Int. J. Advanced Networking and Applications
Volume: 16 Issue: 03 Pages: 6428-6436 (2024) ISSN: 0975-0290

An Energy Efficient Tree-Based Routing Protocol for Wireless Multimedia Sensor Networks

Sharmin Rashid
Department of Computer Science & Engineering, Primeasia University, Dhaka, Bangladesh
Email: sharmin.rashid@primeasia.edu.bd
Md. Ridgewan Khan
Department of Computer Science & Engineering, Patuakhali Science & Technology University, Bangladesh
Email: khan.ridgewan@gmail.com

ABSTRACT

The recent technological advances in micro electromechanical systems have promoted the development of a powerful class of sensor-based distributed intelligent systems capable of retrieving multimedia information, namely Wireless Multimedia Sensor Networks (WMSNs). WMSNs are gaining more popularity day by day as they are envisioned to support a large number of both non-real time and real-time multimedia applications. Routing in WMSN is very challenging task. Multipath establishes several path for data transmission rather than single path. In this paper we proposed An Energy Efficient Tree Based Multipath Routing for WMSN's (EETMRP) technique that will intend to provide a reliable transmission of data for data synchronization at the destination node with low energy consumption. This is done by efficiently utilizing the energy availability and the received signal strength of the nodes to identify multiple routes to the destination. EETMRP discusses that all of the packets of audio, image and scalar data are spread over the nodes lying on different possible paths between the source and the sink, in proportion to their residual energy and received signal strength. The objective of EETMRP is to assign more loads to highest energetic paths and fewer loads to lower energetic paths so that uniform resource utilization of all available paths can be ensured. It uses multiple paths between source and the sink which is intended to provide a reliable transmission environment with low energy consumption, by efficiently utilizing the energy availability of the nodes to identify multiple routes to the destination and efficiently balanced the loads of all sensor nodes. We also focused on node failure and it recovery process during packet transmission and during the start of packet transmission. Finally we proposed an algorithm based on our proposal to simulate and analysis our proposed routing protocol.

Keywords - **Wireless Multimedia Sensor Networks (WMSNs), Routing protocols, Multipath Routing, Energy Efficient Routing, Error Recovery, Quality-of-Service (QoS).**

Date of Submission: October 09, 2024 Date of Acceptance: October 28, 2024

I. INTRODUCTION

The dense deployment of small, low-powered, self-organizing sensor nodes in Wireless Sensor Network (WSN) can sense and process the scalar data such as temperature, humidity, pressure etc. of the physical environment. In the recent years due to the advancement in the micro-electronics and wireless communications, in addition to availability of cheap CMOS cameras and micro phones, have allowed the emergence of a new promising Wireless Multimedia Sensor Networks (WMSNs). WMSN is comprised of small embedded video motes capable of extracting the surrounding environmental information, locally processing it and then wirelessly transmitting it to parent node or sink.[1] It is comprised of video sensor, digital signal processing unit and digital radio interface. Most of the researchers envisioned that performance of the existing applications of WSN drastically can be enhanced with the aid of multimedia sensors which can gather audio, video, image information etc. along with scalar data. Now a days routing in WMSN is a challenging task. Based on this our paper is on an Energy Efficient Tree Based Multipath Routing Protocol for WMSN where we emphasis on multipath routing techniques of WMSN.

II. WIRELESS MULTIMEDIA SENSOR NETWORK

Wireless Multimedia Sensor Networks (WMSNs) is comprised of small embedded video motes capable of extracting the surrounding environmental information, locally processing it and then wirelessly transmitting it to parent node or sink. It is comprised of video sensor, digital signal processing unit and digital radio interface. In wireless communication networks wireless sensor networks (WSNs) have gained significant importance in the last few years. Currently WSNs are targeting a number of application scenarios ranging from civil and military applications to modern healthcare. WSN are basically comprised of scalar sensors capable of measuring the physical phenomenon like temperature, pressure, light intensity, humidity etc. Today WSNs are used on large scale capable of gathering information from the physical environment, processing it and transmitting the processed information to remote server or location [2]. In either of the above listed applications the

Volume: 16 Issue: 03 Pages: 6428-6436 (2024) ISSN: 0975-0290

bandwidth requirements is not stringent and is delay tolerant.

Now the availability of complementary metal-oxide semiconductor (CMOS) camera and small microphones make possible the development of WMSNs capable of gathering the multimedia information from the surrounding environment. The advent of WMSNs has opened a new vision to existing WSNs thereby enhancing its existing capability (due to incorporation of vision sensor and high computational engine) and targeted applications like Intelligent Transportation system (ITS), multimedia surveillance sensor networks etc. Previous research targets the challenges posed by WSNs like limited node computational and communication power, power source, scalability etc. Now with the development of WMSNs additional challenges are added which must also need to be addressed i.e. application specific QoS constraints, coverage area, in network processing, high bandwidth demand, heterogeneous multimedia reliability etc.

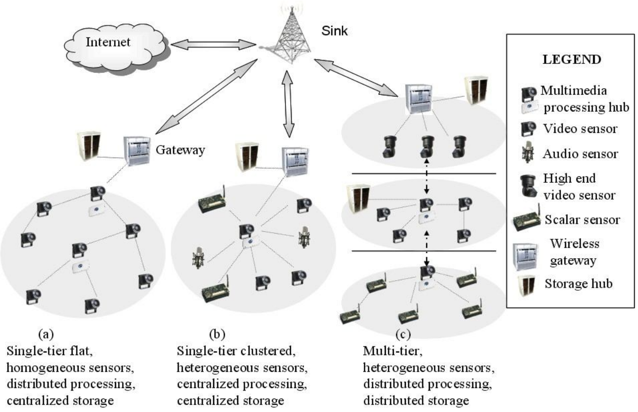


Figure 2.1: WMSN Architecture

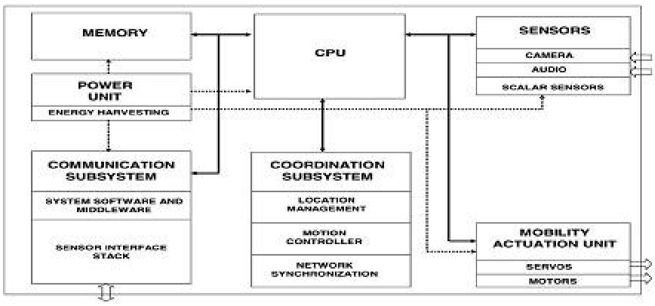


Figure 2.2: Internal organization of a multimedia sensor

III. LITERATURE REVIEW

The important issue to be considered from the communications point of view is routing. Routing in WMSNs is a demanding task. This demand has led to a number of routing protocols which efficiently utilize the limited resources available at the sensor nodes in addition to the features of multimedia data. The numerous challenges and issues that the developer must face while designing routing protocols for the transmission of multimedia data due to its resource- intensive nature compared to the traditional data are discussed in [3,4,6,7].There are many goals of Multipath routing protocols to achieve such as Reliability, Load Balancing, High Aggregate Bandwidth, Minimum End to End Delay, Minimum Energy consumption, High throughput.[5]In this section, we focus on multipath routing protocols for WMSNs that include routing and scheduling functionalities.

Protocol	Network Architecture	Geographic Routing	Operational Layer	Fault-Tolerant Mechanism	Performance Metric
Multi path and Multi-speed (MMSPEED)	Flat	Yes	Routing/MAC	Weak	Delay/Overhead
Two Phase Geographical Greedy Forwarding (TPGF)	Flat	Yes	Routing	Medium	Delay/Hop Count
Multi- Priority Multipath Selection (MPMPS)	Flat	Yes	Routing	Medium	Distance/Delay
AntSensNet	Hierarchical	No	Transport/Routing	Medium	Packet Delivery Ratio/Delay/ Packet Overhead
Power Efficient Multimedia Routing (PEMuR)	Hierarchical	No	Routing/MAC	Medium	Delay/Energy Consumption

Table 3.1: Existing Routing Protocols of WMSN

Protocol	Network Architecture	Location awareness	Multipath	Congestion control	Energy Efficiency	QoS parameters considered
SAR[27]	Flat		✓		✓	Delay/Overhead
Multipath Video Delivery[44]	Flat	✓	✓		✓	Delay/Hop Count
Yao et al.[28]	Flat		✓		✓	Delay
Li et al.[30]	Flat		✓	✓		Delay
LEAR [31]	Flat	✓	✓		✓	Reliability
Poojary and Pai [32]	Flat				✓	Reliability
GEAMS [33]	Hierarchical	✓			✓	Delay-Reliability
N to 1 multipath routing[35]	Flat		✓	✓	✓	Reliability
REER-1 [37]	Flat			✓		Delay

Table 3.2: Comparisons of Existing Routing Protocols of WMSN

By analyzing the above literature we observed that some protocols use data aggregation technique that enhances the efficiency of network by reducing the number of transmitting data, Some routing protocols do not consider node residual energy during construction phase, some protocol maintain additional paths to serve as backup on primary path failure not the failure during transmission of packets. Maximum WMSN protocols discuss on transferring whole video data sending process at a time on different paths. So that the loss of packets will create problem to receive actual video data. Some protocols suffer from the overhead of maintaining routing tables and QoS metrics at each sensor node. So there is an urgent need to develop routing protocols that are more energy efficient, are more reliable and have better control regarding the QoS requirements of multimedia data. In our proposed protocol we try to utilize node resources through Load balancing and to make it more energy saving and increases delivery ratio.

IV. PROPOSED ROUTING PROTOCOL OF WMSN

Routing in wireless multimedia sensor networks is a demanding and very important task. This demand has led to a number of routing protocols which efficiently utilize the limited resources available at the sensor nodes. All these protocols are created to find the minimum energy path. In this paper we propose an energy efficient tree-based multipath routing protocol of wireless multimedia sensor networks which describes that, if we send the multimedia data such as video data in a path then the energy of sensor nodes decrease very rapidly and the load of the path and the time taken to determine an alternate path increases. So that we propose a way to split the video data into audio, image and scalar data and send them through corresponding highest energetic and low energetic path to reduce the overload of the sensor nodes in the network. In this case we follow the multipath routing schemes. Multipath routing schemes distribute traffic among multiple paths instead of routing all the traffic along a single path. In Multipath routing there are two key questions: how many paths are needed and how to select these paths. Clearly, the number and the quality of the paths selected dictate the performance of a Multipath routing scheme. Our protocol is intended to provide a reliable transmission environment with low energy consumption, by efficiently utilizing the energy availability and the received signal strength of the nodes to identify multiple routes to the destination. In our proposed routing protocol all of the packets of audio, image and scalar data are spread over the nodes lying on different possible paths between the source and the sink, in proportion to their residual energy and received signal strength. The rationale behind traffic spreading is that for a given total energy consumption in the network, at each moment, every node should have spent the same amount of energy. The objective is to assign more loads to highest energetic paths and fewer loads to lower energetic paths so that uniform resource utilization of all available paths can be ensured. Multipath routing is cost effective for heavy load scenario, while a single path routing scheme with a lower complexity may otherwise be more desirable. We consider the following two major issues in our proposal.

4.1 Assumptions of the Network Environment

The sensor nodes are distributed randomly in the sensing field. After pushing them into the network, their position will be static during the entire network lifetime. We consider a small area where all of the sensor nodes are placed and the network is composed of source node, intermediate nodes and sink node. The system environment is shown in figure 4.1.

Assume that all nodes in the network are assigned with a unique ID starting from 1 and all nodes are participating in the network and forward the given data. The sensor nodes are assumed to be static for their entire network lifetimes, and the energy level of each sensor node is determined a randomly. Additionally, these sensor nodes have limited processing power, storage and energy, while the source and sink nodes have powerful resources to perform any tasks or communicate with the sensor nodes. Each node is placed in a random transmission range. Once the nodes are deployed, they start their sensing tasks. The sensor nodes can send and receive messages from other nodes. Before selecting the multipath a node is selected as neighboring node to forward the data based on its available energy level signal strength and transmission range. The greater the energy in the node and farther the node from the previous one, is the more likely to be selected as the next hop. The nodes which are not selected in this process are moved to the sleep mode in order to conserve power. The communication is assumed to be bidirectional.

The protocol replies with multiple routes from the source node to the sink quickly according to determine the transmission range, status of each node and the energy level of the nodes, and prepares the paths that efficiently balance the energy of the nodes. It also enables the selected nodes in the path to aggregate all the received packets during a short period of time and to transmit only the aggregated packet to the next node. Each node maintains a neighbor table in which it reserves the information of the own and neighboring nodes' energy level, available buffer size of them and their status (sleep=0 or alive=1). The neighboring table is always updated when the transmission of the packets will start.

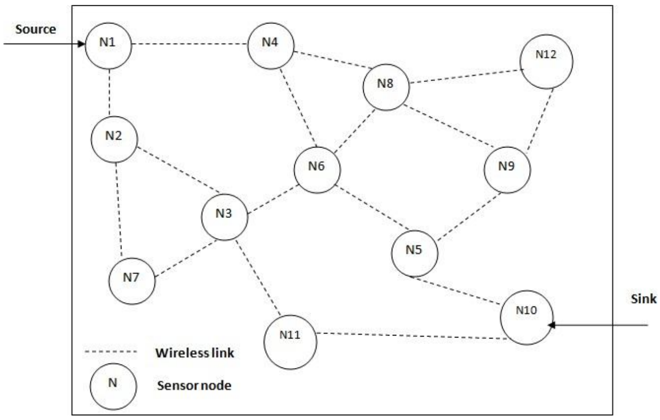


Figure 4.1: System Environment

4.2 Multipath Routing Technique

Multiple routes in WMSNs are necessary to satisfy the desired data rate at the destination node. Benefits of selecting multiple paths among a WMSN: Reduced correlation among packet losses, Increase channel that can support the application's demands in QoS, the power consumption is more evenly spread in the network nodes extending longer life time, ability to adjust to arbitrary congestion occurrences in different parts of the network. A sensor node is selected to forward the data, based on its available energy level and signal strength. Ideally, the greater the energy in the node and farther the node from the previous one, is the more likely to be selected as the next hop. The nodes which are not selected in this process are moved to the sleep mode in order to conserve power. The communication is assumed to be bidirectional and symmetric. The main components of multipath routing are: path discovery, neighbor status table and the information routing algorithm.

4.3 Path Discovery

In path discovery phase sink spreads the route request message to all of the nodes to create neighboring nodes status table that is based on all the information of all the

Volume: 16 Issue: 03 Pages: 6428-6436 (2024) ISSN: 0975-0290

nodes such as node id, distance range, residual energy, free buffer etc. These neighboring nodes are used to forward data towards the sink from the source node. During this process route request message is exchanged between the sensor nodes. The constructed multipath have no common nodes except the source and the destination. Node-disjoint paths are fault-tolerant so there is a minimum impact to the diversity of the routes. Route discovery includes several phases explained as:

Create Neighboring Table: Each sensor nodes in the network maintain a routing table where the table contains all the information of the current nodes and the neighboring nodes. The structure of the routing table of each sensor node which includes the information such as transmission range, residual energy, buffer size and status (sleep/alive) of nodes. The transmission range gives the information of the distance from the current node to all other nodes, residual energy provides the energy level of the nodes, buffer size provides the available packet storage size of each node and the status provides the node is either alive or in sleep mode. If the node is alive then the value of status field will be 1 either 0.

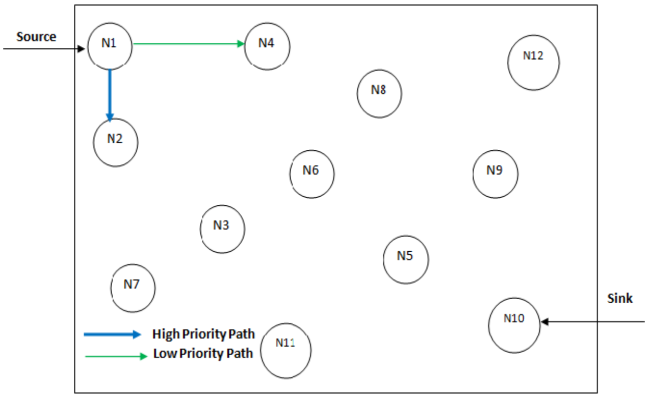


Figure 4.2: Initial path creation (step 1)

In step 1 we see that the source node N1 select the nodes N2 and N4 as its neighboring nodes because they are within the transmission range from the N1. Assume that both of the nodes are in alive mode and the energy level of N2 is 25 units and N4 is 20 Units. Then N1 first selects the N2 node as its neighbor and then select N4. So the path from N1 to N2 is High priority path and from N1 to N4 is next priority path.

In step 2, by repeating the above steps N2 selects N7 and N3 path as its neighbor. N6 is also in its transmission range but N2 does not choose N6 because it is in the sleep mode. The nodes which are in sleep mode conserves energy. According to energy level it chooses N7 as Highest priority path and N6 next level priority path. The above procedure is repeated for all nodes to create the multiple paths until they reach to the sink node.

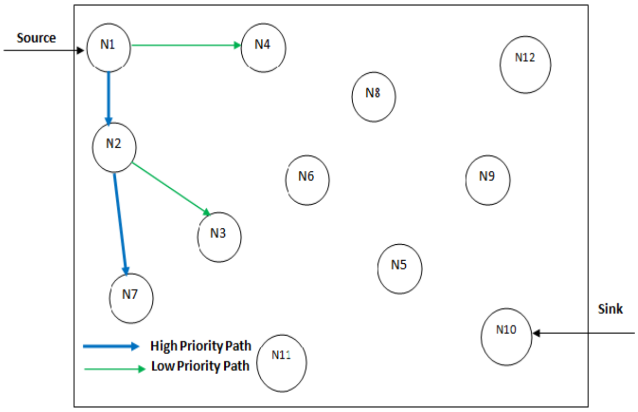


Figure 4.3: Initial path creation (step 2)

In step 3, finally we get the following multipath scenario of the entire network:

Routes are established between the source and destination only when required hence it reduces the sensor node overhead. The multipath routing protocol computes the energy expense is less to transmit, receives and stores it in the routing table. Routing in general differ significantly from the specialized service requirements of multimedia streaming applications. The protocol replies with multiple routes from the source node to the sink quickly, and prepares the paths that efficiently balance the energy of the nodes. It also enables the selected nodes in the path to aggregate all the received packets during a short period of time and to transmit only the aggregated packet to the upstream node. Now, from figure 4 finally we see that node [N1] has a valid route to node [N10] via: N1-N2-N7-N11-N10 and can start to transmit its data packets. Each selected path information will be stored in the database of the sink node.

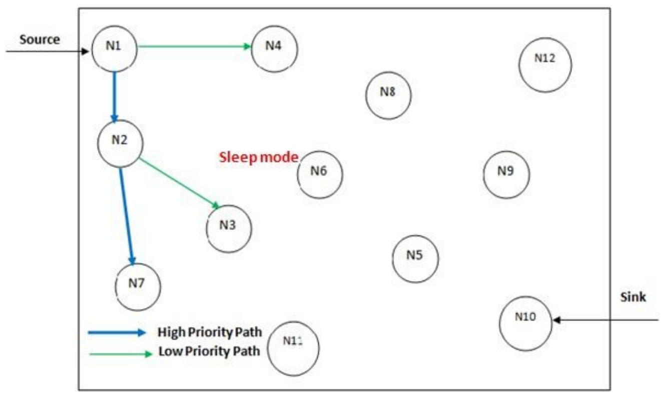


Figure 4.4: Initial path creation (step 3)

Data Sending Process

Once the multiple paths are discovered, source node transmits data packets with data rate initialized. Since the data is multimedia it can be video data. So to recover the sensor nodes from overload in the entire network lifetime the video data is separated into audio, image and scalar data. We provide some metric on these data such as-

Scalar Data: Metric: 0

Image Data: Metric: 1

Audio Data: Metric: 2

Volume: 16 Issue: 03 Pages: 6428-6436 (2024) ISSN: 0975-0290

Data packet comprises of metric number, checksum, sequence number, source id, and information length. If the number of packets is less than the number available paths from source then packets are sent to each path individually according to the path's energy. If the number of packets is greater than the number of available paths then send one packet through each path and remaining packets through highest energy path. Audio data will send in high energetic path, image data into midlevel energetic path and scalar data into low energetic path. After one transmission the battery life of each sensor nodes is reduced by 5 units. The data sending procedure is given below:

Send data: Node [N1] sends data to node [N10]

Whenever a node receives a data packet, it acknowledges the reception by sending an ACK to the source (previous) node. Packet is updated for every hop, which helps the sink node

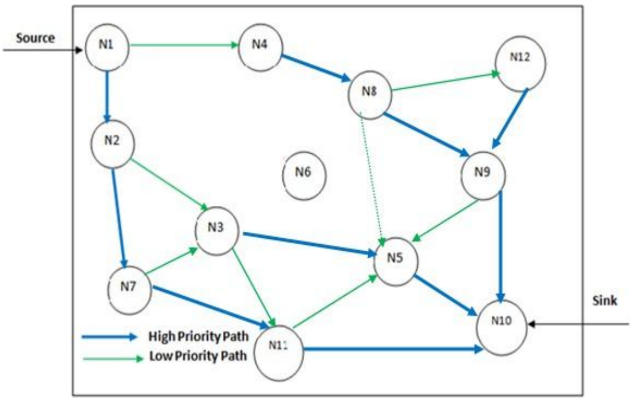
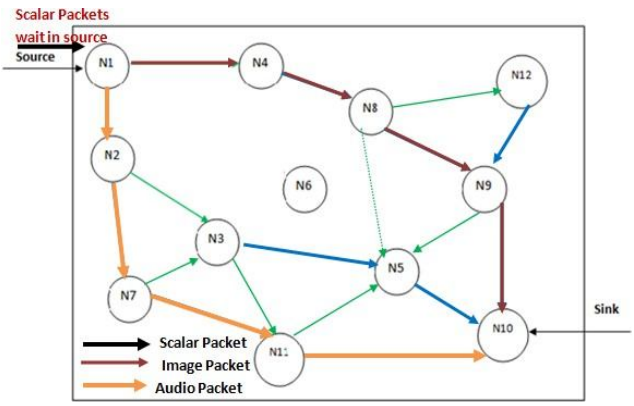


Figure 4.5: Final Paths

to monitor the conditions of the multiple routes being used. The initial data rate used for the routes may not be optimal for the duration of the connection. Sink node redistributes the data rates over multiple paths to optimize the usage of network resources periodically. If suppose one node is used in one route then it should not be used in other route to avoid collision of data while transmission. To achieve this we use a status factor which indicates the status of each node.



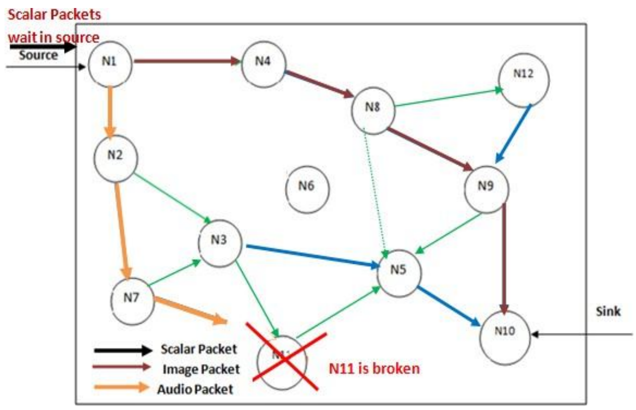
4.6: Source sends data

Multipath Maintenance (Route Error and alternative path searching):

We assume that at least one of the nodes over this discovered path has lost because of natural causes or any other causes, during an active session. Now, we have a broken link between these two nodes. In such case, the previous node will find the alternative path to send the packet to the sink or will send back a Route Reset (RRES) packet to the source node. In order to detect a path failure, the sink also monitors the inter-arrival delay of data packets on each path. When the delay is above a pre-determined threshold, the sink presumes that the path is broken. Then the sink will send a reset message with the path information that is broken to the source through the optimal path to indicate that sink wants to start to re-initiate the paths search phase. Source then sends the packet through alternative path. Alternative paths can be chosen for two causes which are given below:

Cause 1:

From figure 4.7, assume somehow the node [N11] is lost. So, if node [N7] forwards a data packet to node [N11], it does not receive any ACK message. After some tries, it considers this link as a broken link and node [N11] no more reachable through this path. Then N7 node will check



its own routing table and find the next energetic path from it.

Figure 4.7: N7 to N11 link broken

Assume it finds node N3. If node N3 is free from other packet sending then N7 chooses the N3 node as its neighbor node instantly and send the packet to N3. N3 has already two neighbor nodes N5 and N11 where N11 has already been broken. If N5 is free then applying above procedure the packet is sent until reach to the sink node. The scenario is shown in figure 4.8. If N5 is remaining busy to send other packets through another path then N3 stores the packet for sometimes and check the N5 until it will be free. If N5 is free and has enough energy below threshold then N3 sends packet to it. If node N3 finds that the energy of Node N5 is finished after sending other packets and N3 has no alternative paths then N3 sends the Route Reset message to the source where the information of the packet also be held. The source then re-initiates the new path by repeating the path discovery phase and sends the packet again. Here is an important issue, if the sender node finds out that the next node is damaged then it will hold the packet in its queue. When it will receive another packet from the previous node then it will send an acknowledgement message where there will be indication that some problems has occurred so the previous nodes of that path should stop sending the DATA packets. For sending this message an extra field should be added in the header which named QUEUE Field. The field can contain only 0 or 1. If it contains 1 then the previous node will understand some problems has occurred and it will stop sending packet immediately and forward the message to its previous node. This process will go on until the source receives this message and stops sending DATA packet and source will search for another alternative path. When the rerouting or redesigning is done then the previous node of the failed node will send an acknowledgement message making the QUEUE field 0 and DATA transmission will start again. Now the packet is sending through N1-N2-N7-N3-N5-N10 path.

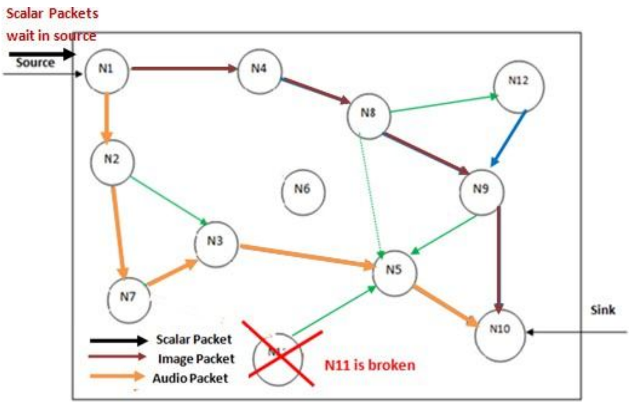


Figure 4.8: Create alternative paths N1-N2-N7-N3-N5-N10

In figure 4.9, node [N3] forwards the RRES message to node [N1] with the packet information. Then, node [N1] updates its route table, and if the session is still active and if there is another valid route to destination in its route table, node [N1] uses this route, otherwise it starts a new route discovery process. Whenever a node receives an RRES packet, it deletes all route entries in its route cache table, which includes this broken link.

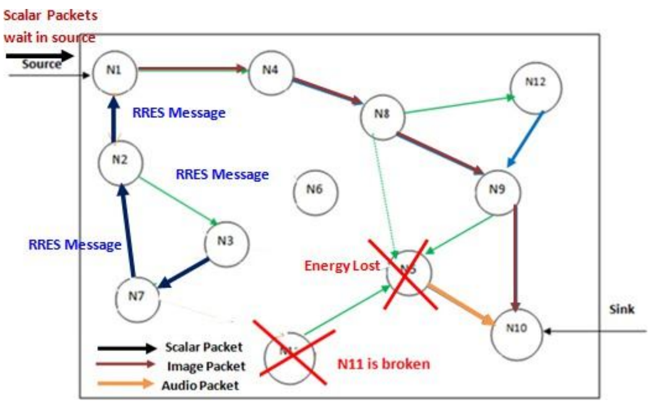


Figure 4.9: Send RRES message to source node

Cause 2:

At the initial stage of packet sending, if the source node finds that its neighboring node of the selected path is broken then it instantly delete the selected path and starts new path discovery process if no any alternative path remains. The scenario is given in figure 4.10:

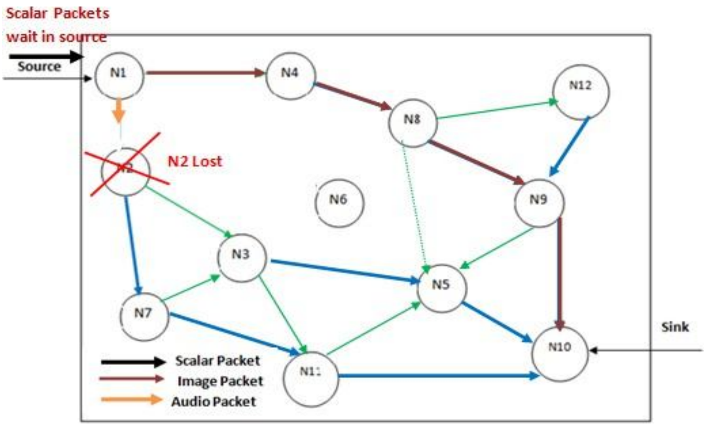


Figure 4.10: Initially the neighbor node of selected path of source lost

In the figure 4.10, we see that the highest priority path is selected that is N1-N2-N7-N11-N10. When source node N1 takes steps to send the packet it find that N2 is somehow broken. So it then deletes the N1-N2-N7-N11-N10 path from its routing table and tries to send the packet from next selected path that is N1-N4-N8-N9-N10 if N4 is not busy either it starts to discovery new path. After reaching all the packets to the sink node with a Time to Live (TTL) period the extract the actual data from the coded data and arrange the data packets according to their sequence number. Store and display the actual data.

V. PROPOSED ALGORITHM

Since our proposed routing protocol is Multipath Routing protocol. So we can implement this protocol by following our proposed Multipath routing algorithm. This algorithm is followed by detailed functions used in the algorithm are as follows:

- **Start**

To implement our proposed routing protocol we can initially use the following functions used in different section of code and which has different purposes such as:

- CreateRoutingTable():

The purpose of this function is to create Neighbour table/ Routing table of each sensor node.

- CreateMultipaths():

This function is used to create possible multiple paths from source node to sink node.

- PacketInformation():

The purpose of this function is to send the packet data at each sensor node.

- PacketSend():

This function discusses about the processing of choosing alternative paths if any node during transmission will fail.

- ExtractReconstructPacket():

Volume: 16 Issue: 03 Pages: 6428-6436 (2024) ISSN: 0975-0290

This function is used to extract and reconstruct all the packets at the sink node.

- End

CreateRoutingTable():

Variables:

Nodes= Total number of nodes in the network
pos[ ].x pos[ ].y= x and y are the co-ordinates of current node position nodeId= Unique id of each node
ResEnergy= Energy of each node (Randomly generated)
ThresholdEnergy= 5 units
Distance= distance from current node to all other nodes.
MinDistance=5 sq.meters. $H_count = 0$

1. Start.
2. Input total number number of nodes.
3. Input the position of each node (x and y coordinates)
4. Calculate the distance from one node to all other nodes

$$distance = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

where x1 and y1 are position of current node, and x2 and y2 are position of the node from which the distance is to be calculated.
For current node to all other nodes- if(*distance* < *MinDistance*) then
if(*pos.ResEnergy* > *ThresholdEnergy*) then Update the Routing table of current node.
Repeat the step 5 for all other nodes. End
At first we get the number of sensor nodes and position of each node from the user.
Each sensor nodes must have unique id number. Initialize the first node with a value 0, find distance between each of neighboring node by the distance formula. If the distance is within the Minimal Distance and node's residual energy is greater than the threshold energy then update the neighbors of current node in neighbor table.

CreateMultipaths():

Variables:

pos[ ].neighbour[ ]= array that store the selected neighbour nodes of the current node pos[ ].route[20]=structure to store all selected possible paths from current node. src=source node
sink= destination node
pos[ ].ResEnergy[ ]= structure to store the energy level of neighbour nodes
pos[ ].status[ ]=structure that stores status of each node (0= sleep mode and 1=Alive mode)

1. Start.
2. Input source and destination node id.
3. Get all information about each node and its neighbors from neighbor table pos[i].neighbour[j]

== 1 where j is neighbor of i
if (pos[i].status[j]==1) [where 1 means node is alive]
if destination is neighbor of source then display as direct path from source to destination
else
From the source node start finding the route to destination, first consider the neighbors of source node, check their status and then select the node with highest energy.(max(pos[i].ResEnergy[j])) then

4. Repeat the step 3 until creating all possible multipaths from current nodes.
5. The selected node will act as source node and repeat the step 3 and create multi- paths until reaching to the destination.
6. Store all the selected paths in route[ ] array in all nodes.
7. End

Get the source and destination node ids from the user and collects all information about each node and its neighbors from neighbor table. If destination is neighbor of source then display as direct path from source to destination. From the source node start searching the route to destination, first consider the neighbors of source node, check their status and then select the node with highest energy. Next make that node as source and continue the same process till the destination is reached. After finding a path from source node to the destination node, the source code then choose next high energetic path and continue the same procedure to find the different possible paths from source to destination and store the each path in a route array.

PacketInformation():

Variables:

TotalData[ ]= array that store total nomber of data packeets
pos[src].nodevalue=data generated at the source
src=source node
data=variable name to store intermediate data. metric= metric value of splitted video data(scalar data=0, image data=1, audio data=2)

1. Start.
2. User input data in the source node.
   data=pos[src].nodevalue
3. Unpack the splitted multimedia data
4. Convert the unpacked data into binary.
5. Assign sequence number,metric value, and calculate the checksum for each packe
6. Arrange data in the data format shown below:

Scalar Packet size: 8-bit
Sequence no: 2 bits Data: 4 bits
checksum: 2 bit
metric: 0

Image Packet size: 16-bit
Sequence no: 2 bits Data: 12 bits
checksum: 2 bit
metric: 1

Audio Packet size: 32-bit
Sequence no: 2 bits Data: 28 bits
checksum: 2 bit
metric: 2

7. Stores the data packets into the TotalData[ ] array.
8. Ready for transmission of packets.
9. End

PacketSend():

Variables:

SelectedPath[ ]= array to store selected path of current node
pos[ ].ResEnergy[ ]= The residual energy of nodes
ThresholdEnergy=5 units TTL= 150 sec

1. Start.

Volume: 16 Issue: 03 Pages: 6428-6436 (2024) ISSN: 0975-0290

2. Get the packets
3. (Check Path Failure from source node)

if any path is failed due to insufficient energy(Insufficient energy less than thresh- old) then delete the corresponding path and choose another path from next queue of source node.(choose next highest energetic path)

4. If (no of paths= the number of data packets) Then send one packet through each path

else

If (no of paths ¡ the number of data packets) Then send audio, image data through corresponding highest energetic, Mid energetic path and remaining scalar data through low energetic path and store the selected path into SelectPath[ ] array.

5. (During Transmission) If any node fails during transmission then the previous node will finds the next energetic selected paths from its routing table and choose that path which is not used in that time either after Time to Live period(TTL) source again delete the path and by repeating step 3 it selects the alternative path, update its routing table and sends the packet until it reach to the sink node.*[The algorithm calls from CreateRoutingTable() function, CreateMultipaths(), PacketInformation() and PacketSend() functions again.]*
6. After one transmission reduce the battery life of of sensor nodes by 5 units pos[i].ResEnergy=pos[i].ResEnergy-5
7. End

Get the intermediate data and check for path failure, if any path is failed due to insuf- ficient energy of intermediate node then delete the corresponding path and choose next high energetic path of the previour node. If no any path remaining to use then after a TTL perion source identify that destination does not send a ACK message of the data of the lost path. So source instantly delete that path and choose alternative path be applying all CreateRoutingTable() function, CreateMultipaths(), PacketInformation() and PacketSend() functions. If number of paths are greater than the number of data packets, then send one packet through each path. If number of paths is less than the number of data packets, then send one packet through each path and remaining packets through highest energy path according to priority metric value. After one transmission reduce the battery life by 5 units.

ExtractReconstructPacket():

Variables:

SeqNo= sequence numbers of data packets used for packet synchronization dest= Destination node Id

pos[dest].nodevalue=data at the destination node data= variable name that store final data FinalData[ ]=array that stores the decoded data

1. Start.
2. Get the packets to the destination node which arrive and store them in data = pos[dest].nodevalue
3. Extract the actual data from the coded data and arrange all packets into their sequence numbers (SeqNo).
4. Store the actual decoded data into FinalData[].
5. End

Collect the data from destination node arrived through different paths. After collecting data from the destination node, extract the actual data from the coded data and arrange the data packets according to their sequence number. Finally store and display the actual data.

VI. DISCUSSION

After proposing the algorithm of our proposed protocol (EETMRP) we have tried hardly to simulate our proposal. For this we tried to implement it in Network Simulator2 (NS2) because now a day the NS2 simulator plays an important role to simulate the network topology, routing protocols. Because of short time we can’t implement our proposed protocol. The development of our proposed algorithm is based on some assumptions about performance analysis of our protocol after implementation which is discussed below:

We developed our proposed protocol and algorithm by comparing with existing pro- posed *Energy Efficient Multipath Routing Protocol for WMSNs (EERP)* [9]. Where the authors of [9] proposed that the multimedia data (video data) will be sent through multiple paths. Since video data sending process needs enough battery energy capacity of sensor nodes so that medium/ low energetic paths cannot be used to send video data through that paths. If packets will send through those paths then the possibility of node lost will high. In our proposal we proposed that the video data will split into image, scalar and audio data, we set priority to all of them and send them mid energetic, low energetic and high energetic paths to balance the loads properly to all the nodes in the network.

They also not discussed about the intermediate node failure during the transmission. They just proposed if any node is failed just before the transmission of packets from source node then that path will be deleted from source’s routing table and if any inter- mediate node will fail then after a TTL period sink send a message to the source node that the particular packet does not reach because of path failure. So this process is also time consuming. Based on this we proposed that if any intermediate node will fail during transmission the node where the packet is placed last will check its own routing table to find alternative path from it to the sink. It will help to resolve the time consuming scenario. Because of facing those limitations of the paper [9] we developed our protocol and algorithm to compare with their algorithm. We compare our protocol with the above protocol successfully but since we can’t implement our algorithm so that we can’t analyze the performance of our algorithm pictorially.

So if we simulated EETMRP we wish our proposed scheme will simulate in various network scenarios to access the performance and effective of an approach. We will provide the following data values after executing our algorithm: We assume that, The main experiment parameters are Number of Nodes, Topology size, Transmission range, Energy of all nodes, Sending power, receiving power, packet size. If we implemented our algorithm then we would consider the following parameters and values in our program:

Network field	200m X 200m
Number of Sensor	30-100
Number of Sinks/Number Sources	1/1
Minimum Transmission Range	5m
MAC layer	IEEE 802.11
Energy Threshold	5 Units
Max Buffer Size	256 Kilobytes
Simulation Time	150 sec

Table 5.2: Simulation Parameters

Our protocol would execute using the following steps:

- Generate the WMSN for the given area and number nodes randomly.
- Apply the proposed multipath routing protocol.
- Compute the performance of the system.

We developed our algorithm in the sense of assumptions of the following Performance Parameters comparisons:

Node energy consumption: The node energy consumption measures the average energy reduced by a node in order to transmit the data packet from source to sink. We would compare between Node energy consumption and Number of packets sent. We assumed that EETMRP would consume less energy as the no. of packet increases in the network because of load balancing approach.

Network Life Time: It is total time during which the network is alive. As the sensor node energy depletes life time of the network is reduced. EETMRP would compare between Data transmission time Versus the Number of paths with varying data size. We assumed that if the numbers of paths are increased in the network then the time taken to delivery packets from source to sink will also increase.

VII. CONCLUSION

In our thesis we have focused on some very challenging issues on routing protocol in WMSNs. In our proposal we have tried to solve the problem of energy efficiency and transmission even after node failure. For solving the problem of energy efficiency we proposed a multipath routing protocol so that we can utilize the energy of all the nodes properly. We have prioritized DATA and sent them according to the priority. We have considered node failure for on-going transmission. After node failure new path will be designed using the broken path and the working paths. Thus the on-going transmission will not be hampered and it will make sure that DATA can be delivered even after node failure or any such kind of incident. Though we haven’t done the simulation yet, from the research we can tell that this protocol will increase the DATA transmission rate and decrease the failure rate. We will work on the improvement of this protocol and meet our future goal.

REFERENCES

[1]. I.F.Akyildiz, T. Melodia, and K. R. Chowdhury (2007) , *A survey on wireless multi- media sensor networks,Computer Networks*, pp.931-950.

[2]. J. Campbell, P.B. Gibbons, S. Nath, P. Pillai, S. Seshan, R. Sukthankar (2005), *IrisNet: an Internet-scale architecture for multimedia sensors*, in: Proc. of the ACM Multimedia Conference.

[3]. I.F.Akyildiz, K. R. Chowdhury, *Wireless multimedia sensor networks- A survey*, Computer Networks, 921-960.

[4]. I.F. Akyildiz, M.C. Vuran, O.B. Akan, W.Su (2005), *Wireless Sensor Net-works: A Survey Revisited*, Computer Networks Journal (Elsevier).

[5]. E. Gurses, O.B. Akan (2005), *Multimedia Communication in Wireless Sensor Net- works*, in Annals of Telecommunications, 60(7-8), 799-827.

[6]. K. Akkaya and M. Younis (2005),*A Survey of Routing Protocols in Wire-less Sensor Networks*, in Proceedings of Elsevier Ad Hoc Network Journal, 325-349.

[7]. S. Misra, M. Reisslein, G. Xue (2008), *A survey of multimedia streaming in wireless sensor networks*, in IEEE Communications Surveys and Tutorials.

[8]. W. Lou, *An efficient N-to-1 mutlipath routing protocol in wireless sensor networks*, Proceedings of IEEE international Conference on Mobile Ad-hoc and Sensor Systems (MASS), Washington, DC, November 2005.

[9]. A. V. Sutagundar, S. S. Manvi, Kirankumar. B. Balavalad, *Energy Efficient Multipath Routing Protocol for WMSN*, International Journal of Computer and Electrical Engineering, Vol. 2, No. 3, June, 2010, 1793-8163.