

Analysis Of Energy Efficient Routing Protocols In Mobile Ad Hoc Network

Mohammad Rafiq
Pursuing M.Tech(CSE)
SRM University,NCR Campus
Modinagar, India

Manoj Kumar
Pursuing M.Tech(CSE)
SRM University,NCR Campus
Modinagar, India

Ms.Sachi Pandey
Asst.Professor (CSE)
SRM University,NCR Campus
Modinagar , India

Abstract

Mobile ad hoc networks (MANET) represent distributed systems that consist of wireless mobile nodes that can freely and dynamically organize itself into temporary ad hoc network topologies. A mobile Ad hoc network is a collection of nodes that is connected through a wireless medium forming rapidly changing topologies. MANETS are infrastructure less and can be set up anytime, anywhere. we compared the energy consumption behaviour of three routing protocols; Ad hoc On Demand Distance Vector (AODV), Dynamic Source Routing (DSR) and the Destination Sequenced Distance Vector Routing (DSDV) with respect to energy consumption. We analyse these routing protocols by extensive simulations in ns-2 simulator and show that how pause time affect their performance. In this study performance is measured in terms of Average Remaining Energy, Average Consumed Energy and Network Life Time.

1. Introduction

In MANET, each node acts both as a router and as a host & even the topology of network may also change rapidly. These types of networks assume existence of no fixed infrastructure [1]. The Communication in MANET takes place by using multi-hop paths. The density of nodes and the number of nodes are depends on the applications in which it is being used. The mobile hosts can move randomly and can be turned on or off without notifying other hosts. If two wireless hosts are out of their transmission ranges in the ad hoc networks, other mobile hosts placed between them can forward their messages, which

effectively build connected networks among the mobile hosts in the deployed area.

2. Routing Protocols

In this section we briefly review the studied routing protocols

2.1 Ad Hoc on-Demand Distance Vector Routing (AODV)

AODV provides on-demand route discovery in MANET. Whenever the nodes need to send data to the destination, if the source node doesn't have routing information in its table, route discovery process begins to find the routes from source to destination. Route discovery begins with broadcasting a route request (RREQ) packet by the source node to its neighbours. RREQ packet comprises broadcast ID, two sequence numbers, the addresses of source and destination and hop count. The intermediary nodes which receive the RREQ packet could do two steps: If it isn't the destination node then it'll rebroadcast the RREQ packet to its neighbours. Otherwise it'll be the destination node and then it will send a unicast replay message, route replay (RREP), directly to the source from which it was received the RREQ packet. A copied RREQ will be ignored. Each node has a sequence number. When a node wants to initiate route discovery process, it includes its sequence number and the most fresh sequence number it has for destination. The intermediate node that receive the RREQ packet, replay to the RREQ packet only when the sequence number of its path is larger than or identical to the

sequence number comprised in the RREQ packet. A reverse path from the intermediate node to the source forms with storing the node's address from which initial copy of RREQ. There is an associated lifetime

value for every entry in the routing table. Suppose that some routes are not applied within their lifetime period, so these routes are expired and should be dropped from the table. But if routes are used, the lifetime period is updated so those routes are not expired. When a source node wants to send data to some destination, first it searches the routing table; if it can find it, it will use it. Otherwise, it must start a route discovery to find a route [4]. It is also Route Error (RERR) message that used to notify the other nodes about some failures in other nodes or links.

2.2 DSR – Dynamic Source Routing (DSR)

DSR is a reactive routing protocol that discovers and maintains routes between nodes. In the route discovery, DSR floods Route Request Packet to the network [4]. Each node that receives this packet, first add its address to it and then forwards the packet to the next node. When the targeted node or a node that has route to the destination receives the Route Request, it returns a Route Reply to the sender and a route is established. Each time a packet follows an established route, each node has to ensure that the link is reliable between itself and the next node. In the Route maintenance, DSR provides three successive steps: link layer acknowledgment, passive acknowledgment and network layer acknowledgment. When a route is broken and one node detects the failure, it sends a Route Error packet to the original sender

2.3 Destination-Sequenced Distance-Vector Routing

The Destination-Sequenced Distance-Vector Routing protocol (DSDV) described in [1] is a table-driven algorithm based on the classical Bellman-Ford routing mechanism [1]. The improvements made to the Bellman-Ford algorithm include freedom from loops in routing tables. Every mobile node in the network maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. To help alleviate the potentially large amount of network traffic that such updates can generate, route updates can employ two possible types of packets. The first is known as a full dump. This type of packet carries all available routing information and can require multiple network protocol data units (NPDUs). During periods of occasional movement,

these packets are transmitted infrequently. Smaller incremental packets are used to relay only that information which has changed since the last full dump. Each of these broadcasts should fit into a standard-size NPDU, thereby decreasing the amount of traffic generated. The mobile nodes maintain an additional table where they store the data sent in the incremental routing information packets. New route broadcasts contain the address of the destination, the number of hops to reach the destination, the sequence number of the information received regarding the destination, as well as a new sequence number unique to the broadcast. The route labelled with the most recent sequence number is always used. In the event that two updates have the same sequence number, the route with the smaller metric is used in order to optimize (shorten) the path. Mobiles also keep track of the settling time of routes, or the weighted average time that routes to a destination will fluctuate before the route with the best metric is received.

3. Simulation Performed Using NS-2

We have used Network Simulator (NS)-2 in our evaluation. The NS-2 is a discrete event driven simulator developed at UC Berkeley. NS-2 is suitable for designing new protocols, comparing different protocols and traffic evaluations. It is an object oriented simulation written in C++, with an OTcl interpreter as a frontend. Simulation of protocols is performed on Linux operating system using ns-2.34. We have different simulations run in all over. Every simulation runs from 0s to 200s. Random waypoint mobility in a rectangular field of 400m * 400m is used.

3.1 Simulation Steps

1) Scenarios are generated using setdest utility which uses random waypoint mobility model. Example to generate scenario is given as:

```
Setdest -v1 -n 30 -p 0.0 -M 4 -t 100 -x 500 -y 500 > scene5
```

Where -v : version 1 or 2 , -n: number of nodes , -p : pause time , -M : maximum speed , -x and -y : area of simulation, -t : simulation time , scene5 : output file.

2) Traffic pattern is generated using cbrgen.tcl file given in indep utilities. In this simulation only one traffic pattern is generated using following method:

```
ns cbrgen.tcl -type cbr -nn 9 -seed 1 -mc 7 -rate 4 > rafiq7
```

Where -type: type of traffic cbr or tcp, -nn: number of nodes, -seed: seed value, -mc: maximum connection sources, -rate: rate of sending packets.

3) After generating traffic patterns and scenarios a tcl script is written for the generation of the trace files.

These created traffic patterns and scenarios are fed in to tcl script and then executed. On execution of tcl script trace files are generated. In this simulation two protocols namely AODV, DSR and DSDV are used to generate trace files which are saved with the extension *.tr which are old trace file formats. There are two trace file formats available one is old trace file format and other is new trace file format. With generation of trace file a *.nam file is also generated which shows animation of the moving nodes and routing of packets. Routing of packets and movement of nodes can be easily depicted by *.nam files.

4) When trace files are generated then it is needed to analyse these files. To analyse files awk or perl scripts are written according to performance metrics which are to be used in performance evaluation. This simulation is performed to evaluate the performance based on metrics namely Average Remaining Energy, Average Consumed Energy and Network Life Time . So Three awk files are used for this simulation.

5) After the analysis of trace files obtained results are stored in a *.xgr file from which x graphs are generated by using Xgraph utility of ns-2.

3.2 Performance Metrics.

There are many performance metrics which are used for analysis of various protocols. In this project I am using three performance metrics, which are:

Average Remaining Energy/Average System Energy: It is taken as the average of the remaining energy levels of all the nodes in the network at the end of simulation.

Average Consumed Energy: It is taken as the average of the consumed energy of all the nodes in the network at the end of simulation.

Network Life Time: It is the time when a node finished its own battery for the first time. The performance is better when network life time is high [5].

3.3. Simulations and Results

The simulations were performed using Network Simulator 2 (Ns-2) , particularly popular in the ad hoc networking community. The mobility model used is Random Way point Model. The traffic sources are CBR (continuous bit rate), number of data connections is 7, data packet size is 512 byte and data sending rate is 4 packets/second. The source destination pairs are spread randomly over the network in a rectangular field of 400m x 400m. During the simulation, each node starts its journey from a random spot to a random chosen destination. The simulation time is 200 seconds and maximum speed of nodes is 20 m/s. The

interface queue is 50- packet drop-tail priority queue. Network scenario for different number of nodes are generated. Impact of changing pause time. In this simulation pause time is varying and considered 0, 50,100,150 and 200 and other network parameters are considered as in the table 1.

Table -1 Simulation Parameters

Parameters	Values
Routing Protocols	AODV,DSDV,DSR
Simulation time	200 sec
Traffic Type	CBR
Maximum Connections	7
Sources	5
Packet Rate	4 (pkts/sec)
Packet Size	512 bytes
Pause Time	0,50,100,150,200 (sec)
Number of nodes	30
Network Area	400mx400m
Transmission Range	250m
Maximum Speed	20m/sec
Mobility Model	Random Waypoint
Interface Queue	50 Packet Drop-tail Priority
Mac type	802_11
Initial energy	200 J (146 J in case of network life time)
Idle power	0.73 W
Tx power	1.4 W

Rx power	0.9 W
Sleep power	0.001 W
Transition power	0.1 W
Transition time	0.005 sec
Antenna Type	Omni-Directional

We simulated this network under each of routing protocols and outputs shown in Figs. 1-3. Figs. 1-3 show a comparison between the routing protocols as a function of pause time.

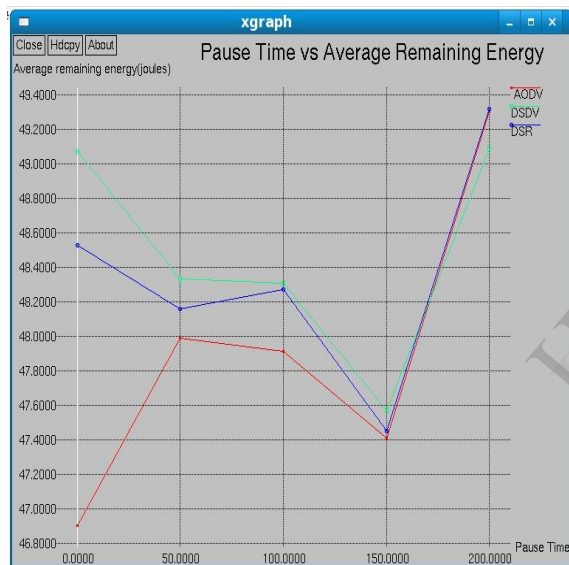


Figure 1. Pause Time versus Average Remaining Energy

Figure 1 shows effect of varying pause time on remaining energy . It shows that DSDV performs better than DSR and AODV most of time . When pause time is 0 DSDV performs better than other two protocols but when we make network static (i.e pause time = simulation time = 200) AODV and DSR performs better in terms of average remaining energy.

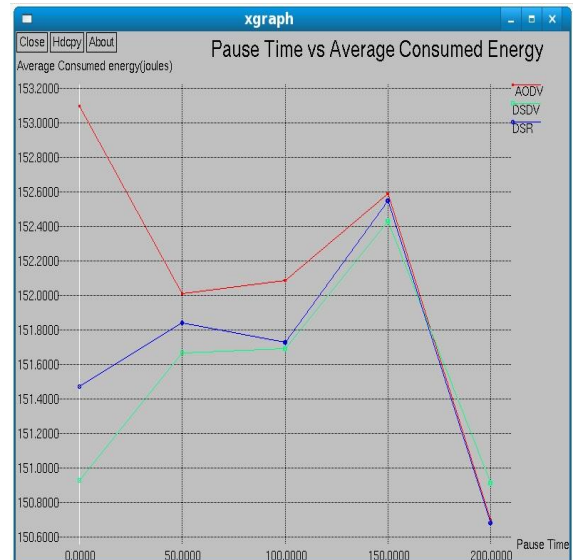


Figure 2. Pause Time versus Average consumed Energy

Figure 2 shows that DSDV Consumes less energy than AODV and DSR most of time but for static network AODV and DSR consumes less energy than DSDV.

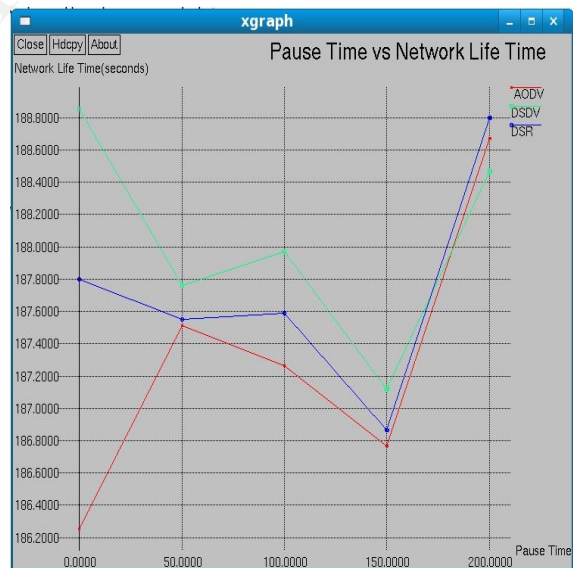


Figure 3. Pause Time versus Network Life Time

Figure 3 shows effect of varying pause time on network life time . It shows that most of time network life time is higher when we are using DSDV and network life time is lower for AODV. For static network network life time is higher when we are using DSR.

4. CONCLUSION

This paper is an attempt to evaluation performance of three commonly used mobile ad hoc routing protocols namely AODV, DSDV and DSR. Performance evaluation is performed in NS-2 simulator by doing many simulations. Comparison was based on Average Remaining Energy, Average Consumed Energy and Network Life Time . Simulation results are shown by many figures. By using simulation results we can understand that DSDV gives better performance in wide range of simulation condition for this network. In future we will compare these protocols by varying number of Sources and speed of nodes.

5. ACKNOWLEDGMENTS

Our thanks to our parents and Mr. Rakesh Yadav (Asst.Professor SRM University) who have contributed towards development of the paper .

6. REFERENCES.

- [1] Beigh Bilal Maqbool and Prof. M. A. Peer “ Classification of Current Routing Protocols for Ad Hoc Networks - A Review” International Journal of Computer Applications Volume 7– No.8, October 2010
- [2] Manijeh Keshtgary and Vahide Babaiyan, “Performance Evaluation of Reactive, Proactive and Hybrid Routing Protocols in MANET ” International Journal on Computer Science and Engineering (IJCSE). Vol. 4 No. 02 February 2012.
- [3] Jun-Zhao Sun, “Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing”, Info techand Info-net, vol.3, 2001, pp316 – 321.
- [4] G. S. Mamatha and Dr. S. C. Sharma, “Analyzing the MANET Variations , Challenges, Capacity and Protocol Issues”, International Journal of Computer Science & Engineering Survey (IJCSES) Vol.1, No.1, August 2010.
- [5] Mina Vajed Khiavi, Shahram Jamali, Sajjad Jahan bakhsh Gudakahriz “Performance Comparison of AODV, DSDV, DSR and TORA Routing Protocols in MANETs” International Research Journal of Applied and Basic Sciences. Vol., 3 (7), 1429-1436, 2012.
- [6] Bilal Maqbool, Dr. M. A.Peer, and S.M. K.Quadri “Towards the Benchmarking of Routing Protocols for Ad hoc Wireless Networks” IJCST Vol. 2, Issue 1 March 2011.
- [7] Dinesh Singh, Deepak Sethi, Pooja “ Comparative Analysis of Energy Efficient Routing Protocols in MANETS (Mobile Ad-hoc Networks)” International Journal of Computer Science and Technology Vol. 2, Issue 3, September 2011.
- [8] K. Arulandam and Dr.B.Parthasarathy “A new energy level efficiency issues in MANET” International Journal of Reviews in Computing 2009.

[9] Mobile communications by J Schiller pearson 2nd edition.

[10] Ad Hoc Networking, Perkins, Addison Wesley, 2001.

[11] Marc Greis, Ns Tutorial, <http://www.isi.edu/nsnam/ns/tutorial/>, accessed on Sep 2012

[12] TCL Tutorial, [http://www.tcl.tk /man/tcl/tutorial/tcltutorial.html](http://www.tcl.tk/man/tcl/tutorial/tcltutorial.html), accessed on Sep 2012.

[13] Trace formats, [http://nsnam.isi.edu /nsnam/index.php/NS-2_Trace_Formats](http://nsnam.isi.edu/nsnam/index.php/NS-2_Trace_Formats) , accessed on Jan 2013

[14] Awk basics,[http://www.thegeekstuff .com/2010/01/awk-introduction -tutorial-7-awk-print-examples/](http://www.thegeekstuff.com/2010/01/awk-introduction-tutorial-7-awk-print-examples/) , accessed On Jan 2013.