connection, this is depicted in the Figure 10, where there are multiple Remote Procedure Calls (RPC) that have established different connections with different clients. We can see the user node "sirisha" connection in the IDE console, which we used for logging in.

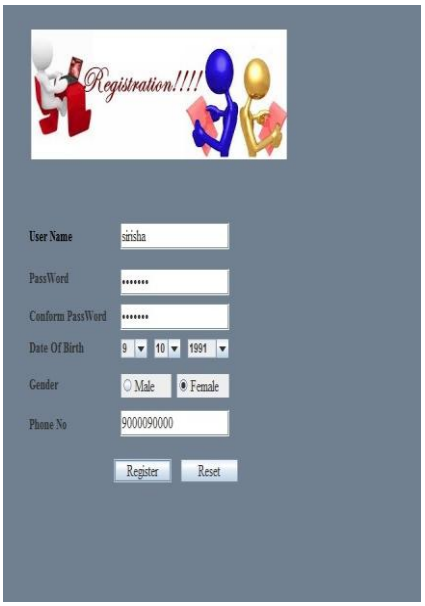

Figure 8.1. User Registration page.

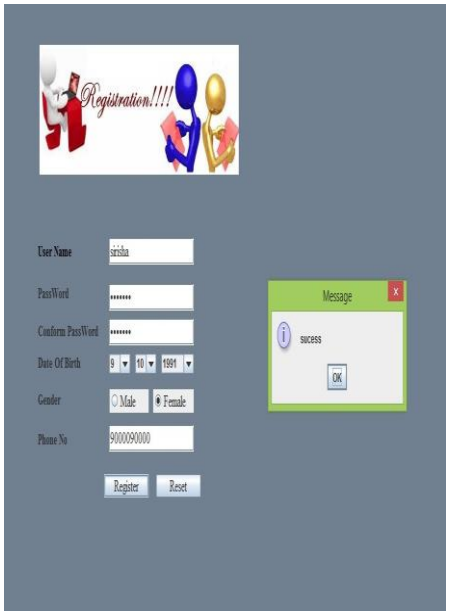

Figure 8.2. User Registration success.

Figure 9.1. User Login page.

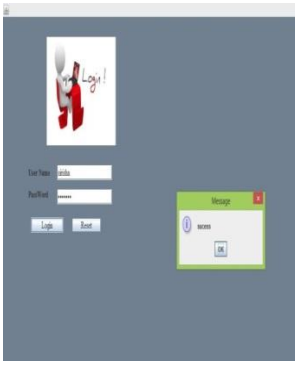

Figure 9.2. User login success.

Figure 11 represents the sensor nodes in the system. Once the data communication is to be taken place, all the nodes becomes mobile, each of these nodes has their own distance tables corresponding to each other. We have

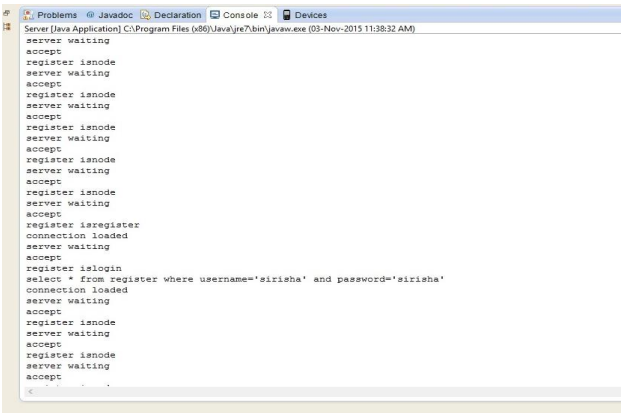

Figure 10. Server connection.

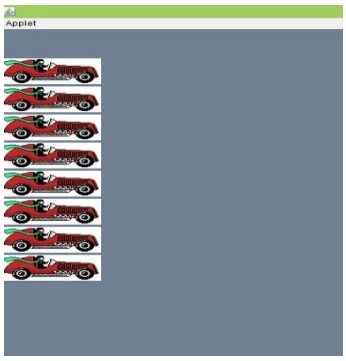

Figure 11. Sensor nodes.

eight moving nodes in our cluster and they are 'imageX', 'image1X', 'image2X', 'image3X', 'MarioX', 'LuigiX', 'BowserX' and 'YoshiX'.

Now after the nodes start the movement, CA needs a CH for maintaining the cluster. The selection of CH is based on the predetermined metric, in this case it is the maximum bandwidth consumed by the node. The reason for this is, if we can determine the node with maximum bandwidth usage, we can safely assume that will be the threshold value for bandwidth of the network. If any node consuming a greater bandwidth than this value, it is identified as the faulty/malicious node (the bandwidth used by each node is in measure of 'bits').

Figure 12.1 represents the moving nodes scenario. We can also see here the distance table for each node with respect to the other nodes. Distance in this case is number of pixels in between two nodes (pixel distance is represented as multiples of 100).

Now the CA needs to select the CH from these nodes to represent the entire cluster which in turn will look after the path quality management in the cluster (In our simulation, CA selects the CH by clicking the 'Cluster Head' button). As already mentioned above, the CH is selected as per the maximum bandwidth usage of the individual nodes. Figures 12.2(i) represents CA selecting the cluster head among the eight mobile nodes in the user interface: 'LuigiX' that is having the maximum bandwidth selected as the CH. Figure 12.2(ii) represents the same in IDE console, the values of each and every node. There are quite a few CH selections and our test case is represented in block rectangular box, while another one is represented in normal rectangular box. We can see the bandwidth consumption of each and every node in all the test cases. Until and unless the CA selects the cluster CH, the bandwidth values of the nodes will be popping out for every cycle they keep completing (Cycle here being movement from one end of the monitor screen to the other, horizontally).

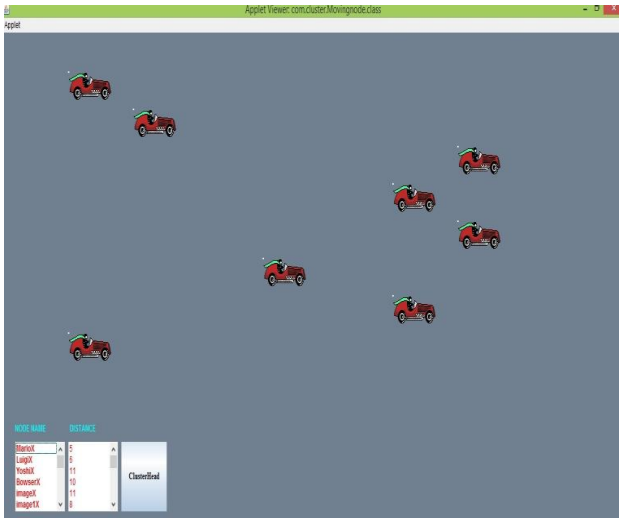

Figure 12.1. Moving nodes in the network.

We're also providing other test cases for reference. Figure 13 represents the CH selections from the IDE console. We represent CH selections in the rectangular boxes.

This selected CH will be responsible for the path quality management in the cluster, where all the CHs along with CA, in turn will be responsible for the path quality in the entire network. As we can see the maximum threshold value is always lie with the CH, if any of the nodes in the cluster possess greater value than that of the cluster head, it will be immediately notified as a faulty node to the CA and none of the data will be transmitted to or via the faulty node. As CH is the most trust worthy node in the entire cluster, it will establish an optimal routing path among the other clusters for data transmission. So

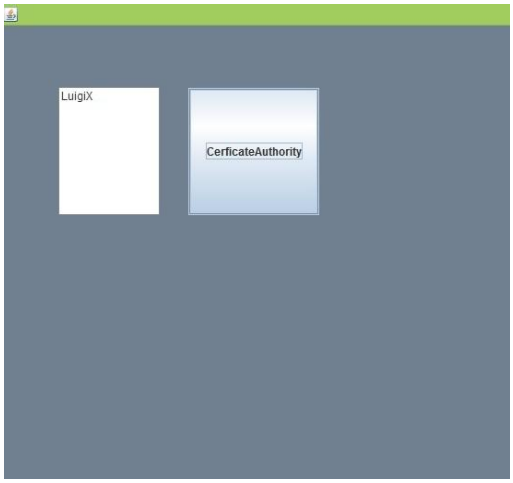

Figure 12.2(i). Cluster head selection depicted in UI.

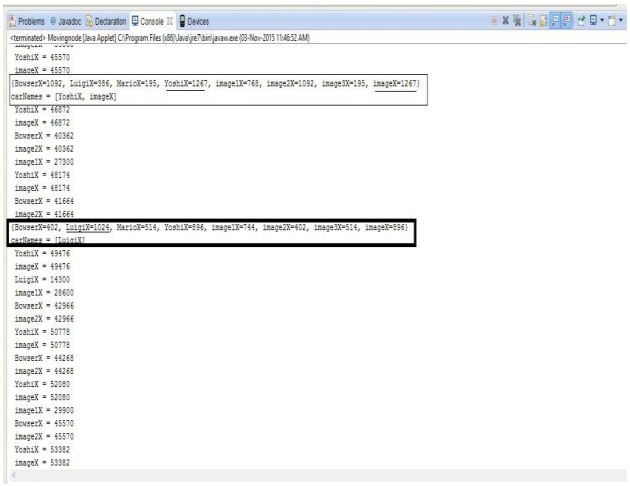

Figure 12.2(ii). Cluster head selection depicted in the IDE Console with each and every node values.

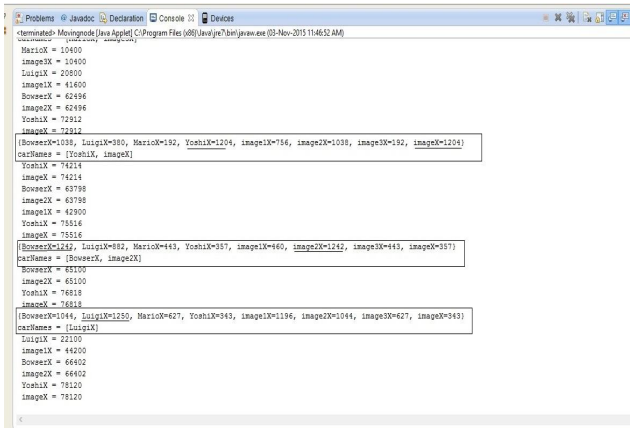

Figure 13. Cluster Head Selections for various test cases in IDE console.

we are considering both the transmission rate as well as the data yield of each and every node in the network, thus enabling the analysis of forwarding quality of each and every node in the network.

5. Conclusion

Firstly, we've identified that depending solely on the existing link based metrics like ETX for path quality and routing determination may not be fruitful in every case. Next, we proposed a scheme where CH determination by the CA, will look after, in fact achieve the required path quality management by taking into account the capabilities of both the path link and the sensor node. Finally, we've evaluated this proposed scheme and proved that usage of this CH methodology will achieve the required forwarding path quality management in WSNs.

6. Acknowledgements

We thank Koneru Lakshmaih University (K L U) for their constant support throughout this research. We are also indebted to "Embedded Systems and Sensor Network (ESSN)" research group of K L U for facilitating this research work.

7. References

1. Singh SK, Singh MP, Singh DK. Routing Protocols in Wireless Sensor Networks –A Survey. International Journal of Computer Science and Engineering Survey (IJCSES). 2010 Nov; 1(2):63–83.
2. Javaid N. Javaid A, Khan IA, Djouani K. Performance study of ETX based wireless routing metrics. IEEE, 2nd Intl Conference on Computer, Control and Communication, IC4 2009, 2009 Feb17-18. p. 1–7.
3. Sang L, Arora A, Zhang H. On exploiting asymmetric wireless links via one-way estimation. Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing ACM MobiHoc. 2007. p. 11–21.
4. Mo L, et al. Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest. Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys). 2009. p. 99–112.
5. Werner-Allen G, et al. Fidelity and Yield in a Volcano Monitoring Sensor Network. Proceedings of the 7th symposium on Operating systems design and implementation (USENIX OSDI). 2006; 381–96.

6. Srinivasan K, et al. Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks. Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys). 2006. p. 419–20.
7. Zhao J, Govindan R. Understanding packet delivery performance in dense wireless sensor networks. Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys). 2003. p. 1–13.
8. Dong W, Liu Y, He Y, Zu T. Measurement and Analysis on the Packet Delivery Performance in a Large Scale Sensor Network. IEEE/ACM Transactions on Networking (TON). 2014 Dec; 22(6):1952–63.
9. Cirstea C, Cernaianu M, Gontean A. Packet loss analysis in wireless sensor networks routing protocols. IEEE, Telecommunications and Signal Processing (TSP), 2012 35th International Conference at Prague. 2012. p. 37–41.
10. Couto D, Aguayo D, Bicket J. A high-throughput path metric for multi-hop wireless routing. ACM, Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, MobiCom. 2003. p. 134–46.
11. Xu R, Wunsch D II. Survey of clustering algorithms, Neural Networks. IEEE Transactions on Neural Networks. 2005 May; 16(3):645–78.
12. Liu X. A Survey on Clustering Routing Protocols in Wireless Sensor Networks. MDPI-SENSORS. 2012 Aug; 11113–53.
13. Davies DL, Bouldin DW. A Cluster Separation Measure. Pattern Analysis and Machine Intelligence, IEEE. 2009; PAMI-1(2):224–7.
14. Karthickraja NP, Sumathy V. A study of routing protocols and a hybrid routing protocol based on Rapid Spanning Tree and Cluster Head Routing in Wireless Sensor Networks. IEEE, Wireless Communication and Sensor Computing, ICWCSC 2010. 2010; 1–6.