Energy Efficient Routing in a MANET using Modified OLSR

Shwetha Parthiban, Pallavi Joshi, Indraneel Sanyal, Graduate Students, Department of Electrical and Computer Engineering, University of Florida, Gainesville, Florida-32608

Abstract-Energy depletion is one of the major problems affecting mobile adhoc networks. The project deals with developing an energy efficient algorithm to increase the network lifetime of a MANET by modifying the OLSR protocol for calculating the routing table in terms of the residual energy of the mobile nodes and the link cost in contrast with the conventional number of hops. Modified OLSR and standard OLSR were simulated under different mobility scenarios in NS-3.

Keywords—MANET, MPR, OLSR, residual energy, network lifetime

# Introduction

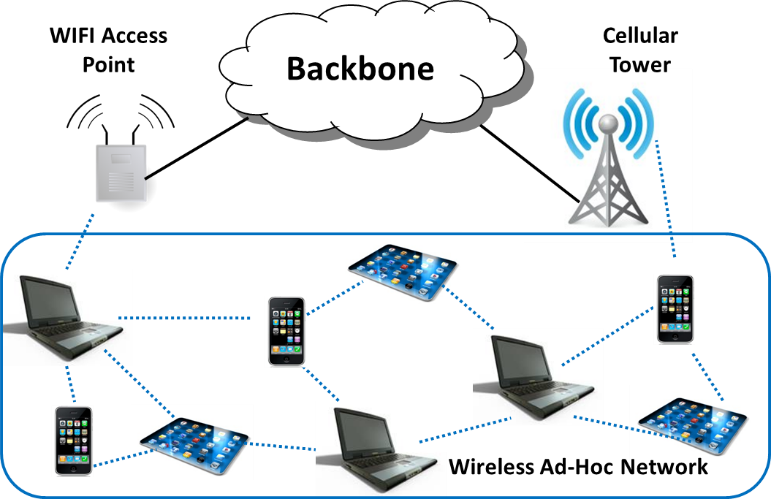
Tremendous growth of mobile communications, smartphones, tablets etc. resulted in growth of high-rate multimedia wireless services.D2D communications are highly preferred in 3GPP and LTE to reduce the investment for the infrastructure for this high growth in mobile communications. [Shwetha’s ppt slides]Mobile Adhoc Networks (MANET) form the basis for D2D communications. MANETs are distributed, self-organizing infrastructure less networks where every node in the network acts a router to take its own decisions and manages its own routing table to route packets between the source and the destination. Due to the mobility of nodes, exchange of control messages between the nodes, overhearing and packet transmissions, energy depletion and thus reduced battery life are major obstacles concerning adhoc networks. Optimized Link State Routing(OLSR),a proactive minimum-hop routing scheme which involves periodic transmission of HELLO and TC messages by the nodes to convey network information is particularly used in dense mobile wireless networks, where delay in transmission is not acceptable. Another peculiarity of OLSR lies in broadcasting control messages to only nodes in its multipoint relay (MPR) set which reduces the additional overhead of flooding messages throughout the network. OLSR does not take into account the energy depletion of its neighbor nodes and continues to route messages through the minimum hop route, thus draining the energy of certain nodes very quickly and causing the network to collapse. In other words, this reduces the network lifetime, which is defined as the time by which the energy of any one node in the network is depleted completed. Energy efficient Modified OLSR implemented in this project takes into account the residual energy of the nodes in determining the link cost and thus changes the way the routing table is computed. The modified OLSR uses the greedy heuristic approach for routing table computation.

Fig .Device to device communication architecture

Figure 1 depicts the architecture of device to device communication involving infrastructure backbone network connecting to the infrastructure less wireless mobile adhoc network. The mobile adhoc network is responsible for taking its own routing decisions by using various routing protocols.

1. Reference Paper Description and Analysis

The reference paper titled “Multi-Metric Energy Efficient Routing in Mobile Adhoc Networks” authored by Evripidis Paraskevas, Subir Das, John S Baras and Kyriakos Manousakis from the Department of Electrical and Computer Engineering, University of College Park. It is a 6 page paper published at the 2014 IEEE Military Communications Conference. The paper’s primary aspiration is in expanding and modifying the OLSR protocol in such a way so as to increase the network lifetime of the nodes by reducing the energy consumed in MANET. After devising the routing scheme, we integrate it into OLSR and run it under various scenarios of static and dynamic scenarios in NS-3 and develop a comparative analysis of the modified OLSR and Standard OLSR with respect to other quantities.



Fig2. Modified OLSR packet transmission diagram

OLSR inherits the stability of the link state algorithm, but due to its proactive nature, it has the additional advantage of providing routes as and when needed. The reason is the routing table of each node gets computed from time to time, thereby providing instant communication between two nodes. The nodes broadcast Hello and Topology Control Packets to its 1-hop neighbors who then updates its table about each destination node. The most important concept of OLSR is MultiPoint Relay (MPR). The MPR nodes are in 1-hop neighborhood of the source nodes which are selected specifically to forward the topology control packets. The MPR actually multicasts the packets rather than broadcasting of information to unwanted nodes and reducing the overhead as well as the energy require to transport unnecessary packets. Each and every node in OLSR maintains its own MPR selection set.

The OLSR protocol is modified such that it computes its routing table not on the basis of number of hops, but on the basis of path link cost from source to the destination node. The routing scheme is designed containing parameters like MAC queue utilization, residual energy and node degree. All these three parameters are then assigned a certain weight and the cost is computed on the basis of the weights of all these three parameters.

MAC queue utilization represents the congestion in the nodes. So, with higher transmission of packets come high congestion and increase in energy consumption. Therefore, the algorithm assigns higher weights to the nodes with larger number of packets to transmit, thereby decreasing the congestion in the network. Residual energy of a node signifies the available energy of the node after each transmission of the packet. This parameter decides actually the next-hop of the packet. So the algorithm assigns larger weights to the node with small residual energy. Node Degree symbolizes the number of nodes in a 1-hop neighborhood. As energy dissipation is directly related to packet transmission, the nodes who only intend to receive the message should get the message and there should not be any unnecessary packet transmissions to other nodes.

The modifications in the TC packet format is to include the weights assigned by the routing scheme and inserting them in the packet overhead such that the weights and cost will come into play. The MPR computation in the algorithm will remain the same as of the conventional OLSR as it covers all the nodes in the 2-hop neighborhood. The computation of the routing table is changed by implementing the algorithm. As we have to include the link cost rather than the number of hops, the algorithm assigns the weights of the routing quantities described above together with the weights of the intermediate nodes to calculate the path link cost of the total route. In implementing so, we use the Greedy-Heuristic algorithm to update the routing table and computes the next destination based on the computed cost.



Fig.3. Modified OLSR flowchart

Where

– number of packets in queue

– residual energy

– degree of node

– maximum queue size

– initial energy of a node

= Number of nodes-1

The Greedy-Heuristic algorithm initializes cost equal to 1 for all the nodes. Then, examination of topology is carried out for the 1-hop neighborhood and we consider some of the cases explained as follows. The first case corresponds to adding the destination node address in the routing table and thereby computing the cost equal to the source node’s cost plus the cost of the weights involved in transmission of the packets. The second case is when the destination address is present in the source node’s routing table, so we determine the cost through the original node or plot a new route by computing the path cost. The third case is when if we don’t have any address corresponding to the node, we don’t add any entry and nor do we compute any route.

After making desired changes in the OLSR routing protocol program, simulations were carried out in the Network Animator of NS-3, transmission of packets through multihop concept was observed and thereby multihop transmission in NS-3 can be pictorially represented in NS-3.

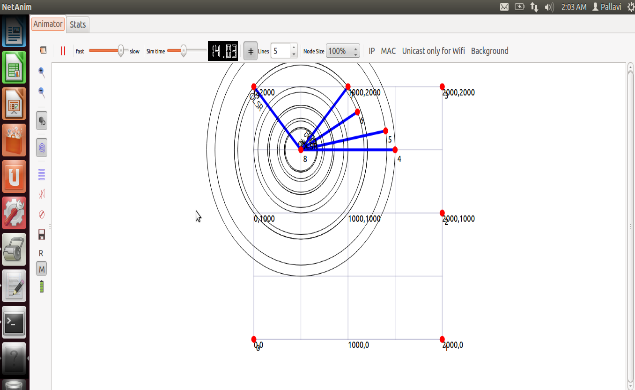


Fig.4. Network Animator Graph

**ALGORITHM1**: **Greedy Heuristic Algorithm**

1: ROUTING TABLE COMPUTATION

2: Clear Routing Table ()

3**: for** *i ←* 1*, N umber of Neighbors*

4: Add a new entry in the Routing Table

5: *Cost ←* 1

6**: end**

7: for *i ←* 1*, N umber of Topology Tuples*

8: *lasthopaddress ← Look Up in Routing Table for last address*

9: *destination address ← Look Up in Routing Table for destination address*

10**: if** *last address ← true &&destination address ← f alse* then

12: New Entry to be added in the Routing Table

13: destination *← desination address*

14: *nexthopaddress ← lastaddress.*

15: *interf ace ← last hop addres.interf ace*

16: *cost ← last hop address (Cost)* +*weight*

17: **else if** *have last hop address ← true && have destination address ← true* Then

18: Routing Table Update (lasthopaddress, destination address,tc\_cost) If Needed

19: **else**

20: No Routing Table Entry Required

21: **end if**

22: **end for**

23 **end all**

1. SIMULATOR DESCRIPTION AND SCENARIOS

All the programs and codes are designed and developed in Network Simulator-3 (NS-3). NS-3 is an open-source simulator developed and maintained by the NS-3 consortium. The simulator is a discrete event open source environment for developing and simulating the codes for networking. It is available under the GNU GPLv2 license for research, development and free educational use. The basic structure of the ns-3 is built around C++ and Python. The ns-3 simulation core supports research on both IP and non-IP based networks. The ns-3 library is built around python with the pybindgen library which then transfers the parsing of the C++ headers already present to gccxml to generate automatically the C++ binding code. Now, these automatically generated files will then be compiled in the python module which will then allow users to interact through python scripts.The simulation tests are calculated for 10 number of nodes with an area of 4sq.km. The size of each packet is 512 bytes with a transmitted power of 10dBm and initial node energy of 7 joules. Simulations was done for the following three scenarios:

1. Static Nodes
2. Mobile Nodes with speed of 2 m/s.
3. Mobile Nodes with speed of 20 m/s.

As NS-3 is based on scripting capability of C++, like C++ it also uses the concept of modularity. The operation of real-time network protocols are stored in the dozens of modules. The unique features of ns-3 are discussed as follows:

1. It is designed as a set of interlinked libraries which are combined together along with external libraries.
2. Ns-3 is primarily used in Linux systems although we still can use it in Windows Visual Studio and the support for other operating systems.

NS-3 simulator has been chosen because of a variety of reasons. The first one being the reference paper analysis and description and its algorithm have all been developed and carried out in NS-3. Secondly, being a relatively new software and all the development and maintenance are still being researched and worked on. Third reason being inclination on learning and understanding of the software and learning the basics of the language programing coupled with socket programming.

The description of the performance parameters are as follows:

1. **Distribution of Node Residual Energy:** This quantity depicts the overall picture of the energy consumption in the network. It gives us the number of nodes with the same percent of energy after the simulation has ended.
2. **Average Residual Energy Per Unit Time**: It gives us the average consumption of energy over per unit time across the complete network.
3. **Network Lifetime:** It signifies the amount of time after which the battery of a node of the network dies out.

SIMULATION SETUP**:**

Two broad scenarios are used and are described below:

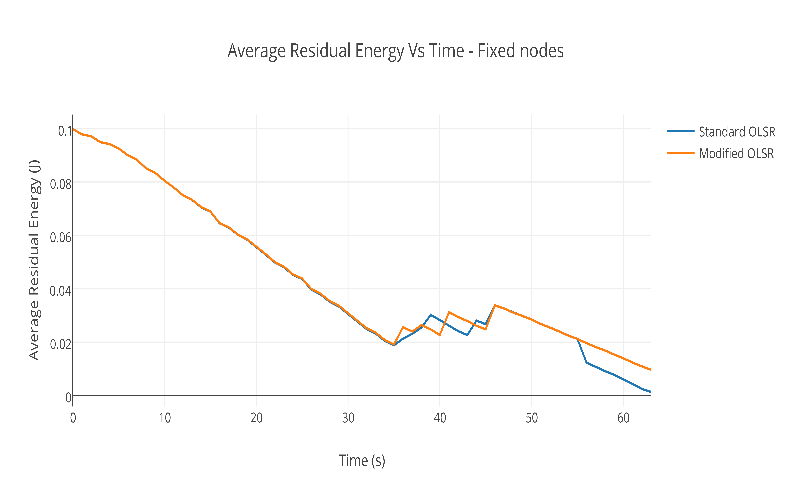
1. **STATIC SCENARIO**: The static scenario with nodes stationery will determine and depict the performance of the protocol only on the basis of energy consumption. The metrics are: Average Residual Energy with Respect to Time and Distribution of Node Residual Energy. The

simulation time for the scenario is set to 100 seconds.

1. **DYNAMIC SCENARIO**: The dynamic scenario comprises of two sub scenarios: low mobility scenario and high mobility scenario. In the dynamic scenarios, the nodes are mobile and travelling with a speed of 2 m/s and 20 m/s. The performance metrics considered here is Network Lifetime which is defined above.

SIMULATION RESULTS:

STATIC SCENARIOS:

  
Fig.5.Graphical Representation of Average Residual Energy v/s time for Fixed Nodes

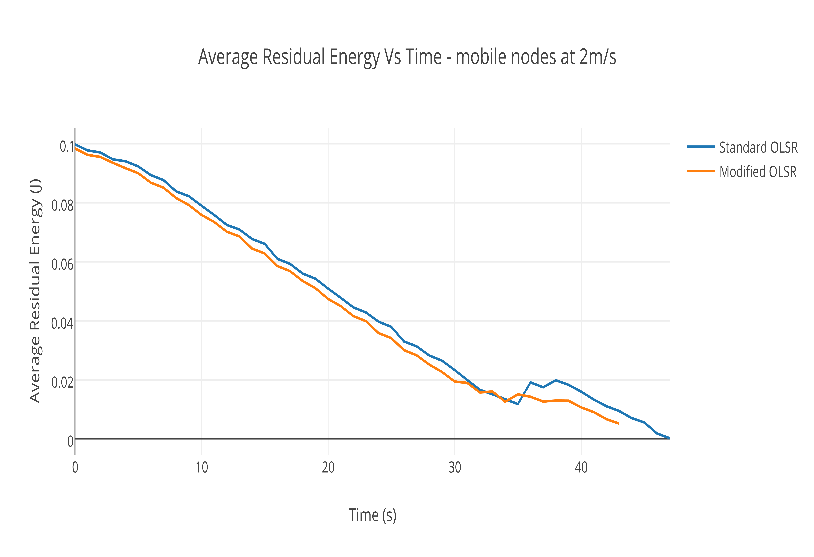
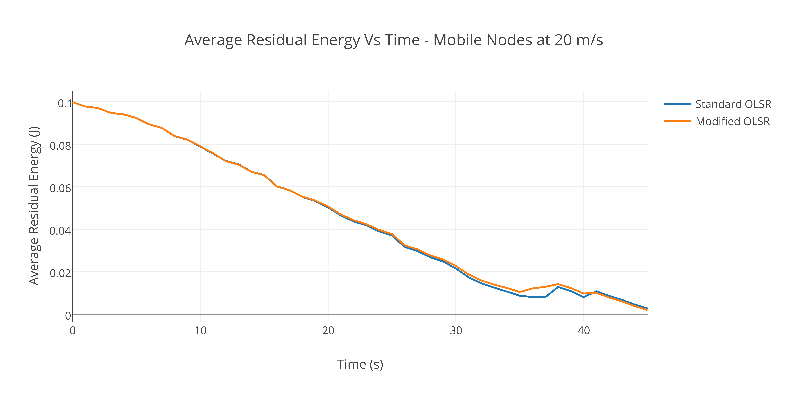


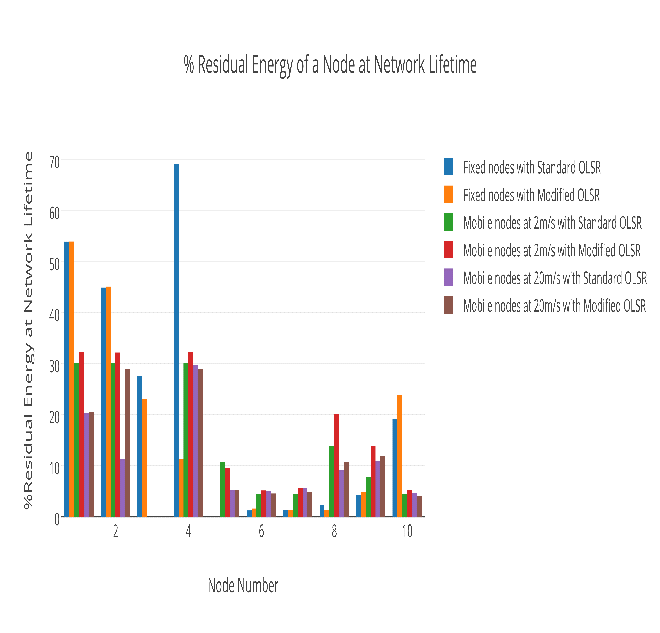
Fig.6.Graphical Representation of Average Residual Energy v/s time for Low Mobility Scenario



**Fig.7.Graphical Representation of Average Residual Energy v/s time for High Mobility Scenario**

|  |  |  |
| --- | --- | --- |
| Network Lifetime | Standard | Modified |
| Fixed OLSR | 35.5201 | 35.4891 |
| Mobility 2 m/s | 33.6965 | 32.2357 |
| Mobility 20 m/s | 33.7611 | 33.7013 |

**TABLE1: Summary of Network Lifetime for Standard OLSR and Modified OLSR**



**Fig.8. Distribution of all scenarios for complete comparison**

As evident from the graphs, observation is though the network lifetime is not affected much by modifying the parameter, the average residual energy v/s time is little bit higher in the case of modified OLSR as compared to the standard OLSR.

Write some description

1. CONCLUSION

The project introduces and does an elementary work in modifying the OLSR protocol with respect to the paper and worked on integrating the parameters introduced in above with the OLSR protocol and achieving the desired results.

Using residual energy as the weight, the path link cost is computed and the protocol identifies cost as the main parameter rather than number of hops. Evaluation of Modified OLSR is carried out under various simulation setups with different simulation objectives. The modified OLSR has a significant increase in the network lifetime.NS-3 is based on C++.However, several inbuilt classes and files in NS-3 make it difficult to understand initially. After getting a grip on the existing libraries, one gets a certain command over NS-3. A major doubt that daunted us was the concept of source and sink. After implementing the OLSR initially, we were unable to see multiple hops using Network Animator, which was later resolved by a careful observation of the routing tables corresponding to each node and by decreasing the transmission power and increasing the distance between the nodes so that all nodes are not within one hop of each other. Finding residual energy was yet another task as we needed to use as many pointers as the number of nodes in the network to access it’s residual energy from the basic energy source class. It was difficult to pass the residual energy and weights to the function for routing table computation due to constraints introduced by the highly complex MakeCallback function, which was avoided by using the BasicEnergySource smart pointer in olsr-routing-protocol.cc and and by declaring the weights as a global variable.

1. ACKNOWLEDGEMENT

We would like to thank Dr. Janice McNair and Mr. Gokul Bhat for the help and cooperation rendered towards us.

1. REFERENCES

[1] Evripidis Paraskevas*∗*, Kyriakos Manousakis*†*, Subir Das*†* and John S. Baras “Multi-Metric Energy Efficient Routing in Mobile Ad-Hoc Networks”, 2014 IEEE Military Communications Conference.

[2]Description of OLSR protocol- RFC 3626- https://www.ietf.org/rfc/rfc3626.txt

[3] [www.olsr.org](http://www.olsr.org)

[4]NS Documentation-<https://www.nsnam.org/docs/release/3.18/tutorial/ns-3-tutorial.pdf>

[5]Gustavo J.A.M. Carneiro- <https://www.nsnam.org/tutorials/NS-3-LABMEETING-1.pdf>

[6]NS-3 Description- <http://www.iitg.ernet.in/cse/rana2013/material/day5part2/presentation.pdf>

[7] Arash Asadi, Qing Wang and and Vincenzo Mancuso, “A Survey on Device-to-Device Communication in Cellular Networks”, April 2014, Pages: 1801 – 1819.

[8] Daquan Feng, Lu Lu, Yi Yuan-Wu, Geoffrey Ye Li, Shaoqian Li, and Gang Feng, “Device-to-Device Communications in Cellular Networks”, April 2014, Pages: 49 – 55

[9] http://www.cs.sjtu.edu.cn/~fwu/res/fig-NSFC11.png

[10] Abdalla, G.M.T., Abu-Rgheff, M.A.and Senouci,S.M ,” Current trends in vehicular ad-hoc networks”, IEEE Global Information Infrastructure Symposium, 2007, Pages: 1−7

[11] Sensor Networks: Wireless Communications and Networking, Vijay K.Garg

[12] <http://www.ics.uci.edu/~keldefra/manet.html>

[13] http://www.ericsson.com/research-blog/5g/device-device-communications/

[14] Ash Mohammed Abbas, Oivind Kure, “Quality of service in mobile Adhoc Networks: A survey”, International Journal Adhoc and Ubiquitous computing.