# Chapter 3
# Aspects of Power-aware Routing

## 3.1 Introduction

The mobile nodes in ad-hoc network has restricted battery power; so energy is a challenging issue that needs to be addressed. The existing energy efficient routing assumes that the nodes have constant power. The reality being that the nodes are varied and have different energy level and it results into hidden/exposed terminal problem that decreases network lifetime. In this section, components that are key elements to reduce power utilization by the existing routing protocols, alternative of routing protocol and the needed features for a power aware routing protocol are addressed. Power aware protocols address transmit power, energy per package and remaining battery power with the aim of finding out shortest path routing with minimum energy expense. Main sources of energy consumption during packet transmission are transmission, receive, idle and sleep. The major sources of energy wastage in MAC are message overhearing, collision, price of control packet and idle listening.

The energy expended in dispatching a data-packet of dimensions S bytes over a granted connection can be mathematically formulated as

$$E(S) = E_1 S + E_2 \tag{3.1}$$

$$E_1 = (P_t^{packet} + P^{back})/R) \tag{3.2}$$

$$E_2 = ((P_t^{MAC} S^{MAC} + P_t^{packet} S^{header})/R) + E^{back} \tag{3.3}$$

$P_t^{packet}$ is the transmission power of data-packet, $P^{back}$ is the background power of data-packet, *R* is the rate of transmission, $P_t^{MAC}$ is the transmission power of the MAC packets, $S^{MAC}$ is the size of the MAC packets, $S^{header}$ is the size of MAC packet header-trailer and $E^{back}$ is the background energy required to send packets.

To simplify the analysis it is assumed that $P^{back}$, $E^{back}$ are equal to zero.

$$E(S) = (P_t^{packet}/R)S + E^{back} \tag{3.4}$$

This model [4] is only one of several models where $P^{back}$, $E^{back}$, $S^{header}$ and $P_t^{MAC}$ are assumed to be zero. If all are zero, the $S/R$ can be ignored since it becomes a scaling factor that is the same for all links for any given packet and thus does not affect the shortest-path routing metric and it reduces then to a minimum power metric.

In this thesis, only $E^{back}$ is not zero with the result that the link costs depend on the size of the packets. For very short packets the link cost is approximately the constant $E^{back}$ for every link and so routing will be minimum hop. For very large packets, the link-cost is approximately proportional to $P_t^{packet}$ so routing will be minimum power routing. This suggests that energy aware route discovery, energy aware route choice, and energy aware route maintenance behavior will vary depending on packet size.

## 3.2 Factors to Reduce Energy Consumption

There are several aspects that can be exploited to reduce the energy consumption in wireless networks. These are discussed as follows:

### 3.2.1 Required Power and Node Distance

The minimum transmit power $P_t$ needed for transmission with threshold $P_{thr}$, without considering fading can be modeled as

$$P_t(d) = P_{thr} d^n K \tag{3.5}$$

where *d* is the expanse between the two nodes, n is the route loss exponent (usually n takes worth of 4), and K is unchanging. In the case of maximum, the smallest jump routing utilized in typical ad-hoc routing protocols like in DSR, the transmit power is repaired at 100 mW.

Using equation (3.2), (3.3), and (3.4), the smallest transmission energy needed for successful transmission in terms of minimum transmit power $P_t$ with dimensions S can be given as

$$E_t(S, P_t) = K'_1 P_t(S + S^{header}) + K_2 \tag{3.6}$$

where $K'_1$ is the constant that translates the product of transmit power and packet size to energy and $K_2$ is the constant value of the energy overhead that is independent of the packet size. Substituting the value of $P_t$ obtained from equation (3.5) in (3.6), the equation can be written as

$$E_t(S, d) = K''_1(S + S^{header})d^4 + K_2 \tag{3.7}$$

where $K''_1 = KP_rK'_1$. The current versions of the AODV, DSR, and other protocols discuss fixed transmit power for all packets. The corresponding transmission energy is

$$E_{max}(S) = K_3S + K_2 \tag{3.8}$$

where $K_3$ is a constant that translates packet size to energy consumed. The energy saving using transmit power control is given by

$$\frac{E_{max} - E_t(S, d)}{E_{max}(S)} \tag{3.9}$$

### 3.2.2 Multi-hop Routes Save Energy

Let us consider three nodes *a*, *b*, and *c* in a direct line as shown in Figure 3.1 and the smallest jump routing selects the path from node *a* to node *c*. If the route a-b-c is chosen to transmit packets rather than, taking node *b* as relaying node, the total transmit power required can be given as

$$E_{multi}(S, d, d_1) = E_t(S, d_1) + E_t(S, d - d_1) \tag{3.10}$$

where *d* is the distance between node *a* to node *b* and $d_1$ is the distance from node *a* to node *c*. The savings, $E_{sav}(S, d, d_1)$ that can be obtained through multi-hop route can be given as

$$E_{sav}(S, d, d_1) = E_t(S, d) - E_{multi}(S, d, d_1) \tag{3.11}$$

Therefore,

$$E_{sav}(S,d,d_1) = E_t(S,d) - E_t(S,d_1) - E_t(S,d-d_1) \quad (3.12)$$

where *d* is a source path to sense the power alterations in the source route as the data-packet is forwarded to the destination. Each node senses the power at which it obtains the data-packet and compute the new smallest convey power. $P_{new}$ and the new minimum transmit energy, $E_t(S, P_{new})$ is essential for successful packet transmission by the previous node.

The present node tallies the energy value $E_t(S, P_{new})$ which is advertised by source node in energy-aware DSR and in the routing table in AODV.

$$E_t(S, P_{new}) - E_t(S, P_t)| > M_{\triangle} \quad (3.13)$$

where $M_{\triangle} > 0$ is the threshold worth in dB. When the threshold worth is more than 0, the node groups a flag in the routing packet about the changed power cost of the link and composes a new power value that is attached to the routing header. Now one time the place visited gets the facts and figures package, it checks in the data-packet header and sends back gratuitous path reply packet encompassing the path with the new power data. The sender node and bypassing nodes updates their routing table and cached data with the current power data and thus hold pathway of the energy cost changes of the connections of the path.

In case of node mobility, smaller power paths can form after the primary path discovery of a reduced energy path. The path maintenance logic should find out and take benefit of these paths. A node can observe the data-packets not destined for it and convey ACK from the receiver. The node determines the energy cost to the transmitter $E_{toTX}$ from the data-packet power data and the power cost from itself to the receiver node $(E_{toRX})$ from the ACK. $E_{TXmin}$ is given by the equation

$$E_{toRX} + E_{toTX} < E_{TXmin} \quad (3.14)$$

For 802.11b bi-directional links with transmit power control, the $RTS/CTS$ signaling in the four-frame exchange may not work as specified. In this thesis, $P_{thr}$ and the maximum transmit power value are assumed as same for all nodes.

This section elaborated the mechanisms required to modify an existing on demand protocol to make it energy aware. It is noticed that the same features could be used to make both DSR and AODV energy aware. The next section discusses the DSR protocol chosen to implement the energy aware features.

## 3.3 Selection of Routing Protocol

The existing ad-hoc routing protocols do not exploit the factors discussed above to conserve energy. The aim is to develop an energy aware protocol that is an extension of an existing on-demand routing protocol. In Mobile ad-hoc networks, the routing protocols are primarily of two categories: table driven (static) and on-demand (dynamic). Static routing is not energy efficient compared to dynamic routing. Dynamic source routing is an on-demand routing protocol that manages energy effectively. In DSR protocol, the nodes dynamically frame the routes using multi-hop procedures. The packet headers contain an entire ordered list of nodes through which the data-packet would flow. There is reduction in delay overhead and bandwidth in DSR as there is no need for periodic advertisements, instead it relies on MAC layer for informing about link failures. There are two main phases in the DSR algorithm, the route discovery and route maintenance. DSR works fine for networks with about two hundred nodes and for high mobility of nodes. The work in [61] shows that on-demand routing protocols are more efficient in energy consumption as compared to table-based routing protocols. The DSR protocol is selected over the AODV protocol for developing the energy aware on-demand routing protocol.

This protocol uses source routing and route caches. This serves as an advantage over AODV since multiple routes are available to a given destination. Combining source routing with promiscuous listening, DSR has access to a greater amount of information as compared to AODV that has entry per destination stored in its route table and does not allow promiscuous listening. Since DSR does not have any periodic advertisements, it saves bandwidth and reduces power consumption. AODV requires symmetric links between nodes. DSR on the other hand, utilizes asymmetric links when symmetric links are unavailable. Due to these reasons DSR protocol is selected to incorporate energy aware features to develop an energy aware ad-hoc routing protocol.

## 3.4 Essentials of Minimum Energy Routing Protocol

The existing on-demand routing protocols do not support energy parameter as a parameter for routing. Here the essential features of minimum energy routing protocol is discussed.

### 3.4.1 Energy Cost of Route

The existing on-demand routing protocol considers the hop count for finding a route from sender to receiver and not the energy metric. The existing protocol does not have any method to compute and propagate the minimum transmission power to the routes. The energy cost of a route is the sum of packet processing cost, signaling cost and energy needed for data-packet transmission.

### 3.4.2 Transmission Power Control

The transmission power $P_t$ under a given threshold power $P_{Thresh}$ varies with the distance between the sender and receiver node. For saving energy, the minimum energy routing protocols need to transmit data-packets at $P_t$ rather than at fixed transmission power. There is a need for dynamic transmission power control and this can be computed by using equations (3.2) and (3.3). By knowing $E_1, E_2$ and $P_t$ for every link the minimum energy routing protocol can evaluate the link energy cost using equation (3.2). It is essential to incorporate the transmission power metric in the existing protocol to make the protocols energy sensitive.

### 3.4.3 Detection of Minimum Energy Route

In [28], it was shown that minimum energy routes must discover minimum energy routes and those they transform into multi-hop routes. To find a route to the destination, the sender sends RREQ packet and waits for RREP packet. The existing protocol cannot detect minimum energy route that can be illustrated by considering four nodes a, b, c, d in a straight line. Consider a-b-c-d as the minimum cost route. To initiate the route discovery, node a broadcasts RREQ packet that is heard by nodes b and c, and both the nodes broadcast again that is heard by all the nodes. But as node c has broadcasted the same request earlier, it ignores the RREP from b. Node d will send back reply to the requests it hears from node b and c. So the routes discovered by node *a* are a-b-d and a-c-d which are not minimum energy routes.

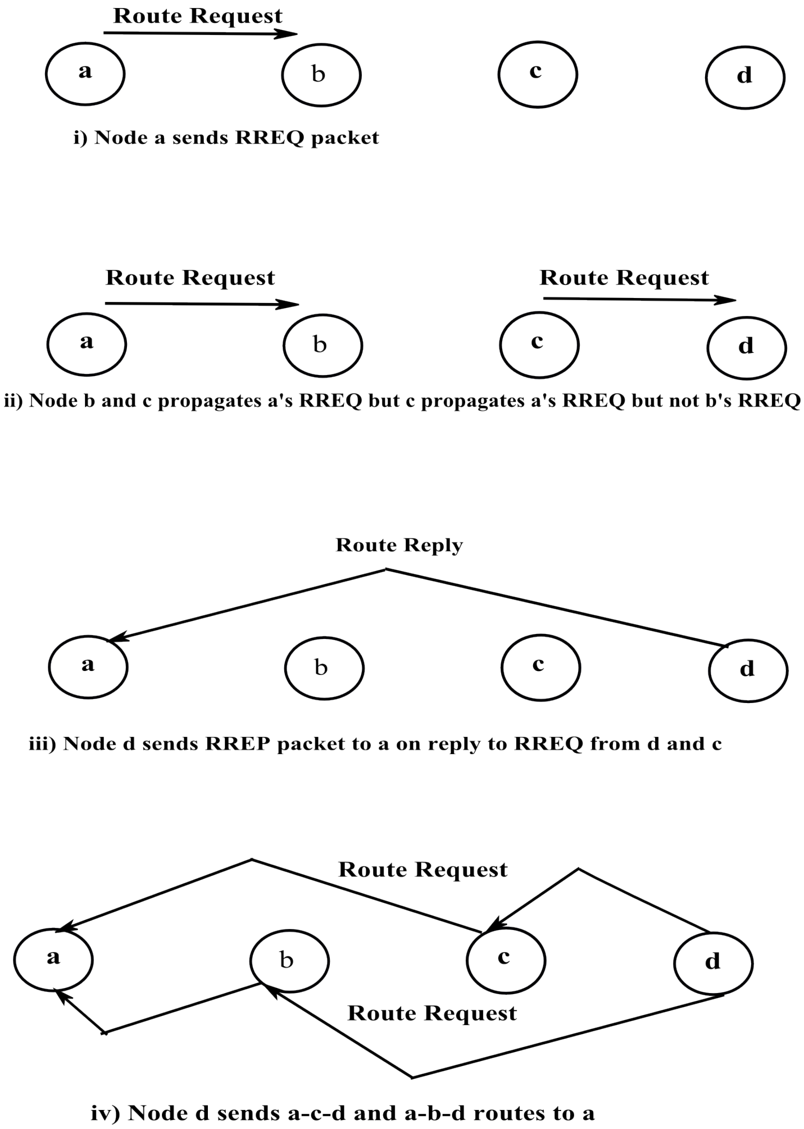



Figure 3.1: Route Discovery in existing DSR

### 3.4.4 Tracking of Minimum Energy Routes

The intermediate nodes must be responsible to correctly forward the packets to the exact destination. Due to node mobility and medium constraints, the nodes may not receive the ACK from link layer and the same packet is resent until the predefined value is reached. After reaching the limits of resending, the link will be treated as broken and it is to be deleted from the cache and generates route error packet (RERR) to be sent to the sender node and all the neighbour nodes. The minimum energy routes must be maintained in regular basis as the nodes are highly mobile in nature. The alterations in the energy cost of the links are to be notified such that the energy expended is close to minimum value. In case, the existing minimum energy route rises due to node mobility, this changes needs to be immediately notified to the sender node to select another minimum energy route. These tracking techniques and connectivity are to be incorporated in minimum energy route, when nodes go apart or come together.

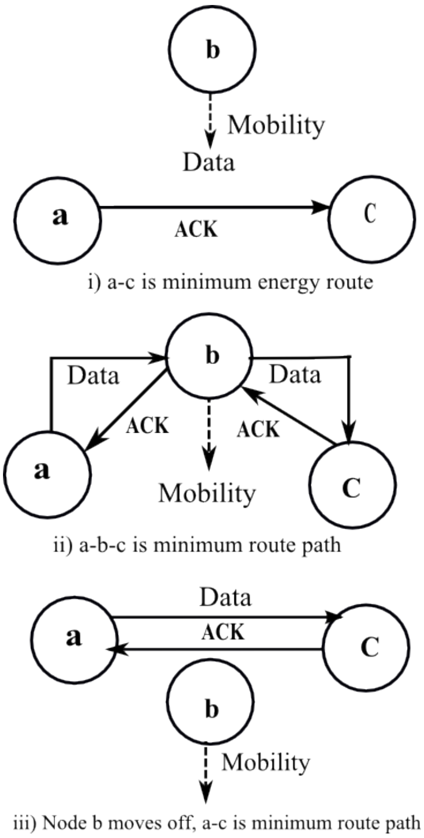



Figure 3.2: Minimum-energy Route Tracking

Consider a scenario with three nodes *a*, *b*, and *c* as shown in Figure 3.2. Initially, a-c link is the minimum energy route. Due to mobility say node *b* moves in between node pair a-c and at this point the optimum energy route is a-b-c. So, route maintenance has to go beyond to achieve basic connectivity.

### 3.4.5 Scalability

The overhead incurred by the minimum energy protocols ought to scale well with the amount of nodes within the network. Minimum energy routes primarily translate to multi-hop routes as the range of nodes within the network increase. The amount of hops within the minimum energy route increases due to the overhead in discovering and maintaining these routes. Existing versions of on-demand ad-hoc routing protocols do not possess most of the desired options of a minimum energy routing protocol as incontestable within the previous section. This section describes mechanisms for the straightforward implementation of those options within the routing logic of the present on-demand protocols.

### 3.4.6 Energy Cost of Link

Given the worth of transmission power $P_t$ in equation (3.5), the worth of $E_1$ and $E_2$ in equation (3.2) and (3.3) can be computed. This power worth $P_t$ is included within the route request packet information and therefore the packet is rebroadcasted by the receiver node. The destination node reverses the route within the route request packet and inserts this power information for every hop within the routing header of the route reply packet to route the route reply packet to the supply node exploitation transmit power management at every hop. Every node within the reply path stores the worth of this minimum transmit power needed to get to subsequent hop, $P_t$. The supply node gets the precise power values for every hop from the route reply from that it will calculate the energy price of the route by using equation (3.5) and an acceptable worth for $E_2$.

### 3.4.7 Broadcast Power Control

This mechanism is enforced by modifying the route request packet header to incorporate the ability at that the packet was transmitted by the supply node. Let $P_{TX}$ be this transmit power in dBm. The receiving node receives this packet at power $P_{RX}$ in dBm. $P_{Thresh}$ is the minimum power level needed for reception in dBW. The minimum power needed for the transmission of the packet in order that it is with success received by the receiver $P_{TXmin}$ and is then calculated in dBW by the receiving node as

$$P_{TX_{min}} = P_{TX} + P_{Thresh}P_{RX} + M \tag{3.15}$$

where *M* is a margin to beat the matter of unstable links owing to channel fluctuations and node quality. The receiver node will browse the worth of $P_{TX}$ from the header of the received packet. For routing the data-packet, just in case of protocols like AODV [62], nodes of the network will merely look up the worth of the minimum transmit power from their routing table and transmit the data-packet at the controlled power. For supply routing on demand protocols like DSR [63], this minimum transmit power price for every hop is enclosed within the routing header of the data-packet by the supply node and every node forwarding the packet merely appearance up successive hop within the supply route, and therefore, the minimum power needed to urge there and transmits the packet at the controlled power level.

### 3.4.8 Minimum Energy Route Discovery

This mechanism permits the node to store the route power information it hears within the route request packets. The node then snoops on route replies not directed to the node and checks if it lies on a lower energy path than publicized within the route reply victimization the hold on route power information. Just in case it lies on a lower energy path, the node sends a gratuitous route reply with the lower energy path to the source. During this manner the nodes will discover the minimum energy route.

## 3.5 Route Maintenance and Link Tracking

This mechanism permits the nodes in a very data-packet's supply route to sense the power changes within the supply route because the data-packet is forwarded to the destination. Every node senses the power on receiving the data-packet and computes the new minimum transmit power $P_{TX_{minnew}}$ and minimum transmit energy $E_{TX_{minnew}}$ needed by the previous node for successful transmission of the packet through equation (3.12), (3.5) and (3.6) respectively. The present node compares this energy worth to the first worth $E_{TX_{minnew}}$ calculated from $P_{TX_{minnew}}$ which is publicized in the supply route just in case of DSR and is gift within the routing table in case of AODV and if

$$E_{TX_{minnew}} - E_{TX_{min}} > 1 \tag{3.16}$$

where $T$ is a threshold worth in decibel, the node sets a flag within the routing header of the data-packet regarding the modified energy price of the link and writes this new power worth within the routing header. Currently once the destination gets the information packet, it checks the flag within the routing header of the data packet and sends a gratuitous route reply packet containing the route with the new power information to the supply. The intermediate nodes and also the supply node update their cache/routing table with this new power data and so keep track of the energy price changes of the links of the route. In case of node mobility, lower energy routes can form after the initial route discovery of a low energy route on which the data-packet flow is being sent. The route maintenance logic ought to discover and take advantage of those routes. This mechanism modifies the MAC ACK header to incorporate the transmit power data of the ACK frame. A node will listen in on a data-packet not directed to itself then on the ACK for the data-packet from the receiver. The node computes the energy price from itself to the transmitter node $E_{TX}$ from the data-packet power data and also the energy price from itself to the receiver node $E_{toRX}$ from the ACK power data. Let $E_{TXmin}$ be the publicized energy price to urge from the transmitter to the receiver node.

$$E_{toRX} + E_{toTX} < E_{TXMIN} \tag{3.17}$$

Now if the node understands that it lies on a lower energy path between the transmitter and also the receiver nodes. It then sends out a gratuitous route reply to the supply of the data-packet informing it regarding the lower energy route (DSR) or to the sending node (AODV). This mechanism allows the nodes to stay track of the optimum minimum energy route over time.

### 3.5.1 Scaling of Minimum Energy Routing Protocol

To ensure scaling, the protocol should converge to the minimum energy route keeping the overhead delimited. This could solely be done if the routing and power information obtained within the route replies is hold on and processed in an intelligent manner in order that each route reply ends up in a clearer topology read of the network at every node. This implies storing the routing and also the power information during a unified graph organization of node's current read of the topology. Hu et al.[64] use an analogous structure, link cache for storing the data obtained from the route replies. The employment of an analogous organization, the energy aware link cache is advised here. The effectiveness of the link cache may be ascertained from the actual fact that the trail cache can come from only one path from node 0 to node 3, whereas the link cache can examine the six attainable ways and select the foremost economical one out of them. The energy aware link cache saves information regarding the energy price of every link $E_1$ and $E_2$ from equation (3.1). Dijkstra's path formula is employed to search out the present minimum energy price path through the graph to the destination node. So every route reply contributes in increase a stronger topology read of the network and a much better energy value graph.

### 3.5.2 Bi-directional Links Computation

The previous section assumes that the links between the nodes are bi-directional. So, $P_{Thr}$ and the transmit power is the same in each directions. This assumption might not continuously be true. Hence, for a link between nodes a and b, $P_{TXmin}$ for the link $a-b$ might not be adequate to $P_{TXmin}$ for the link b-a. Just in case a node c desires to estimate the $P_{TXmin}$ price victimization [65] for $c - b$ once it snoops on packets transmitted by b to a, it cannot do, therefore, while not knowing the $P_{TXmin}$ price of node b. To beat this, the $P_{Thr}$ and transmit power values of the nodes are to be publicized. So, node *c* currently will use node $b's$, $P_{Thr}$ value, and transmit power value to calculate $P_{TXmin}$ for the link $c - b$ [66]. In this implementation, $P_{Thr}$ and maximum transmission power values are same for all nodes.

### 3.5.3 Modification in Protocol Cost

Modifying the present protocol logic so as to implement the minimum energy routing options can incur some price. This section enumerates the prices of every feature in terms of the categories of modifications needed. The values of modifications are often classified as follows:

- Routing software system modification cost: This reflects the value of the changes that have to be compelled to be created within the routing software system of the present protocol so as to support the feature.
- Firmware modification cost: This reflects the value of the changes that have to be compelled to be created within the 802.11 microcode of the present protocol so as support the feature.

- Radio hardware modification cost: This reflects the value of the changes that have to be compelled to be created within the wireless local area network card hardware so as to support the feature.

The link energy price computation feature wants changes solely within the routing header to incorporate the transmit power information of the packet in order that the energy price of the link will be calculated at the receiver node. Therefore, solely the routing software package has to be changed to implement this feature. The transmit power management feature needs dynamically setting the transmit power of the wireless LAN card for every packet to be sent. This can need changes within the radio hardware of the cardboard similarly as routing software package and 802.11 code changes. The minimum energy route discovery would require solely routing software package changes because it solely involves storage and retrieval of the energy price info of the links within the network.

The minimum energy route maintenance and energy price following feature needs transmit power information to be enclosed within the MAC ACK packets that wants 802.11 code modifications besides routing software package changes. Finally, the quantifiable feature can change to be done solely in routing software package to implement the link cache.

### 3.5.4 Proposed Packets in Energy-aware DSR

**Data Packet:** It is a seven field packet format. It contains IP header, fixed header to store the length of DSR header, source route header, intermediate node addresses, source node power values, ACK to next hop, and data.

**Route Request Packet (RREQ):** It has five fields, IP header, fixed header, route request header, list of addresses of intermediate nodes, and power values.

**Route Reply Packet (RREP):** It has eight fields with IP header, fixed header, route reply header, reply path power values, list of reply addresses of intermediate nodes, source route header, list of route reply addresses, and source route power values.

**Ack Packet (ACK):** It has three fields with IP header, fixed header,and Ack headers.

**Route Error Packet (RERR):** It is a seven field format with IP header, fixed header, route error header, unreachable node addresses, source route header, list of source routes, and source route power values.

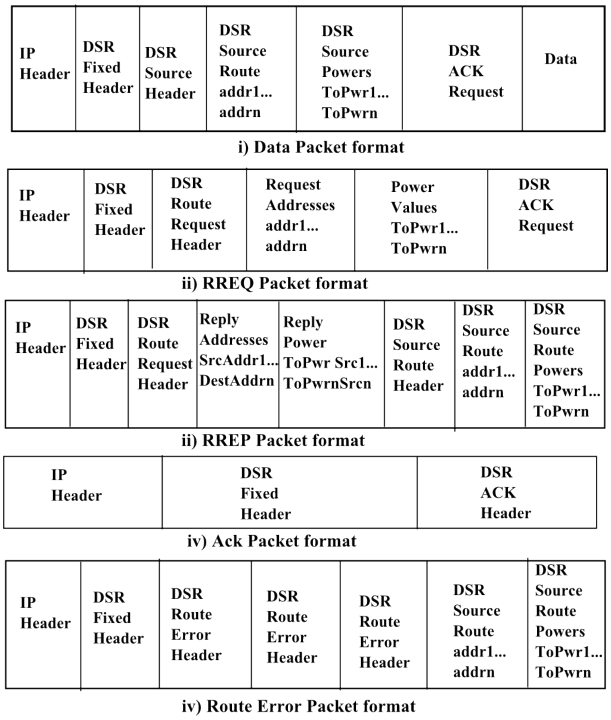



Figure 3.3: The Proposed Packets.

The mechanisms mentioned within the previous section within the DSR protocol logic is enforced in network simulator (NS to get a minimum energy routing version of the DSR protocol. The network simulator version 2.34 [54] is used for simulations. The energy model to be used was the one bundled with the NS-2.34 package with the receiving price of a packet set as zero so as to change the analysis. The two frame exchange theme for the MAC layer information transmission is used. Some alternative changes within the DSR code apart from the mechanisms mentioned are made to get the minimum energy routing protocol version. The changes are:

- **Cache replies off:** The updating of routes is essential in ad-hoc networks due to mobility of nodes. For minimum energy routing the route replies from the buffers are disabled.
- **ACK power control:** To reduce signaling cost per hop, transmission power is controlled in the MAC ACK packets.

DSR is an on-demand reactive routing protocol that has quick adaptation to routing changes and node mobility. There is no periodic router updates or advertisements. It is a strict source routing that specify each node along the path. RREQ and RREP packets gather source routes so that once a route is framed the sender can get the entire route and subsequently place the data-packet for transfer. The route cache in DSR is used to maintain frequently used routes in order to avoid new route discovery mechanism which consumes lot of network resources in the way that each new discovered route is saved in the route cache of the corresponding node for future use, a node can also learns from route request to add new routes to its cache it also learns from route error packets to update its cache.

For selecting optimum energy route, the nodes battery level and currently available energy is to be considered. Energy factor $EnergyFactor$ of a node at an instance is essential that can be determined by dividing initial energy $E_{initial}$ of node by the remaining energy $E_{remain}$. The transmission and receive energy is dependent on the transmission duration of the packet $T_t$.

$$EnergyFactor = \frac{E_{initial}}{E_{remain}} \tag{3.18}$$

$$E_{remain} = E_{idle} + E_{sleep} + E_{transmit} + E_{recv} \tag{3.19}$$

On the premises of energy and stability of node k, the issue of energy with stability is calculated with least energy stability factor $EnergyStabFactor$ of the transmission path.

$$EnergyStabFactor(k) = EnergyFactor(k) + StabFactor(k) \tag{3.20}$$

The path with minimum value of energy factor and stability factor from source *S* to destination *D* in the ith path is selected. In the DSR protocol, the network interface queue of a node is the output queue of packets from the network layer that is waiting to be transmitted by the network interface. The default length of the queue is 50 packets in size. This queue holds the packets while the network interface is transmitting other packets.

Traffic load factor is defined as follows:

$$LoadFactor(i) = \frac{Np(i)}{Nf(i)} \tag{3.21}$$

where $Np(i) = Nf(i) - Nr(i)$
Nr(i) : The remaining network interface queue size of node *i* at an instance
Nf(i) : Initially full interface queue size of node *i*

Np(i) : At an instance, the number of data packets in interface queue of node *i*
LoadFactor(i) : Traffic load factor of node *i*
Now, traffic load factor of the ith route is calculated as follows:

$$LoadFactor_{SD(i)} = \frac{\sum_{k \in N_{SD(i)}} Load_{k(i)}}{N_{SD(i)} + 1} \tag{3.22}$$

$N_{sdi}$ : Set of nodes on ith path from source s to destination d
$N_{sdi} + 1$ : Number of nodes in ith path
The path selection factor is determined as follows:

$$P_{fsdi} = \frac{ES_{fsdi}}{L_{fsdi}} \tag{3.23}$$

The route will be selected with highest $P_{fsdi}$ value for the data transmission.

The total energy of the nodes includes energy in transmission mode, reception mode, overhearing mode and idle mode. Transmission energy $P_{TX}$ is directly proportional to the length of the data packet $D_{len}$ that is on the transmission duration of the packet $T_t$. It can be stated as

$$E_{TX} = D_{len}/2 * 10 \tag{3.24}$$

$$P_{TX} = E_{TX}/T_t \tag{3.25}$$

where $P_{TX}$,$E_{TX}$, and $T_t$ stand for transmission power, transmission energy and elapsed time for sending data of $D_{len}$ size, respectively.

In the receiving mode, reception energy $E_{RX}$ is directly proportional to data packet length $D_{len}$. It is stated as

$$E_{RX} = D_{len}/2 * 10 \tag{3.26}$$

$$P_{RX} = E_{RX}/T_r \tag{3.27}$$

where $P_{RX}$, $E_{RX}$, and $T_r$ stands for reception power, reception energy, and elapsed time for receiving data of $D_{len}$ size, respectively.

In the idle mode, the node is neither sending nor receiving, yet it waits in the receiving mode, so the power consumed in idle mode $P_{IDLE}$ is proportional to $P_{RX}$. Power consumed in idle mode must be reduced.

$$P_{IDLE} = P_{RX} \tag{3.28}$$

In the overhearing mode, power $P_{over}$ is wasted as the neighbouring nodes listen to the RREP packets that is destined for other nodes. It is proportional to reception mode.

$$P_{over} = P_{RX} \tag{3.29}$$

### 3.5.5 Network Model

A network is considered as a directed graph consisting of nodes and links $G = (N, E)$ where $N = 1 - - - N$. For $(i, j)\epsilon N$ and $(i, j)\epsilon E$ where *i* is in the transmission range of *j*. It is assumed that the network is unknown where the nodes are not aware of the number of neighbour nodes form where it can receive message or number of neighbour nodes where it may send message. For any particular link $(i, j)$ between transmitting node I and receiving node j, let $P_{i,j}$ denote the energy to transmit/receive data-packet. The energy dissipated in transmitting $E_{transmit}$ and receiving $E_{receive}$ a data-packet depends on the duration of transmission time that is the length of the data-packet.

## 3.6 Route-Request and Route-Maintenance Algorithm

1. A source node frames a RREQ packet by inserting its ID, DSR fixed header, route request header, energy factor, traffic load factor, and source power levels.

2. On receiving the RREQ packets the neighbour nodes will check in its route-cache by observing the unique sequence number, so as to find a route to the destination to be returned to the sender, else it rebroadcasts the same RREQ to its neighbours after adding its address to the RREQ packet. This information is then added up in its cache.

3. Tasks of the relaying nodes
   3.1 The intermediate node checks the source nodes energy level, source node ID, threshold power, and destination ID. It is saved in the route cache.
   3.2 It appends its energy level and ID, and forwards to the next neighbors.
   3.3 If the RREP is not the first request, then the intermediate node checks for existing route, in case it is present the route with minimum hop and least energy is selected, and RREP is framed and send to the source node. The intermediate node will determine the

power by using

$$P_{new} = P_{TX} - P_{RX} + P_{thr} + P_{over} + P_m \tag{3.30}$$

where $P_m$ is the power essential to overcome the unstable links due to node mobility and channel fluctuations, $P_{thr}$ is the minimum power levels that is essential to receive data-packets. This value is then returned as route reply to the sender node.
3.4 If no route exists, then repeat Steps 3.1, 3.2, and 3.3.

4. Source node will calculate the mean value of all the values of $P_{new}$ of all the nodes and send a RREQ message to the node whose $P_{new}$ value is nearest to the mean value.

5. When the node receives a RREQ message it will send REQ message to its own neighbours and this process will be continued till the destination node reaches.

6. RREP packet reply is the same as in traditional DSR.

***ALGORITHM FOR ROUTE DISCOVERY PROCESS***

**Step 1:** Source node *S* wants to send packet to destination *D*, it checks route cache
If (there is a route to *D*)
(a) Set up the route validation message and transmit through the path available in the route cache and begin the timer
(b) If (ACK reaches before the timer ends)
(c) Transmit the data packets through this path
Else
Set up the RREQ packet by appending the transmission power, source ID to the RREQ packet header and flood to the neighbours.

**Step 2:** If $((powerofnode < P_{thr})$ && $(Neighbour \neq Destination))$
Discard RREQ packet
Else if $((powerofnode \geqslant P_{thr})$ && $(Neighbour = Destination))$
If (it is the first route request)
Determine the path factor $PathFactor_1$ and pile up in the RREQ table and the node will wait for $\Delta t$ time for more RREQ;
Else if ( it is next RREQ && $(time < \Delta t$ time))
Calculate $PathFactor_2, PathFactor_3, PathFactor_4, ..., PathFactor_n$
Else if (additional two RREQ && $(time = \Delta t$ time))
See RREQ table and compare RREQs PathFactor value; destination send RREP packet with

these two RREQ paths which have highest and next highest PathFactor values (means main path and alternate path)
Else if ((less than or equal to two RREQ) && ($time = \Delta t$ time))
Transmit the RREP packet through these two RREQ paths
Else
No need to revise;
**Step 3:** Else $((powerofnode > P_{thr})$ && $(Neighbour \neq Destination))$
The Neighbouring nodes remove the RREQ header and determine the
$EnergyStabFactor = (StabFactor + EnergyFactor)$, LoadFactor, and transmission power;
If $(nodesEnergyStabFactor < headersEnergyStabFactor)$
Replace headers EnergyStabFactor = nodes EnergyStabFactor ; add values of EnergyStabFactor, LoadFactor and transmission power in header field of EnergyStabFactor, LoadFactor, power and broadcast RREQ to their neighbour.
Else
No change in headers value of EnergyStabFactor and replace headers Lf with calculated average value of LoadFactor and broadcast RREQ to their neighbour.
Where
$PathFactor_1$: At a particular instance, the Path factor of first RREQ message.
$PathFactor_2$ : At a particular instance, the Path factor of second RREQ message.
$PathFactor_n$ : At a particular instance, the Path factor of nth RREQ message.
*EnergyStabFactor*: Minimum energy stability factor.
*LoadFactor*: Traffic load factor of ith path from source to destination. Higher load factor means congested route that leads to delays and packet loss.

***ALGORITHM FOR ROUTE MAINTENANCE***

**Step 1:** The source node transmits data to the destination via the main path;
if (there is occurrence of ERR message in any of the node along the route)
An ERR message will be send back to the source node;
**Step 2:** On getting the ERR message the sender node will find out the alternate route;
if ( an alternate route exist in the route cache)
Transmit route validation message through the alternate path and begin the timer;
if (an ACK message reached before timer ends)
The Source node will transmit the remaining data packet through this alternate path;
Else
The Source node will begin new route discovery;

## 3.7 Implementation

In the NS2.34 simulator, QUEUE folder, COMMON, MAC, DSR were changed to implement the algorithm. The DSR Tcl scripts has also been changed for simulating various scenarios by varying the simulation area, number of nodes and transmission/receive power.
The $dsragent.cc$ incorporated the modules for calculating the minimum energy stability factor, traffic load factor, and path selection factors by using the equation number 3.22, 3.23, and 3.24, that was incorporated under handlecost $(SRPacket\&p)$ function. The functions like $HandlePacketReceit()$ and $handleRouteRequest()$ were modified.
The function $handleRouteRequest()$ was changed to receive multiple copies of RREQs and decision to whether reply was done based on the value of Pf. It was set in a way to reply to maximum number of RREQs with main and alternate routes. To discard multiple copies of RREQ the function $ignoreRouteRequestp()$ was used. *replyFromRouteCache* is replaced by *EAreplyFromCache*. This function will unicast the RREQ to the destination rather than to reply the route from cache to initiator. EA-DSR route reply sends to only first two route requests; not to all route requests to reduce the large route replies from the destination. The $dsragent.c$ fie is included in this function with the function name of *rreq-purge*. To tackle power, stability awareness, and load balanced information the source route formats have been modified. Other source files were also modified to suit the experimental objectives of this work. The header file *packet-stamp.h* is one of the most important source files in COMMON folder. It defines the information that is stamped with the packet. The $sender$ node stamped vital information for EA-DSR on the packet. This information includes its nodes stability, residual battery energy and traffic load i.e., queue length. The receiver node has to take out this information from the packet for further processes. The $packet.h$ file defines the packet structure. The packet format contains data energy, and traffic load, i.e., queue length, and headers. Every layer accesses the general header *struct hdr-cmn* on the packet structure. Therefore, this is used to swap over information between the layers. In EA-DSR protocol, the remaining battery energy of receiver node is used for link cost computation on the network layer. The physical layer should send the remaining energy of the node to the network layer using *hdr-cmn*. For simulation the average node speeds are $0.1m/s, 1.0m/s$ and $10m/s$ and the packet rate is $1packet/sec/node$ and the traffic pattern was CBR. The simulation area was changed from 1000m X 1000m to the area given in the simulation parameters.

Table 3.1: Simulation Parameters

| | |
|---|---|
| MAC LAYER TYPE | IEEE 802.11 |
| Reception Queue Length | 50 |
| Round Propagation Model | TwoRayGround |
| Transmission Power (txPower) | 1.4W |
| Reception Power (rxPower) | 1.0W |
| Idle Power | 0.53W |
| Sleep Power | 0.13W |
| Initial Energy | 1000 Jules |
| Transmission Range | 250 Meter |
| Packet Size | 512 Bytes |
| Channel Capacity | 2 Mbps |
| Frequency | 2.4 GHz |
| Transmitted Signal Power | 0.2818 |
| Packet Generation Rate | 2 Packet/Sec<br>4p/s |
| Area Environment Size | 3000m X 3000m |
| Number of Nodes | 10 - 100 |
| Simulation Time | 600 Sec |

## 3.8 Results

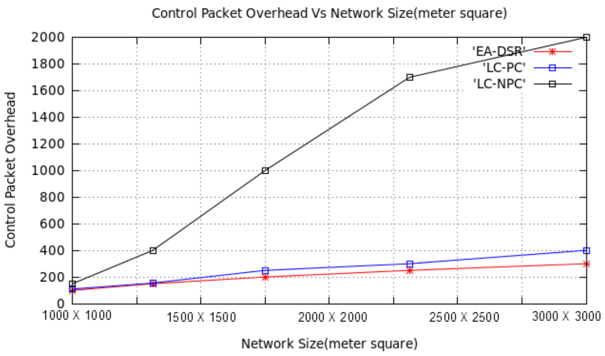
The performance of the energy-aware DSR with path cache was compared to energy-aware DSR with energy aware cache. The results show that energy-aware DSR with energy aware cache performs better in static and slow mobility scenario. The energy consumption of minimum energy routing is reduced due to transmit power control. Three versions of energy savings in DSR were compared

- DSR with link cache having no power control **(LC-NPC)**.
- DSR with link cache having power-aware and power control choices **(LC-PC)**.

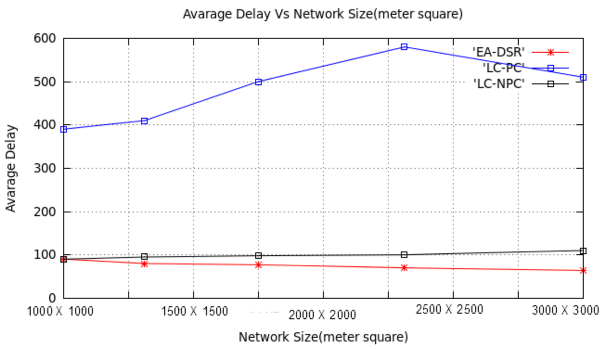
- DSR with energy-aware link cache (**EA-DSR**).

### 3.8.1 Scalability Study

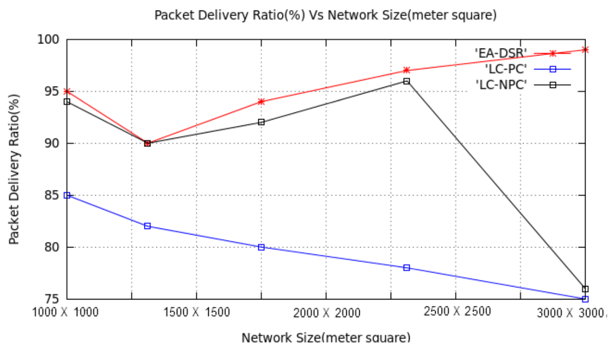
The scalability of **EA-DSR, LC-PC,** and **LC-NPC** was studied by varying network area from 1000 m X 1000 m to 3000 m X 3000 m. The algorithms were carried on the three performance metrics. The proposed algorithm exhibits to be more scalable as the number of nodes than the algorithms having no power control and having power control choices. The overhead of all the algorithms is increased with increasing network area. The control packet overhead is less in the proposed method than the other two algorithms. The end-to-end delay of the proposed method does not show increasing trend than the other two methods since CBR packet type is used. In smaller networks, the average length of routes is shorter but the networks are more congested while in larger networks, the average length of routes is longer but the networks are less congested. The proposed approach achieves a higher packet delivery ratio than the other two methods.




(a)




(b)

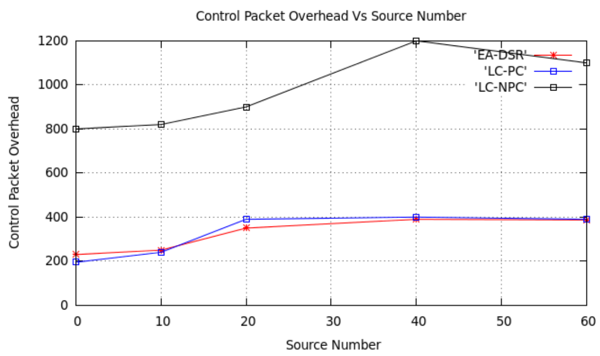



(c)

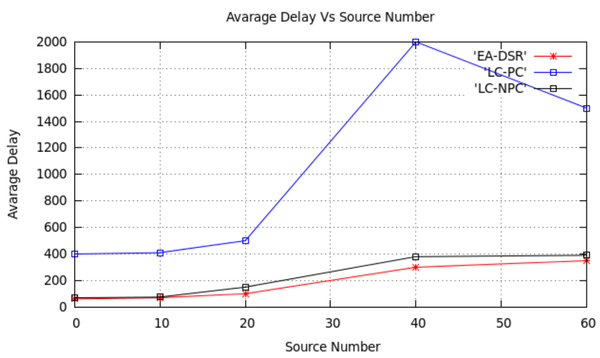
Figure 3.4: Scalability Study

### 3.8.2 Source Nodes

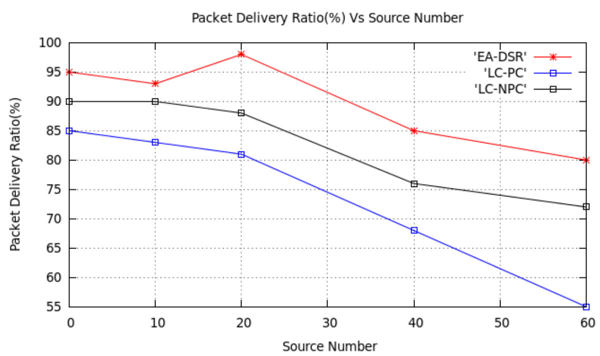
The set of simulations were run on networks with 425 nodes in an area of 2000 m X 2000 m and the node velocity was varied from 10 to 20 m/s and the CBR sources varied from 10 to 60, and node speed was set at 20 m/s. The control overhead of the proposed approach remains almost constant with the increasing number of sources and is not affected by the number of CBR sources. As the number of sources increases, the overhead of other two methods increases. As the number of sources increases, the delays of all the protocol increase because the networks become more congested. Congestion is caused by increasing data packets and routing packets in the other two methods. In our approach, since control overhead does not increase with the increasing CBR sources, the increasing delays are caused simply by increasing data packets. The packet delivery ratio drops as the number of the sources increases because the network becomes more congested. **EA-DSR** has the highest packet delivery ratio. When the number of sources is 60, the packet delivery ratios of proposed approach, **LC-NP**, and **LC-PC** are 69.2%, 63.9%, and 47.2%, respectively.




(a)




(b)

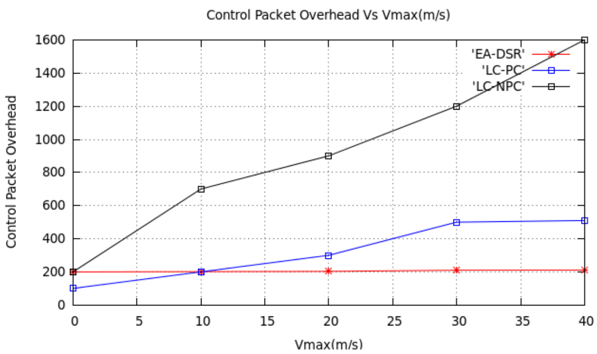



(c)

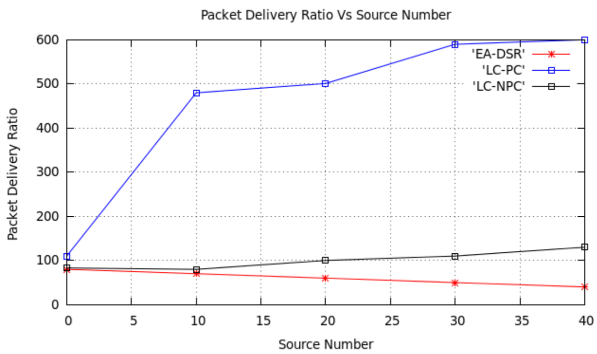
Figure 3.5: Varying Source Nodes

### 3.8.3 Node Speed

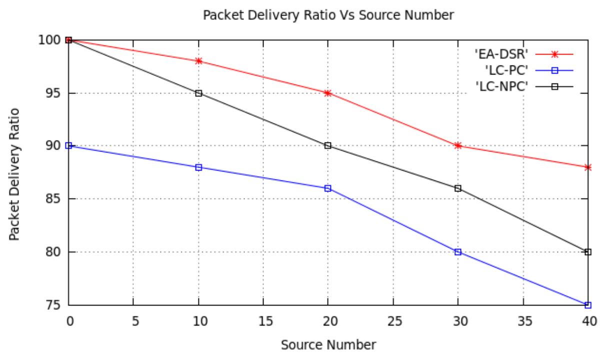
The control overhead of three protocols increases as velocity (Vmax) increases. The control overhead of the proposed approach varies due to the energy updates. The control overhead in **LC-NP** and **LC-PC** increases because of updating in the link cache. The increasing mobility of nodes causes frequent link breakage that result into more housekeeping in the link cache. The average delays for the proposed approach considerably is constant compared to the other two methods. But as the speed of nodes increases the delay is increased in the other two methods due to no power awareness in the algorithms that causes frequent node to die off due to power saturation. The packet delivery ratio for the three decreases as the node speed increases. Besides the congestion caused by increasing overhead, the packet delivery ratios for **EA-DSR** approach and **LC-PC** decrease as the position information becomes less accurate as mobility rises.




(a)




(b)




(c)

Figure 3.6: Varying Node Speed

## 3.9 Conclusion

Energy efficient issues are the important issues to be addressed as the nodes have limited power as the nodes are involved in data communication and data processing. The network gets partitioned whenever a node runs out of energy that leads into complications. Thus, routing in mobile ad-hoc network must use the remaining battery power in an efficient way. The routes with power stability and traffic load is to be used so as to increase the network life time. To bring about the goal in receipt of longer period for a network, energy should be reduced throughout active and inactive states. The approach to minimize the active communication energy is transmission power control and load balancing approaches. The approach to minimize energy during inactive approach is using power-down mode. Main objective of energy-aware routing protocols is to cut down energy consumption while transmission of packets. In this chapter, variable range energy aware dynamic source routing was discussed that is based on stability factors, energy, and load factors. The target is to select the routes with energy efficiency, load factor, and stability. The nodes transmit data-packets with variable transmission power management control such that transmission power is dynamically tuned as per hop-by-hop power control. It is discovered that the use of variable range transmission power control overcomes the disadvantages of common range transmission power management in terms of energy consumption and enhances the network lifetime. The proposed algorithm outperforms the other versions of DSR as the proposed approach considers load factor and power factors before selecting the routes. The energy management can be further improved by using cross-layer approach, wherein the network layers share the information before finding a route.