An Energy-Efficient Dynamic Source Routing Protocol for Mobile Ad Hoc Networks

NAIGENDE DUNCAN‡
College of Computing and Information Sciences,
Makerere University Kampala

BULEGA TONNY EDDIE (PhD)
College of Computing and Information Sciences,
Makerere University Kampala

ABSTRACT

The Dynamic Source Routing Protocol (DSR) is one of the most reliable and effective protocols in the Mobile Ad Hoc Networks (MANETs). It is also one of the few MANET protocols whose routing scheme can easily be optimized. But the routing overhead generated by its routing algorithm still leaves substantial amounts of energy being wasted. Route Request (RREQ) and Route Maintenance packets generate overhead control packets that occupy bandwidth, consume energy and may overwhelm a network if not controlled. This paper proposed EEDSR, an extension of DSR that reduces routing overhead by limiting the number of route discovery and maintenance packets in the MANET. The scheme involves bigger packet headers for the source route discovery packets since they contain information about the energy levels of the nodes in the route cache. In EEDSR, since the RREQ packets are flooded once for each communication period, routing overhead is minimized.

General Terms: Routing Overhead, Dynamic Source Routing (DSR), Energy-Efficiency

Additional Key Words and Phrases: Energy-Efficient Dynamic Source Routing (EEDSR), Mobile Ad Hoc Networks (MANETs), Power-aware, On-demand routing.

IJCIR Reference Format:
Duncan Naigenda and Tonny E. Bulega. An Energy-Efficient Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. International Journal of Computing and ICT Research, Vol. 6 Issue 2, pp 23-32. http://www.ijcir.org/volume6-issue2/articler3.pdf.

---

‡ Author's Address: Duncan Naigenda and Tonny E. Bulega, College of Computing and Information Sciences, Makerere University Kampala.

International Journal of Computing and ICT Research, ISSN 1818-1139 (Print), ISSN 1996-1065 (Online), Vol.6, Issue 2, pp. 23-32. December 2012.

1. INTRODUCTION

Mobile ad-hoc networks (MANETs) are wireless networks with no fixed infrastructure [Royer and Toh 1999]. MANET nodes can either be hosts or can act as routers when the two end-points are not directly within their radio range. A critical issue for MANETs is that nodes are normally power constrained [Djenouri and Badache 2006]. Available battery technology is not growing fast enough to meet this constraint. It is via routing and routing protocols that we can possibly alleviate this constraint. Extensive research in routing protocols for MANETs has been carried out, with particular emphasis being placed on reactive routing protocols as opposed to proactive ones at saving energy [Gikaru 2004].

Among the energy-efficient routing protocols, DSR has been found to be very useful especially in developing new power-aware routing protocols [Tamilarasi et al. 2008]. However, the continuous flooding of route request (RREQ), route reply (RREP) and route error (RERR) packets by the DSR algorithm brings with it high routing overhead that causes substantial energy exhaustion of the nodes. While previous research has looked at minimizing routing overhead as a means to saving node energy in DSR, few if any have looked at controlling the frequency of flooding the RREQ packets.

This paper proposes EEDSR, an extension of DSR that looks at controlled and periodic flooding of RREQ packets as opposed to that in the original DSR algorithm. At every communication period T, the algorithm sets E_{thresh} , the energy threshold and goes ahead to flood RREQ packets. It is only when 75% of the network nodes have energy below E_{thresh} , that another flooding of RREQ packets in the MANET is done.

2. RELATED WORK

In [Djenouri and Badache 2006], an extended DSR algorithm prolongs node lifetimes by basing on prediction of nodes' mobility patterns. It distributes the routing task to almost all network nodes.

In [Murugan et al. 2005] the DSR-C (DSR-Cache) reduces routing overheads and improves route discovery latency by utilizing a higher cache hit rate to minimize constant dissemination of routing information by the nodes.

In close relation to [Murugan et al. 2005], [Tamilarasi et al. 2008] reduces routing overhead by cutting down the number of route reply packets during route discovery to one selected route. Unlike the standard DSR, the header size of the data packets is also reduced. Data packets are also transmitted with as minimum energy as possible.

In [Gikaru 2004], there is an in depth approach to minimizing routing overhead. It is controlled at both node level and network level. The scheme is designed for all MANET routing protocols although the tests and the algorithm are done based on AODV. But DSR being better than AODV at energy optimization [Jiang and Garcia-Luna-Aceves 2001], extending DSR would result into a more energy-efficient protocol.

This paper is structured such that Section 1 introduces the work. Section 2 highlights related research on saving energy via minimizing routing in DSR. Section 3 gives a brief overview of the basic DSR protocol and outlines the adjustments made on the DSR algorithm. In Section 4, simulation results are presented and analyzed. We conclude THE findings in Section 5.

3. DSR PROTOCOL OVERVIEW

We make a brief presentation of the basic DSR protocol with emphasis on its route request, route reply and route maintenance behavior in the algorithm. A more thorough presentation of EEDSR shall be made, since it is mainly the impact of its attributes to energy optimization that we want to focus on in this paper.

3.1 Dynamic Source Routing Protocol (DSR)

The Dynamic Source Routing [Johnson et al. 2007] is an on-demand protocol based on source routing. It consists of two main mechanisms that allow the discovery and maintenance of routes in the MANET. In the Route Discovery mechanism, a source node, S wishing to send a packet to a destination node, D obtains a source route to the destination. If the source does not have a route to the destination, it performs a route discovery by flooding the network with route request (RREQ) packets [Sathish et al. 2011]. The RREQ packet contains the destination node address, the source node address and a unique *Request ID* [Doshi et al.

2003]. Any node that has a path to the destination in question can reply to the RREQ packet by sending a route reply (RREP) packet. The reply is sent via the route recorded in the RREQ packet [Sathish et al. 2011]. Several possible routes from *S* to *D* form a 'route cache'. If the 'route cache' possesses multiple routes to the destination, the routing logic selects the route with minimum hop to the destination.

In Route Maintenance, a node wishing to send a packet to a destination is able to detect, while using a source route to the destination, if the network topology and/or channel quality has changed [Johnson et al. 2007]. If this is the case then it must no longer use this route to the destination because a link along the route is broken. Route Maintenance for this route is used only when the source node is actually sending packets to the destination [Gikaru 2004]. In this case, route error (RRER) packets are sent to the source node via the intermediate nodes such that they update their caches by removing the route in error.

A routing entry in DSR contains all the intermediate nodes of the route rather than just the next hop information maintained in other reactive protocols. A source puts the entire routing path in the data packet, and the packet is sent through the intermediate nodes specified in the path [Royer and Toh 1999]. To limit the need for route discovery, DSR allows nodes to operate their network interfaces in promiscuous mode and snoop all (including data) packets sent by their neighbors. Since complete paths are indicated in data packets, snooping helps in keeping the paths in the route cache updated. To further reduce the cost of route discovery, the RREQs are initially broadcasted to neighbors only (zero-ring search), and then to the entire network if no reply is received. Another optimization feasible with DSR is the gratuitous route replies; when a node overhears a packet containing its address in the unused portion of the path in the packet header, it sends the shorter path information to the source of the packet (Automatic Route Shortening) [Jha et al. 2010].

Another important optimization includes the technique to prevent "Route Reply Storms" because many route replies may be initiated simultaneously a delay time proportional to the hop's distance can be used in order to give higher priority to near nodes.

DSR also employs "Packet Salvaging". When an intermediate node forwarding a packet detects through Route Maintenance that the next hop along the route for that packet is broken, if the node has another route to the packets destination it uses it to send the packet rather than discard it [Johnson et al. 2007].

3.2 Limiting Routing Overhead

This implementation of EEDSR includes two essential limitations:

- RREQ packets are sent periodically (once per communication period, *T)*. Strict route caching is done to minimize constant flooding of RREQ packets. Each node therefore ought to have a minimum energy required to route the standard packet size from itself to any destination in the network.
- Each node uses fixed transmit power/energy in order to avoid some nodes using more resources than others while performing similar tasks. In EEDSR route selection is dependent on minimum hop and remaining energy values of the nodes. The route selection is dependent on minimum hop and remaining energy values of the nodes. The route discovery and route maintenance mechanisms are so controlled for purposes of limiting routing overhead [Cameron 1998].

3.3 Optimal Route Discovery

Broadcasting of node power information in the whole network is done alongside RREQ packets as doing it separately may incur some overhead and also cause delays in data delivery. At the beginning of the route discovery process, shortest-hop is used as the metric since the nodes have same battery capacity/energy levels.

Shortest-hop metric here is still in consonant with minimum total transmit power. But as the nodes drain their batteries, remaining battery capacity is used as the metric. When an intermediate node receives a route discovery (RREQ) packet, the following steps are undertaken by the algorithm;

Step 1: Node checks the source node's energy level, E_s , source node *ID*, threshold energy value, E_{thresh}, that was advertised in the network, and the destination node *ID*. It saves these attributes in its route cache.

Step 2: It appends its node *ID*, energy level, E_i and forwards the packet to its neighbouring nodes.

Step 3: If the route discovery broadcast is not the first one (i.e, if it is another communication period, *T'*), the node will check its route cache to see if there exist routes to the destination.

Step 4: In case there is one, it chooses the one with shortest- hop and with minimum total energy to the destination. In case there are no routes, steps 1, 2 and 3 are performed.

3.4 Optimal Route Maintenance

The EEDSR route maintenance indicates change in link quality (especially in terms of energy). A link whose nodes do not possess the threshold energy, E_{min} or E_{thresh} minimum power required to receive and forward a packet to the next destination) shall be 'isolated' and later on dropped from active communication until the next period, *T*. flow of route maintenance packets, reduces the number of routing updates and conserves battery power in the process.

Optimal route maintenance packets are sent in the network when the following are experienced;

1- **Energy Depletion**: When any network node has its energy level or battery capacity going below the E_{thresh} for that period, route maintenance packets shall be sent to neighbor nodes informing them of 'isolation' of the depleted node from the network during the remaining time for that period.

2- **Excessive Mobility**: All network nodes take on a limited mobility level while running the EEDSR algorithm. If a node exceeds this level, communicating nodes shall 'neglect' it while forwarding packets. This is because in EEDSR, the positional stability of a neighbor node is a crucial factor. Packets shall be sent to less mobile nodes. The algorithm operates well in a fairly stable network topology. This limits the flow of route maintenance packets, reduces the number of routing updates and conserves battery power in the process.

A source route is considered 'broken' when the route it describes through the network is not viable. This is when more than 3/4 of the nodes in the route possess less than the required threshold power to forward the minimum data packet size, i.e,

$$\sum_{i=1}^{j-1} E_i \quad < \quad \tfrac{3}{4} E_{ij} \qquad (1)$$

where E_{ij} is the total energy of a route from node *i* to node *j* , E_i is the remaining energy in each node.

Each transmission has a specific quantity of data to forward. This in effect maximizes throughput while collectively increasing network lifetime as almost all nodes have a role to play in data forwarding. In addition, it reduces contention at the MAC layer hence avoiding or minimizing the flooding of route discovery packets.

3.5 Maximizing Network Lifetime

The EEDSR protocol algorithm selects routes such that the transmission and reception of packets is distributed on the network so as to maximize battery lifetime and therefore network lifetime. The remaining battery capacity of a node or host is taken as our measure for determining the node's lifetime. This in turn determines the network lifetime.

Assuming E_{ti} is the remaining energy or battery capacity of the node *i* at a time *t*. Node *i* will route the required minimum number of data packets with a probability *P*(*i*).

P(*i*) will be determined in terms of the remaining battery capacity or node energy E_{ti}. If E_{fi} is the battery capacity of a fully charged battery for node *i*, then,

$$P(i) \quad = \quad E_{ti} / E_{fi} , \qquad (2)$$

Where P(*i*) lies between 0 and 1 [Chen et al. 2004].

There is a minimum value for *P*(*i*) above which a node shall be considered fit to forward the required minimum number of data packets to the destination. A node whose value is below that shall not participate in data forwarding and shall therefore be ‘isolated’. This value is given by,

$$Pmin(i) = Emin/E_{fi}. \quad (3)$$

Where $P_{min}(i)$, is the minimum probability below which a node will be isolated [Doshi et al. 2003].

Therefore, for every communication period T,

$$Pmin(i) = Ethresh/Efi. \quad (4)$$

For each communication period T, $P_{min}(i)$, will be approximately the same for all nodes. The battery consumption model assumes that the node’s battery lifetime reduces linearly with time if the node is neither transmitting nor receiving packets. This therefore means that the ‘isolated’ node is likely to run out of battery energy even if it is not participating in communication. At the next Route-Request broadcast, such a node will not receive the broadcast packet. To avoid this, the ‘isolated’ node goes in ‘sleep mode’ and only wakes up when the new threshold value E'_{thresh}, is being set.

3.6 EADSR Algorithm

The EADSR protocol is an energy-aware version of DSR. It is obtained by adding a power header to the original DSR packet header [Bhadhare et al. 2003]. The power header is a list of power values that are the minimum transmit powers for each link. This power value for every link is propagated by each node by adding a field to the header with the value.

The RouteTable adds these power headers. It also extracts these values from the headers and adds the information to the link cache. An extra input port that processes snooped packets and discovers new minimum energy routes is added [Bhadhare et al. 2003]. When new routes are found it sends out a gratuitous route reply to the source of the packet. This process also involves constantly mentoring the power cost of each link in the route.

The DSR source header is modified to include an additional flag that indicates a change in the power costs of the links in the route. The destination node checks this flag and sends out a gratuitous reply to the source informing it about the new costs of the route.

The RequestTable elements is also modified to implement minimum transmit power computation to calculate the minimum transmit power value of the link. This value is then added to the power list in the route request packet when it has to be rebroadcast. The RequestTable element now propagates the new metric through the network.

4. SIMULATION AND RESULTS

Simulation was done using NS-2.34 Simulator in Ubuntu-10.04-desktop-i386.iso Linux environment. The process took three major steps of:

1. **Preparation process:** This involved preparation and derivation of OTcl scripts. It also involved generation of nodes, node movement and/or traffic patterns. This also helped implement the protocol when a combination of C++ and OTcl scripts were added into the NS-2’s source base.
2. **Simulation process:** Involved the description of the simulation in an OTcl script and actual introduction of the metrics and parameters into the simulation set up. It involved the practical simulation and derivation of the research data and results.
3. **Analysis of Results:** Results were analyzed via xgraph visualization using the generated trace files.

4.1 Simulation Environment

The simulations were carried out on DSR, EADSR and EEDSR under similar environments. The table I below shows the parameters that were considered in the simulations.

Table I. Simulation Parameters

Parameter	**Value**
Environment Size	800 * 800 m^2
Number of nodes	20
Initial node energy	1000 Joules
Traffic type	Constant Bit Rate (CBR)
Queue length	50
Mobility model	Random Way Point
Maximum node speed	20 m/s
Antenna type	Omnidirectional

4.2 Routing Overhead

Routing overhead is the sum of all transmissions of routing packets sent during the communication process. It is used here to compare the adaptation of the routing protocols to the limited bandwidth of MANETs, plus their efficiency in relation to node battery power. Generating more routing packets in the network increases the possibility of collision hence more energy consumption [Gikaru 2004]. Routing overhead is a determinant metric to many energy-efficient metrics like per packet cost (energy spent per packet). A high per packet cost means that much routing overhead has been incurred in transmitting one packet. This also translates into much energy spent per packet.

In figures 1 and 2, the average routing overhead for EEDSR is lower than for EADSR and DSR for all pause times. For lower pause times (less than 100), the EEDSR overhead are higher. This is due to the high node mobility which results into link breakages. Route caches are therefore visited more often to replace the broken routes. Soon, the caches run out and new route requests have to be flooded in the network. For higher pause times (100 to 400), there are less link breakages, hence lower routing overheads as the route caches are visited less often.

Figure 1. Average routing overhead against pause times for 10 traffic sources.

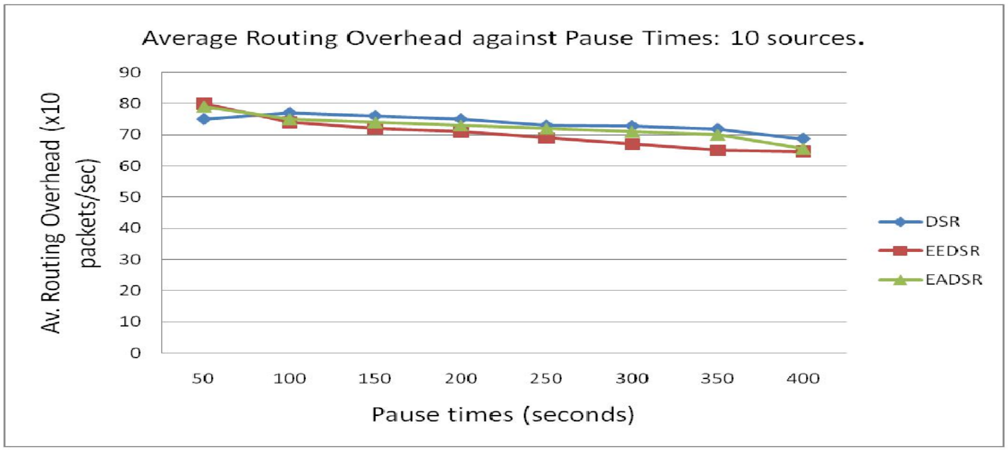
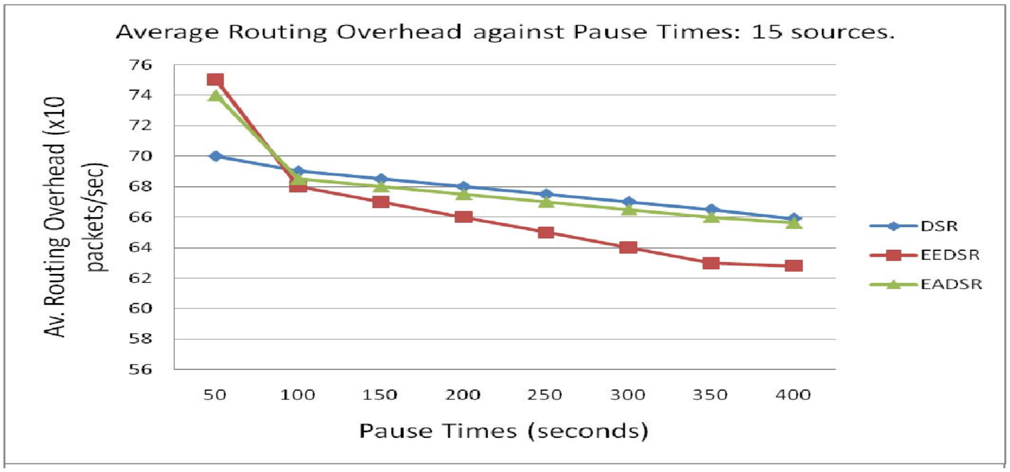


Figure 2. Average routing overhead against pause times for 15 traffic sources.



4.3 Network Throughput

Network throughput is the amount of data successfully sent and received (without errors) by the entire network within the simulated data transfer time. In energy terms, the higher the throughput the better performing the protocol [Tarique et al. 2010]. Nodes ensuring high network throughput are considered optimum with their energy sources, meaning that the underlying protocol algorithm is running efficiently.

Low throughput is the amount of data successfully sent and received (without errors) by the entire network within the simulated data transfer time. In energy terms, the higher the throughput the better performing the protocol [Djenouri and Badache 2006].

In figures 3 and 4, the poor network throughput for DSR is due to aggressive caching. There is no mechanism to expire stale routes or to determine freshness of routes when multiple choices are available. EEDSR improves this situation by 'isolating' low energy nodes and setting new communication periods. This improves freshness of routes and improves throughput in the process.

At high mobility (low pause times). EEDSR still has lower throughputs compared to EADSR. This is because the frequent route breakages result into invalidation of cached routes in EEDSR.

At low mobility (high pause times) throughput for EEDSR increases as there are less link breakages. The caches are visited less often, there are less overheads as RREQ and RERR packets are generated less.

Figure 3. Network throughput against pause times for 10 traffic sources.

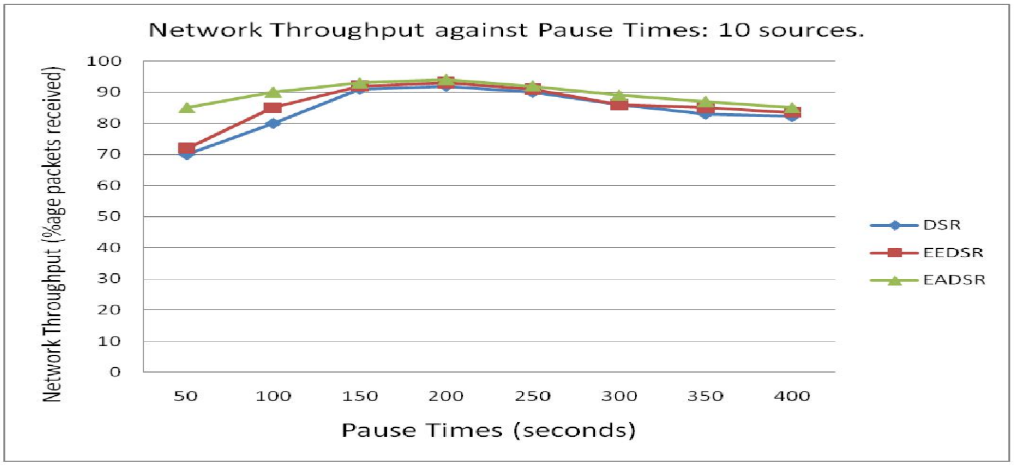
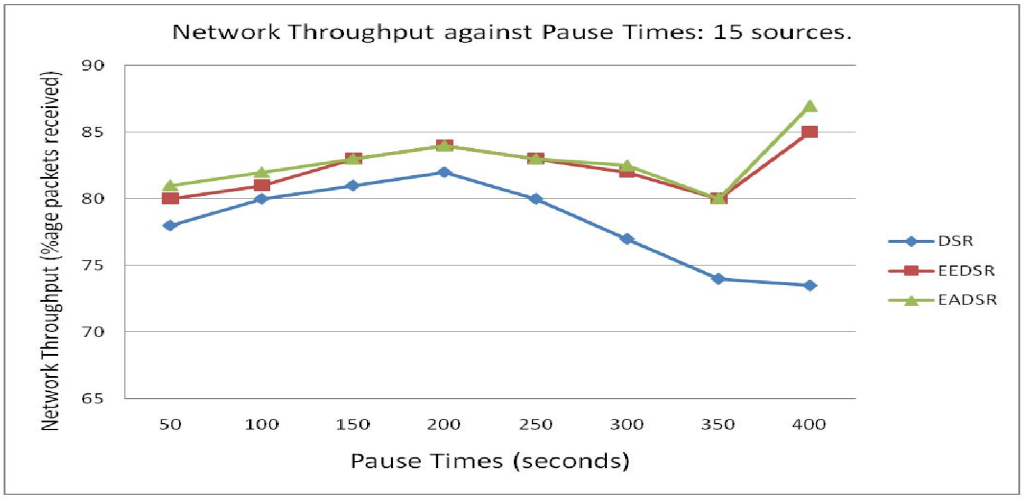


Figure 4. Network throughput against pause times for 15 traffic sources.



4.4 Network Lifetime

The remaining battery capacity of a node/host is taken as our measure for the node's lifetime. Here, network lifetime is defined as the time in seconds from the start of the simulation till when 75% of the total number of nodes in the network get exhausted of energy.

For high mobility (0 pause time), in figure 5, EEDSR and EADSR both have graphs that follow a pattern different from that of DSR. This is because both EEDSR and EADSR are energy-aware or power-aware algorithms whereas DSR is not. The network lifetimes keep decreasing with increasing sources because there will now be more packets generated at the nodes. The nodes forward more packets and therefore run out of battery energy faster.

In figure 6, at low mobility (pause time), EEDSR has fairly constant network lifetime values as compared to EADSR has fairly constant network lifetime values as compared to EADSR and DSR. Generally, network lifetime for EEDSR decreases with increase in number of sources and decrease in mobility.

Figure 5. Network lifetime against number of sources at zero pause time (maximum node mobility)

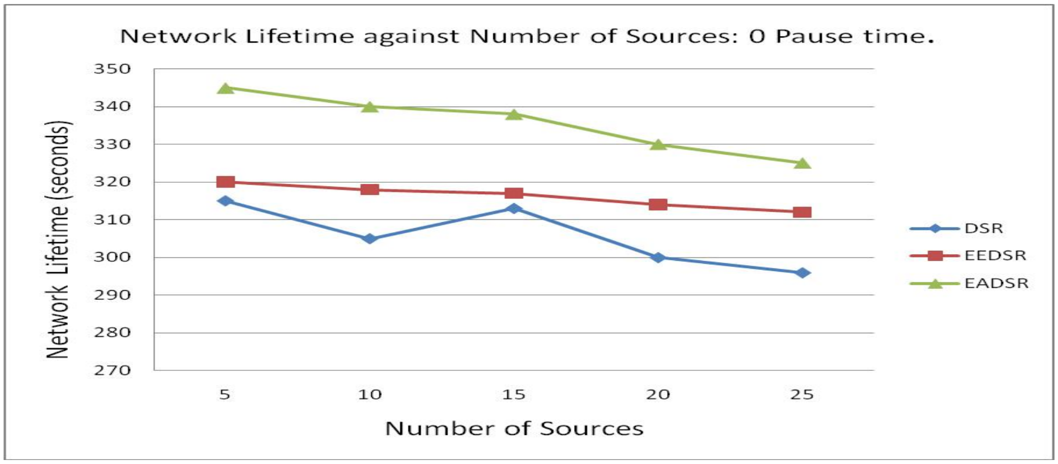
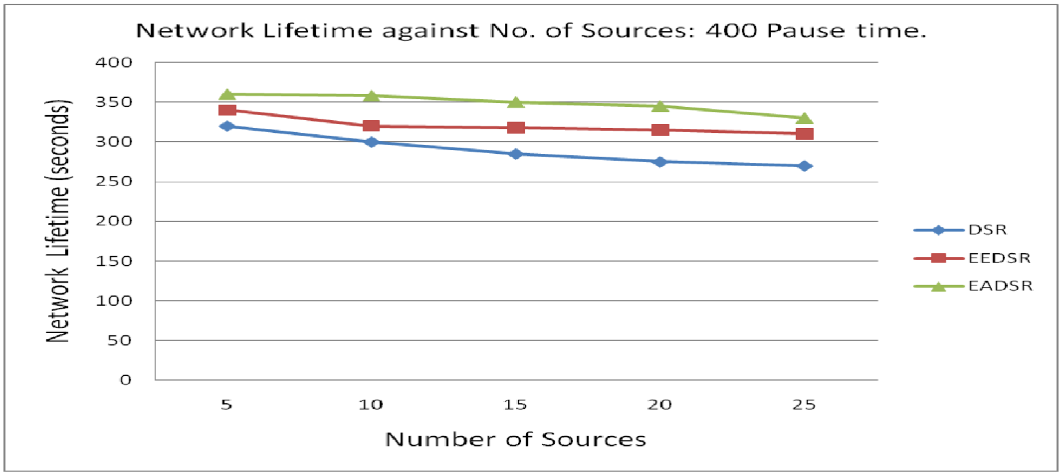


Figure 6. Network lifetime against number of sources at 400 pause times (minimum node mobility).



Limiting routing overhead is one of the ways of saving node energy in Mobile Ad Hoc Networks. In this paper, we showed that EEDSR limits routing overhead of DSR by minimizing the number of generated RREQ packets through periodically flooding them. Including EADSR in the simulations was meant to make performance comparisons between EEDSR and an already existing power-aware routing protocol.

Figures 1 and 2 show that EEDSR on average has lower routing overhead than EADSR and the original DSR especially at low mobility; making it a good energy efficient DSR algorithm at limiting routing overhead.

Figures 3 and 4 show substantially higher throughput for EEDSR compared to DSR especially at average mobility and low mobility. This means that EEDSR is more efficient at successive data delivery compared to DSR.

EEDSR is found to perform poorly at high node mobility as shown by the simulation results at low pause times. Future research would therefore emphasize improving performance at high mobility. More research could also look into a geographically controlled flooding of RREQ packets in the EEDSR algorithm. This would also greatly cut down on routing overhead as the geographical location of the destination node would be the basis for flow of RREQ packets.

ACKNOWLEDGMENTS

We would like to thank Mr. Robert Madeyo of iWayAfrica Uganda for the great help in Linux OS and C++ Programming which helped a lot to the successful completion of this paper.

Many thanks also go to the Networks Department, colleagues at College of Computing and Information Sciences, Makerere University for some of the ideas that generated the paper.

REFERENCES

BHADHARE, S., DOSHI, S., BROWN, X., T., AND SANGHANI, S. 2003. Comparison of Two Wireless Ad Hoc Routing Protocols on a har- ware test-bed. In *IEEE, Wireless Communication and Networking Conference (WCNC).*

CAMERON, B., L. 1998. Routing in Ad Hoc Networks of Mobile Hosts: A Simulation Exercise. *MECH 590: Directed Studies with Dr. Gerard McLean.*

CHEN, J., SHUN-YI, C., YAO, J., N., K. 2004. Empirical Livelihood Confidence Intervals for a Population Containing many Zeros.

DJENOURI, D. AND BADACHE, N. 2006 . New power-aware routing protocol for Mobile Ad Hoc Networks. In *The International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC)*, vol. 1, no. 3, Inderscience Publisher, 126-136 .

DOSHI, S., BHADHARE, S., BROWN, T., X. An On-demand Minimum Energy Routing Protocol for a Wireless Ad Hoc Network. In *Mobile Computing and Communications Review,* vol.6, no.3 .

GIKARU, W. 2004. Mobile-based Routing Overhead Management in Reconfigurable Wireless Ad Hoc Networks, PHD Thesis, Frankfurt Univ- ersity.

JHA, R., K., LIMKAR, S., V., DALAL, U., D. 2010. A Performance Comparison of Routing Protocols (DSR and TORA) for Security Issue In MANET (Mobile Ad Hoc Networks). In *IJCA Special Issue on “Mobile Ad-hoc Networks” MANETs, 2010, pp 78 – 83.*

JIANG, H., AND GARCIA-LUNA-ACEVES, J. 2001. Performance Comparison of Three Routing Protocols for Ad Hoc Networks. IEEE, 547-554.

JOHNSON, B., D., HU, Y., AND MALTZ, D. 2007. The DSR Protocol for Mobile Ad Hoc Networks for IPv4. *IETF MANET Draft.*

MURUGAN, K., BALAJI, P., SIVASANKAR, S., AND SHANMUGAVEL, S. 2005. Cache-based Energy Efficient Strategies in Mobile Ad Hoc Networks. In *Personal Wireless Communications, International Conference on Wearable Communications (ICWC).*

ROYER, M., E., AND TOH, C-K. 1999. A Review of Current Routing for Ad Hoc Mobile Wireless Networks. In *IEEE Personal Communicat-ications, 46-55.*

TAMILARASI, M., CHANDRAMATHI, S., AND PALANIVELU, T., G. 2008. Efficient Energy Management for Mobile Ad Hoc Networks. In *Ubiquitous Computing and Communication Journal, vol. 3, no. 5.12-19.*

TARIQUE, M., HOSSAIN, A., ISLAM, R., AND HOSSAIN, A. 2010. Issues of Long-Hop and Short-Hop Routing in Mobile Ad Hoc Network: A Comprehensive Study. In *International Journal of Network Protocols and Algorithm ISSN*, vol. 2, 1943-3581, (July 2010).

SATHISH, S., THANGAVEL, K., AND BOOPATHI, S. 2011. Comparative Analysis of DSR, FSR and ZRP Routing Protocols in MANETs. In *International Conference on Information and Network Technology IPCSIT vol.4*