CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Wireless networks are an emerging technology that will allow users to access information and services electronically, regardless of their geographic location. As the applications using the Internet are increasing and as people are getting used to the advantages of having frequent Internet access, they expect to use network applications even in the situations where the Internet itself is not available. Wireless networks provide solution to such requirements by allowing users with wireless communication devices to access the Internet to meet the communication needs of the moment. Wireless networks [1] can be classified in to two types.

- *Infrastructured networks*
- *Infrastructureless (ad hoc) networks*

1.1 Infrastructured and Ad Hoc Networks

Infrastructured network is a network with fixed and wired gateways. In these networks, communication takes place between mobile nodes and the access point but not directly between two mobile nodes. The access point acts as a bridge to other wireless or wired networks. Several small wireless networks can form large wireless network with the help of access points forming a wired network in between. The mobile unit can move geographically while it is communicating. When it goes out of range of one base station, it connects with new base station and starts communicating through it. This is called *handoff*. As most of the network functionality lies in the access point, the design of an infrastructured network is relatively simple. The structure of the infrastructured network is similar to that of the switched Ethernet in which a central element called Switch controls the flow. The infrastructured networks have got some disadvantages over the ad hoc networks. They do not give the flexibility that should be offered by a wireless network. They cannot be used in places where no infrastructure is available. Examples of such situations are disaster relief, battlefields, and etc.

An *ad hoc network* is a collection of mobile nodes forming a temporary network without the aid of any existing infrastructure or centralized administration. Each node can communicate with other nodes without using any access point to control the medium access. In order to communicate with each other, the nodes have to be in the radio range of each other. In ad hoc networks, the complexity of the wireless node increases. This is because all nodes of these networks behave as routers and take part in discovery and maintenance of routes to other nodes in the network. These networks offer a great flexibility when needed for unexpected meetings and in the communication scenarios far from any infrastructure. Typical applications of mobile ad hoc networks (MANETs) [2] include conferencing, home networking, emergency services, Personal Area Networks, Embedded applications, and PC interaction.

1.2 Energy Efficient MANET

In ad hoc networks, each mobile node acts as both a router and an end node that takes part in route discovery and maintenance. So, the failure of a node can greatly affect the performance of the network. As wireless networking has become an integral component of modern communication infrastructure in recent years for its applications in mobile and personal communications, energy efficiency will be an important design consideration due to the limited battery life of mobile terminals. The essence of using wireless devices is that they can be used anywhere at anytime. One of the greatest limitations to that goal is finite power supply. Since batteries provide limited power, a general constraint of wireless communication is the short lifetime of mobile terminals. Therefore, power management is one of the most challenging problems in wireless communication.

This thesis surveys and analyzes the research done at the routing layer. The energy aware routing protocols can be classified into two types. The goal of the first type of routing protocols is to minimize the total energy consumed when transmitting packet, which *minimizes the energy consumption per packet flow*. However, the routing protocols using these approaches may also use the same paths repeatedly due to their minimal energy consumption, thus reducing the overall network lifetime. This approach is called *transmission power control approach*. The goal of the other type of protocols is to *maximize the system lifetime*, which is the duration from the beginning of the transmission to the first node's energy depletion. This type of protocols put more focus on load balancing among the nodes in the network rather than minimizing the energy conservation for individual packet transmission. This approach is called *load-balancing approach*.

Even though this thesis addresses the incorporation of energy conservation at various layers of the protocol stack, the main goal is to analyze the performance of some existing load balancing routing protocols for ad hoc networks. All the protocols used for this study use energy efficiency schemes at the routing layer on top of the standard *Dynamic Source Routing (DSR)* [21] protocol. [28], [30], and [31] describe the protocols used for this study.

The metrics used to compare the performance of these protocols are end-to-end delay, packet delivery ratio, variance in the remaining battery levels of all the nodes in the network, and the peak-to-mean ratio of energy consumption of all the nodes. Reducing the variance in the battery power levels automatically increases the average network lifetime. The energy efficient protocols showed better performance in terms of variance in the energy consumption and the peak-to-mean ratio, while the performance is worse in terms of the delay and packet delivery ratio.

1.3 Thesis Organization

The thesis is organized as follows. It is divided into five chapters. Chapter 2 surveys the various schemes used to achieve energy efficiency at various layers of the protocol stack. The layers include MAC sublayer, LLC sublayer, transport layer and application layer. Chapter 3 describes the traditional routing protocols for ad hoc networks and the energy efficient routing protocols used for this study as well. Chapter 4 presents the simulation results and gives the discussion of the observations made on the results. Chapter 5 gives the conclusions for this thesis.

CHAPTER II

ENERGY EFFICIENCY SCHEMES FOR WIRELESS NETWORKS

This chapter discusses various techniques used to achieve energy efficiency at different layers of the protocol stack. The main areas of concentration are Medium Access Control (MAC) sublayer, Logical Link Control (LLC) sublayer, transport layer, and application layer. Section 2.1 gives an overview of some existing energy efficiency schemes implemented at the MAC sublayer. Section 2.2 describes some energy efficiency schemes implemented at the LLC sublayer. Section 2.3 presents some energy efficiency schemes implemented at the transport layer. Section 2.4 presents some energy efficiency schemes implemented at the application layer. Energy conservation schemes implemented at the network layer are addressed in the next chapter.

2.1 Medium Access Control (MAC) sublayer

The *medium access control* (MAC) layer is a sublayer of the data link layer and is responsible for controlling the medium access. This sublayer allocates the medium among a number of mobile nodes competing for accessing the medium. This section describes some MAC layer protocols that take energy efficiency into consideration. These protocols include *IEEE 802.11* [1] [3] (Section 2.1.1), *Dynamic Power Saving Mechanism* (*DPSM)* [5] (Section 2.1.2), *Power Aware Multi-Access protocol with Signaling (PAMAS)* [6] (Section 2.1.3), *Geographic Adaptive Fidelity Routing (GAF)* [7] (Section 2.1.4), *Span* [8] (Section 2.1.5), and *Power Aware Routing Optimization (PARO)* [9] (Section 2.1.6).

2.1.1 IEEE 802.11 standard

The *IEEE 802.11* [1] protocol for wireless LANs is a multiple access technique based on *CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)*, and is derived from the MACA protocol described in [4]. This is a random access scheme with carrier sense and collision avoidance through random backoff. This access scheme is called Distributed Coordination Function (DCF). The basic mechanism is shown in Figure 2.1.

When a mobile node has some packets to transmit, it senses the channel first. If the channel is idle for duration of *DIFS (DCF Inter-Frame Spacing)*, the mobile accesses the channel and transmits all pending data packets. Otherwise, the node waits for duration of DIFS before it enters the contention phase. Each mobile node chooses a *random backoff time* within a *contention window* and defers transmission for this random amount of time. The node senses the channel again after that time. If the channel is busy, the node enters the backoff state again until the channel is idle for at least DIFS. Otherwise, the node can get access to the channel.

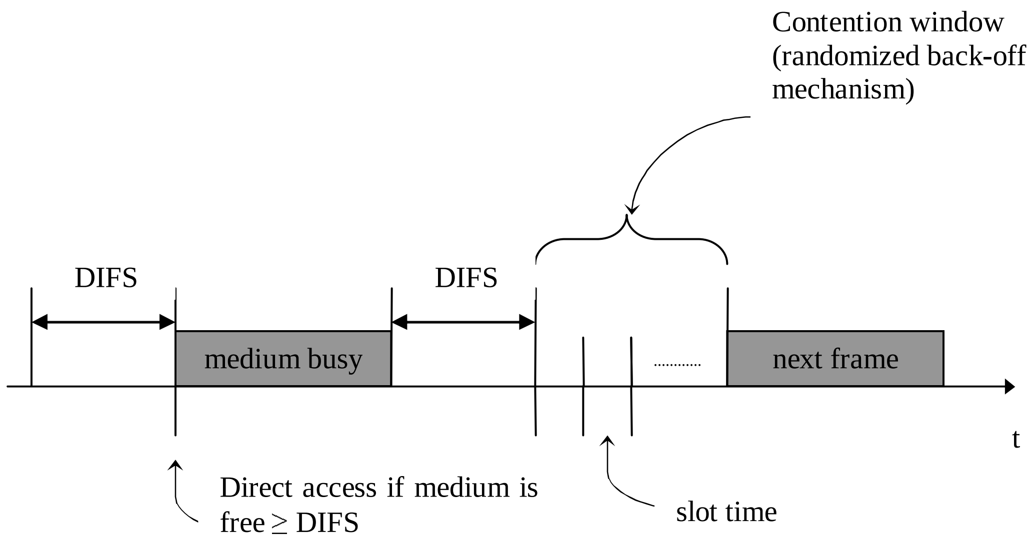

Figure 2.1 Contention window.

For infrastructured networks, the IEEE 802.11 [1] standard uses the following technique for power conservation. The mobile node that does not want to participate in forwarding the packets switches to sleep mode and informs the base station about its decision. The base station buffers all the packets destined to this node. The base station sends periodic beacons to inform the sleeping node about the buffered packets. When the node wakes up, it listens to the beacon and contacts the base station to retrieve the data packets.

For ad hoc networks, the power management scheme described cannot be applied since the ad hoc network does not have an access point. The followed scheme is used for power management in ad hoc networks and is shown in Figure 2.2.

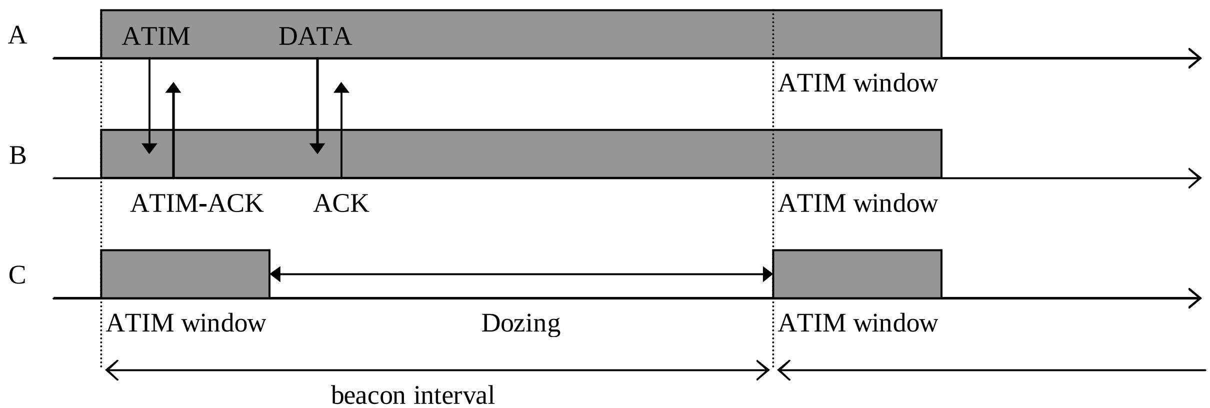

Figure 2.2 Power saving mechanism used in DCF for ad hoc networks.

The nodes that have packets to be received are announced using A*d hoc Traffic Indication Map (ATIM)* and the announcement period is called *ATIM window*. All the nodes stay awake for duration of ATIM window at the beginning of the beacon. If node *A* has a packet to send to node *B*, it sends an *ATIM frame* to node *B*. upon receiving the ATIM frame, node *B* stays awake for the entire duration of the beacon. The node *C*, which did not receive any ATIM frame, goes to sleep and wake up for the next beacon.

2.1.2 Dynamic Power Saving Mechanism (DPSM)

Dynamic Power Saving Mechanism (DPSM) [5] is an optimization to the power saving mechanism used by the DCF in IEEE 802.11. In IEEE 802.11 power save mode (PSM), the duration of ATIM window is fixed. The size of the ATIM window affects the energy consumption greatly. If it is too small, there may not be enough time to announce all the packets. If it were too large, the time for data transmission would be less than that is required. Thus, the dynamic sizing of the ATIM window should be very useful in improving the lifetime of the network and throughput.

Based on the observed network conditions, each node chooses its own ATIM window size. Even the dozing time is longer than that in PSM in IEEE 802.11. The node enters doze state as soon as it completes all the transmissions announced in the ATIM window. The scheme is described in Figure 2.3. If node *A* has a packet to send to node *B*, both nodes enter the doze state after the transmission is completed, if they do not have any other packets to transmit or receive. They do not stay awake for the entire beacon interval as in PSM. When node *A* sends an ATIM frame to node *B*, it includes the number of packets it will be sending to node *B*. Thus, node *B* knows how many packets it should receive.

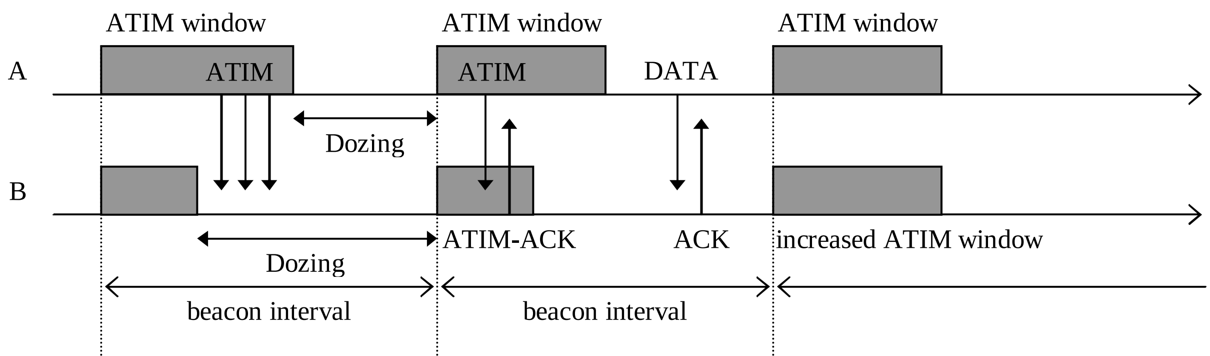

Figure 2.3 DPSM.

In DPSM, a specified set of ATIM window sizes are specified for each node with the minimum window size being equal to *ATIMmin* and each allowed window is called a *level*. The transmission of ATIM frames is done using the CSMA/CA with randomized backoff as specified in IEEE 802.11. The ATIM window size is piggybacked on the packets transmitted by all nodes. Thus, all nodes know the ATIM window size of the other nodes. The buffered packets are sorted by the sizes of the ATIM window of the destination. The destinations with small ATIM window size are preferred. Dynamic adjustment of ATIM window size is done depending upon *(i)* the number of pending packets that could not be announced during the ATIM window *(ii*) the overheard information about the ATIM window sizes of the other nodes *(iii)* receiving the ATIM frame after the ATIM window *(iv)* failure to deliver a packet

2.1.3 PAMAS protocol

The *Power Aware Multi-Access protocol with Signaling (PAMAS)* [6] is an energy efficient media access control protocol for ad hoc networks. This protocol uses a separate *signaling channel* apart from the channel used for data transmission. The RTS/CTS messages are transmitted using this separate channel.

When a node has a packet to transmit, it sends a RTS packet and waits for the CTS packet to be received. If the node gets the CTS packet, it starts transmitting the data packets. Otherwise, it goes into exponential backoff. The receiving node waits for the data packets after transmitting the CTS packet. If the packet does not arrive within a particular amount of time, it goes to the idle state. If the packet starts arriving, the node transmits a busy tone over the signaling channel and receives the packet. When the receiving node hears a transmission of RTS message over the signaling channel, it transmits the busy tone of duration equal to double that of CTS message. Thus, the CTS message the neighbor is expecting will collide with the busy tone and is lost. This is how the receiving node makes sure that no other transmission interrupts its data reception.

Nodes consume power while transmitting data, receiving data and while overhearing other nodes' transmissions. PAMAS achieves the goal of energy efficiency by making the nodes power themselves off whenever they needed to be. The node can power itself off under two situations: (i) if a node has no packets to transmit and one of the neighbors starts transmitting (ii) if one neighbor is transmitting and another one is receiving. The node can know that its neighbor is transmitting by hearing the transmissions over the data channel. It can also know that a neighbor is receiving a data packet by listening to the busy tone transmitted over the signaling channel. Each node decides whether to power itself off or not. But the duration for which the node should be powered off is decided by the *probe protocol* described in [6]. The authors tested PAMAS protocol in a random network topology, a line topology and a fully connected network topology. The protocol seemed to show best result in dense networks and the power savings were low in sparse networks. Also, PAMAS exhibits best performance under light loads and low savings under high loads [6].

2.1.4 Geographic Adaptive Fidelity (GAF)

Geographic Adaptive Fidelity Routing (GAF) [7] keeps a constant level of routing fidelity by turning off unnecessary nodes. GAF implements this policy by using the system and application level information. The node density is also used to adjust routing fidelity. The source and destination nodes are always ON and the intermediate nodes are turned OFF after making sure that the connectivity is still maintained in the network. In GAF, the whole network area is divided into small virtual grids. Each node in the network is associated with a virtual grid. The nodes in the same grid decide on which nodes should sleep and which nodes should be awake. GAF assumes that each node knows its location information relative to other nodes. All the nodes in a grid are supposed to be in range with all the nodes in any adjacent grid. That means all the nodes in a grid are equivalent while forwarding packets.

In GAF, nodes can be in one of the following three states: *sleeping*, *discovery*, and *active*. Initially, the nodes are in *discovery* state. In this state, nodes exchange discovery messages to know the other nodes present in the same grid and the nodes in this state are awake. The discovery message contains node id, grid id, node state, and node active time. The node stays in the discovery state for a duration of T_d seconds and then enters the active state. The nodes in the active state re-broadcast discovery messages every T_d seconds. The node stays in the active state for T_a seconds and then switches to the discovery state. The nodes in the sleeping state switch to discovery state after T_s seconds. The state transitions are shown in Figure 2.4.

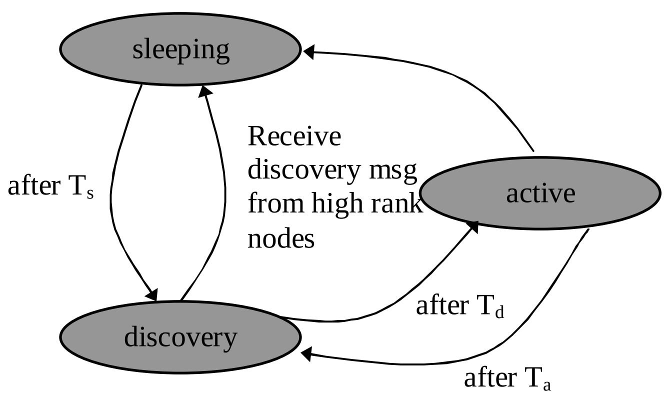

Figure 2.4 State transitions in GAF.

The nodes in active or discovery states switch to sleeping state when they find out that some equivalent node is awake to handle routing. Nodes decide on which node should handle routing depending on the node ranking. To achieve load balancing in GAF, the node are ranked depending on their remaining battery levels. Thus, the nodes with less remaining battery capacity decide to sleep, giving a chance for the other nodes in the grid. The values of the parameters T_a, T_d, and T_s can be chosen depending on the application.

2.1.5 Span

Span [8] is a power saving algorithm in which nodes make local decisions on whether to participate in forwarding packets on behalf of the other nodes. The node makes its decision depending on how many of its neighbors are awake and its remaining battery level. Some of the nodes in the network are elected as coordinators. These nodes stay awake continuously and participate in packet forwarding. The remaining nodes in the network will be in power save mode and periodically wake up to see if they need to become a coordinator.

Span is a proactive protocol and each node periodically broadcasts HELLO messages containing the node's status, its neighbors and the current coordinators. Each node maintains information about the coordinators, the neighbors and the coordinators of neighbors. Span runs on top of link and MAC layers and it also accesses the routing protocol. A non-coordinator node decides if it has to become the coordinator according to the coordinator eligibility rule. The rule is that if two neighbors of a non-coordinator node cannot reach each other, the node should become a coordinator. The algorithm maintains the network so that every populated radio range has a coordinator. If all nodes announce to become a coordinator at once, there is a possibility of contention. To avoid contention, the announcements are delayed with a randomized backoff delay. Each node selects a random value of delay and waits for the duration of that delay. If it does not hear any HELLO messages from the other nodes by the end of that duration, the node sends a HELLO message. A node withdraws as a coordinator if every pair of its neighbors can reach other directly or through some other nodes.

2.1.6 PARO

PARO [9] is a power aware routing optimization that minimizes the transmission power needed to forward packets between nodes in an ad hoc network. PARO is based on the principle that transmission at higher power to a node that is far away is less energy efficient compared to the transmission at lower power to a node that is nearer. In PARO, one or more intermediate nodes act as redirectors between source and destination nodes even if the source and destination nodes are in direct transmission range of each other. This reduces the transmission power necessary to deliver packets between source and destination nodes. This protocol assumes that the radios are capable of adjusting transmission power used for communication.

In PARO, before transmitting a packet a node updates the packet header to indicate the power required to transmit the packet. The other nodes overhearing this node’s transmission use this overheard information and the received power to calculate the minimum transmission power needed to reach this node. In the same way, nodes know the minimum transmission power to reach all nodes. PARO does not maintain routes to other nodes in advance, but the routes are discovered on a per-node basis. In the first iteration, the source node communicates with the destination without the use of any forwarding nodes. The node capable of overhearing both the source and destination computes if the packet forwarding reduces the transmission power compared to direct transmission. If it is so, the intermediate node may become a forwarding node and send a *route-redirect* message to the source and destination about the power efficient route. More redirecting nodes can be added to each route after each iteration of PARO as shown in Figure 2.5.

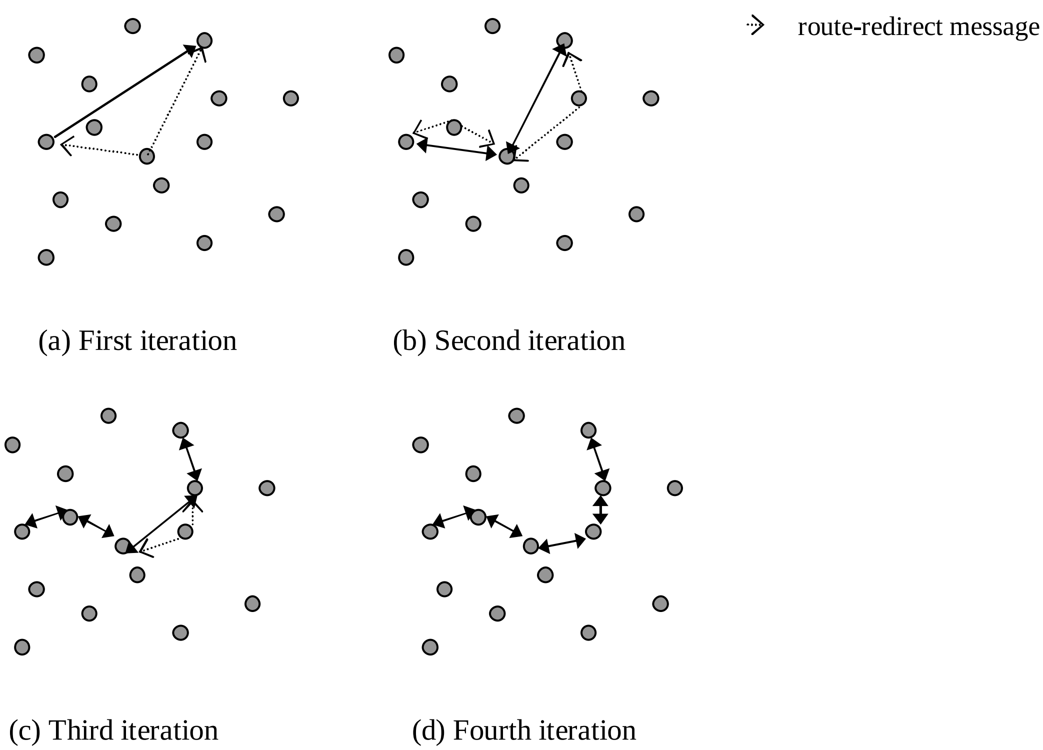

Figure 2.5 Route-redirect mechanism in PARO.

PARO comprises of *overhearing, redirecting,* and *route-maintenance* mechanisms. The overhearing algorithm uses the overheard information to know about the current range of other nodes. Then the overheard packets are passed to the redirecting algorithm to see if the power optimization can be achieved by using the intermediate nodes. If it is the case, the node becomes a redirector and sends *route-redirect* messages to the source and destination nodes and also updates its redirect table. Then the classifier module processes the overheard packet according to the MAC and IP addresses. When a packet is received by PARO, it searches the redirect table to see if a route towards the destination exists. If the node cannot find any route, the packet is transmitted using maximum transmission power. When the destination node replies, then the route optimization is done as described above.

2.2 Logical Link Control (LLC) sublayer

Logical Link Control (LLC) is a sublayer of data link layer and is responsible for framing, flow control, and error control. In this section, we focus on the error control functionality of the LLC sublayer. Common techniques used for error control include Automatic Repeat Request (ARQ) and Forward Error Correction (FEC) schemes. But these schemes try to achieve this by the retransmission of data, which results high power consumption and increases overhead. Section 2.2.1 describes an error control scheme with adaptive ARQ. Section 2.2.2 describes a scheme that uses adaptive FEC/ARQ for error control. Section 2.2.3 gives a discussion on the efficient operating points in terms of source distortion and energy consumption. Section 2.2.4 describes a link layer technique that controls the frame size adaptively.

2.2.1 Adaptive error control with ARQ

An ARQ strategy that includes an adaptive error control protocol is presented and studied in [10, 11]. The authors introduced a new metric in [10] and it is the total number of correctly transmitted packets during the lifetime of an energy source. They suggested an adaptive version of Go-Back-N ARQ protocol, which can be applied to any protocol having a Markov model. The research was done with the assumption that the energy is having a flat power profile. That means the source maintains a constant output power until the battery is dead. The classic ARQ schemes overcome errors by retransmitting the packets that were in error. But as the number of retransmissions increases, the energy expenditure will be more, thus making the transmissions bad even if the channel is in good condition. So, this protocol avoids the persistence in data retransmission.

The energy efficient scheme proposed in [11] is as follows. The transmitter sends the data one block per slot and receives acknowledgements after *m* slots from the beginning of the transmission. It continues the transmissions as long as it receives the acknowledgements properly. Once it notices some errors in the transmissions, the transmitter enters a *probe mode* assuming that the channel condition is bad. In this mode, the transmitter sends a probing packet every *t* seconds. The probing packet is a small packet with a header and a very little payload and consumes small amount of energy. When the receiver sends an acknowledgement for the probing packet saying that the channel condition is improved, the sender enters a *normal mode* and resumes its previous state. The receiver sends an ACK packet whenever it receives a valid packet and a NAK whenever it feels the packet is invalid. When the transmitter receives a NAK, it enters the *probe mode*.

2.2.2 Low Power Error Control with adaptive FEC/ARQ

This subsection describes an energy efficient error control scheme proposed by Lettieri et al. [12] that combines ARQ and FEC strategies. Though the existing FEC and ARQ strategies are very useful in reducing the effect of noise, the schemes do not adapt to the QoS requirements and varying channel conditions. The error control architecture proposed in this section makes each packet stream adapt an error control scheme by itself depending on the QoS requirements. The channel characteristics are observed at runtime and necessary steps will be taken in order to maintain proper energy efficient error control.

This protocol is based on the fact that no single error control scheme can be applied for all traffic types and channel impairments. The error control scheme should be able to adapt to changes in QoS requirements and channel conditions maintaining energy efficiency at the same time. The scheme works as follows. The wireless link interface is modeled such that the packets arrive in multiple streams. Each packet stream is associated with service quality parameters, which are used to select the proper combination of ARQ and FEC schemes. These combinations include Go-Back-N, Cumulative Acknowledgement (CACK), and Selective Acknowledgement (SACK). The basic idea is that a packet is retransmitted if its acknowledgement is not received within a certain amount of time. This duration is calculated based on the number of packets waiting in the transmission pipeline. In order to maintain energy efficiency, the error control scheme may also be adapted at run time if the channel conditions are varying. The hardware can also be made reconfigurable by using FPGAs, in order to support high data rates. The paper also presents the results and analysis under different scenarios.

2.2.3 Energy Efficient Coding and Transmission

This subsection discusses a scheme that provides optimum operating points in terms of the end-to-end source distortion and the energy consumption of the mobile node due to compression, channel coding and transmission. The optimum strategy proposed in [13] is based on the location of the mobile node. In [13], the authors incorporate the energy consumption due to signal processing and due to transmission, into joint source channel encoding problem and provide mathematical tools to calculate efficient operating points. The goal was to allocate the energy between source compression, channel coding and transmission to minimize the total energy dissipation while keeping the source distortion constant.

Two optimization problems were considered: In the first scenario, various compression algorithms were considered, which provide different compression rates for some fixed distortion. In this scenario, the authors tried to optimize the total energy consumption for transmission and scheduling strategies for fixed distortion as the signal travels through the channel. When the mobile is close to the base station, the best strategy is to compress less and focus on the transmission. When the mobile goes away from the base station, emphasis should be on data compression.

In the second scenario, the transmitter is not allowed to choose among different source coders and works with a single compression technique. Now, changing the compression rate makes the distortion to follow the operational rate distortion curve of the encoder. As the channel quality degrades, optimal source coding becomes smaller.

2.2.4 Adaptive Frame Length Control

This technique [14] is based on the dynamic sizing of the MAC frame. It is always advantageous to use large frames to reduce the header and physical layer overhead and to use small frame lengths to reduce frame error rates when channel is noisy. So, the MAC frame length should be changed dynamically according to the channel conditions. This also improves the energy efficiency. In order to achieve this, changes should be made to the data link layer.

The packet length adaptivity is achieved by inserting a fragmentation and reassembly unit, which fragments the IP packet at the transmitter and reassembles them at the receiver. The transmitter adds some overhead to each fragment in order to recognize which packet the fragment belongs to. The size of the fragment is decided according to the channel conditions. Even though the smaller packets are less prone to errors, an ARQ scheme is implemented to make sure the fragments are not in error. When all the fragments are received at the other end, the IP packet is reconstructed and is passed to the higher layers. The channel estimator measures the packet corruption and monitors the feedback from the radio to get the statistics of the channel. The channel estimator can do this by monitoring the packet loss by using the sequence number of the packet. Thus, the BER can be calculated from the packet error rate (PER). This entity decides the packet size in order to achieve the required BER and passes this to the fragmentor at the transmitter.

2.3 Transport Layer

The basic function of the *transport layer* is to accept data from the upper layer, split it into smaller units if needed, pass these to the network layer and ensure that the pieces arrived correctly at the other end. Transmission Control Protocol (TCP) is the most reliable, end-to-end transport protocol designed for wired networks, where packet loss is mainly due to congestion and buffer overflow. So, this protocol might not be optimal for wireless links, where the mobile nodes run with battery power and energy efficiency is a major concern, besides throughput. The congestion control mechanism implemented by TCP uses a backoff algorithm and tries to reduce the size of the flow window, thereby allowing fewer amounts of data to enter the network. This scheme also helps in achieving energy efficiency in case of correlated errors. But the retransmissions increase energy consumption and it is not always a good idea to use this scheme for wireless links. Several modifications were made to TCP in OldTahoe, Tahoe, Reno and NewReno and their energy conservation characteristics are analyzed in [15]. This section briefly describes some techniques implemented at the transport layer to achieve energy efficiency. Section 2.3.1 describes the *Wave & Wait Protocol (WWP)* [16] designed to achieve energy efficiency at the transport layer. Section 2.3.2 gives an overview of an *intelligent suspension/resumption* scheme. Section 2.3.3 describes how energy efficiency can be achieved based on the buffer utilization.

2.3.1 Wave & Wait Protocol

The *Wave & Wait Protocol (WWP)* [16] is designed to achieve energy efficiency by controlling the number of packets transmitted and can handle only the non-real-time applications. Short description of the protocol is given as follows. The protocol transmits 500 bytes assuming that the data of interest to the user falls within that limit. If the user wants more data, he reestablishes the connection for the next 500 bytes. The protocol conserves battery power at the cost of amount of transmitted data.

In WWP [16], the connection is established using six-way handshake. The current network congestion condition is also determined using this six-way handshake as shown in Figure 2.6. The transmitter sends a *wave* consisting of a fixed number of segments to the receiver, and awaits the response. The number of segments in the wave is set depending on the *wave level*, which in turn is decoded by the estimated congestion level. The higher the congestion is, the lower the wave level is.

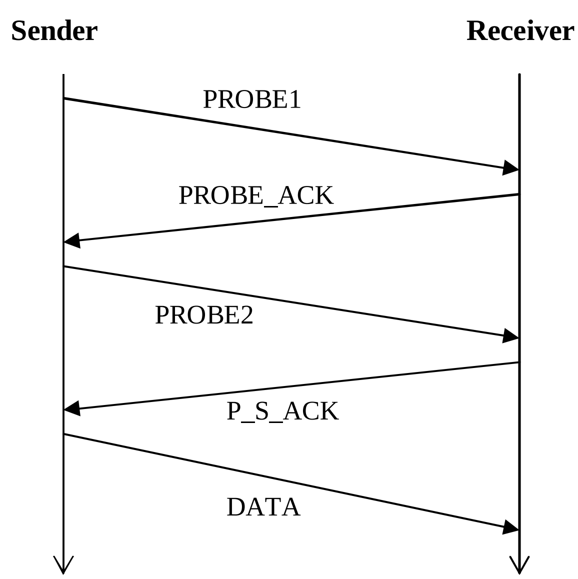

Figure 2.6 Probe cycle.

The receiver estimates the network congestion using the received wave level and sends one negative acknowledgement (N-S_ACK) in response to the entire wave. The N-S_ACK gives an estimate of the amount of data lost by specifying the level of the next wave. The transmitter retransmits those parts of lost data as a part of the next wave. If the N-S_ACK specifies the next level as zero, that means the network is too congested. When the sender receives such N-S_ACK, it waits for some amount of time. Then, it sends a PROBE1 packet and waits for the acknowledgement for the PROBE1 packet. It keeps sending these packets till it receives the acknowledgement. Once it receives the acknowledgement for the PROBE1 packet, it immediately sends the PROBE2 packet and awaits the acknowledgement P-S_ACK for the PROBE2 packet. The P-S_ACK packet also contains the wave level of the next wave the receiver is expecting. When the network condition gets better, transmitter starts transmission again at the wave level specified by the receiver.

2.3.2 Intelligent Suspension/Resumption Scheme

Lilakiatsakun and Seneviratne proposed a scheme [17] that enhances the energy efficiency and performance for TCP by minimizing the probability of packet loss. This scheme suspends transmission when the link characteristics are poor in terms of error probability and resumes transmission when the link condition improves. This scheme needs no modification to TCP, but is designed as an additional function. The implementation of this scheme is as follows. First, the channel condition is determined by measuring the SNR. The channel can be in either *good* state or *bad* state. When the channel is in bad state, the node stops transmitting. The transmission is resumed when the channel condition becomes good. This is implemented by using a thin control layer between the network layer and the transport layer and is shown in Figure 2.7.

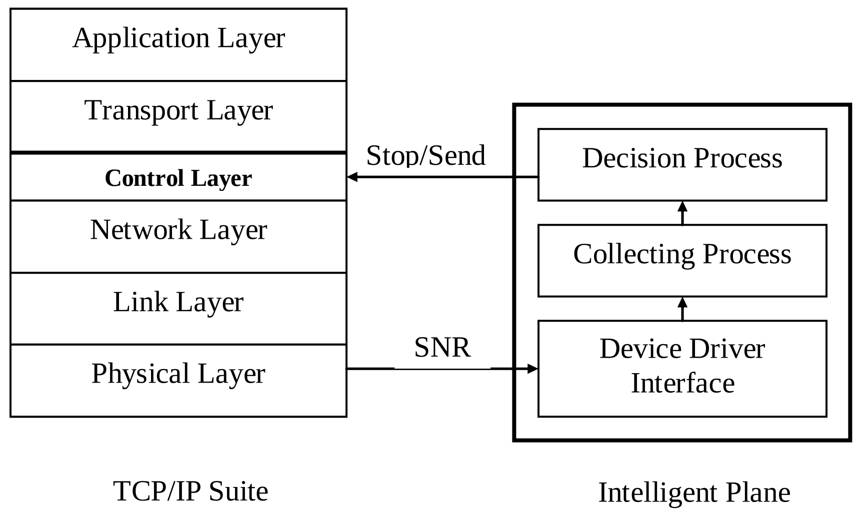

Figure 2.7 The use of control layer and intelligent plane.

An *intelligent plane* does the collection and analysis of SNR measurements. The intelligent plane consists of a collecting process and a decision process. The collecting process gathers the SNR information through a device driver interfacing function. The decision process decides what the scheme should do next. The SNR is compared to a threshold value. If it is stable and higher than the threshold, the channel is meant to be in good state. Then, no signal is sent to TCP. If the SNR changes and is decreasing, the node is assumed to be in motion. If the SNR is less than the threshold, the control layer sends a signal to TCP so that it stops generating any new packets. The signaling is done by using a receiving window of size zero. The control process buffers all the acknowledgements for the previously transmitted packets till the channel condition improves. When the channel condition is good, the control layer sends the buffered acknowledgement packets to the TCP. This indicates that the TCP can start transmission.

2.3.3 Low-Power TCP Buffering

This scheme [18] achieves energy efficiency by reducing the power consumed by the wireless network interface. By monitoring the TCP runtime parameters, the idle periods for the network interface can be decided accurately. Tuning the TCP parameters is also done in order to achieve energy efficiency. This scheme exploits the TCP buffering mechanisms to improve the TCP energy efficiency. By looking at the TCP receive buffer utilization; the network interface can be put in sleep mode during the periods of inactivity. In this scheme, the network interface need not wake up periodically to check for any transmissions addressed to it. TCP experiences inactivity either when the buffer is *full* or when the buffer is *empty*. The scheme is described below.

The scheme is implemented on a Compaq iPAQ PDA with Cisco Aeronet network adapter. The TCP buffer full condition is demonstrated by using an FTP download to the FLASH memory of the PDA and the iPAQ runs Linux. As the operation of writing to FLASH memory is slow, the buffer gets full. When the buffer is full, the receiver advertises a *zero window* and the transmitter do not send any more packets. As the data is read by the application from the buffer, the window gets a non-zero value. According to the normal TCP, the receiver advertises the non-zero window only when the window size is increased by at least a full sized segment. Thus, there is a period of inactivity between the zero and non-zero window advertisements. During this period, the network interface can be switched off. Also, the buffer occupancy threshold when a non-zero window is advertised is a tunable parameter. The card can be in sleep mode until the application completely drains the buffer, thus saving more energy.

Another situation of inactivity occurs when the buffer is empty. The energy efficiency techniques should be implemented carefully in this situation, because some packets might be in transit even if the buffer is empty. If the network interface is shut off by the time the packet reaches the receiver, the packets will be lost. As the lost packets can be recovered by retransmissions, some scheme should be implemented so that the network card is turned back on by the time the retransmissions reached the receiver. One way to achieve this is by predicting the traffic flow from the status of the outstanding sockets. This uses the traffic profiles generated by the applications. A timer is activated whenever the receive buffer is empty. This is checked whenever a transport layer read instruction is completed or when an acknowledgment is received. If the timer expires before receiving any subsequent read or write operations, the network interface is shut off. The card is activated when a new TCP transmission has to be carried out. This policy achieves high-energy efficiency for the applications that generate bursty traffic.

2.4 Application Layer

The *application layer* provides services that are specific to an application. The application layer deals with file transfers, accessing the remote file systems etc. The application directly deals with the application layer. This section describes various schemes used to achieve energy efficiency using the application-based information. Section 2.4.1 describes a scheme that conserves the energy at the cost of application quality and is application specific.

2.4.1 Energy Aware Adaptation for Mobile Applications

This scheme [19] describes how applications can modify their behavior dynamically to conserve energy. Linux operating system provides this adaptation to save battery power at the cost of application quality. To get a battery life of specified duration, the operating system can control adaptation by applications. This is done by predicting the energy demand from the past usage.

The implementation uses the tools PowerScope and Odyssey. PowerScope is a tool that determines the fraction of the total power consumed by different processes. It helps in knowing the components mostly responsible for consuming power. Adaptation in Odyssey is done by sacrificing data quality to power consumption. For example, a system playing a full-color video can switch to black & white video, when the bandwidth drops. Odyssey uses an attribute called *fidelity* to do this adaptation. *Fidelity* is the degree to which the data at the client matches a copy at the server. Each application specifies the levels of *fidelity* it supports.

Odyssey is built into the operating system. A component called *viceroy* monitors the availability of resources and manages their use. There are also code components called *wardens* for each type of application. They encapsulate the type-specific functionality. There is one *warden* for each data type in the system. Odyssey is integrated into Linux as a VFS file system, along with a set of API extensions for expressing resource expectations. If the resource levels are less than the level specified by the application, Odyssey notifies it through an *upcall*. Then, the application adjusts its fidelity level to match the new resource level and sends these new levels to the Odyssey. Energy can be saved significantly by lowering fidelity of the applications.

CHAPTER III

ENERGY EFFICIENT ROUTING PROTOCOLS FOR AD HOC NETWORKS

This chapter presents the research done at the routing layer for ad hoc networks. Section 3.1 gives an overview of the routing protocols for ad hoc networks. Section 3.2 describes different energy efficient schemes used at the routing layer in mobile ad hoc networks.

3.1 Routing Protocols for Ad Hoc Networks

This section gives an introduction to the routing protocols for ad hoc networks. Section 3.1.1 describes the table-driven routing protocols for ad hoc networks. Section 3.1.2 describes the on-demand routing protocols for ad hoc networks.

The rules and conventions used in the conversation between two nodes are collectively known as a *protocol*. It is an agreement between the two communicating parties on how the communication should proceed. Each node in the ad hoc network participates in the formation of network topology. Routing protocols provide information necessary for each node to forward packets to the next hop along the way from the source to the destination. Based on when and how the routes are discovered, these routing protocols for ad hoc networks can be divided into two categories [2]:

- *Table-driven routing protocols*
- *On-demand routing protocols*

3.1.1 Table Driven Routing Protocols

Table-driven routing protocols, also called proactive protocols are the protocols in which the nodes keep track of the routes to all destinations in the network. Each node maintains routing tables to store this information. These tables are updated periodically in order to keep the up-to-date routes to all the destinations. Whenever there is a link breakage due to node movement, the nodes send update messages throughout the network. The advantage of these protocols is that there is a minimal initial delay for applications, because the route can be immediately selected from the routing table. The disadvantage of using these protocols is that additional traffic is needed in order to keep up-to-date routing information. An example protocol of this category is described below.

The *DSDV (Destination-Sequenced Distance-Vector Routing)* [20] is a Table-Driven Routing protocol. Each node in DSDV maintains a routing table containing the "next hop" information to all the reachable destinations in the network. The sequence number distinguishes the routes. The route with the highest sequence number is always preferred, if multiple routes are known to a destination. Whenever an existing link to a destination is broken, the node needing a new route advertises a route with infinite metric and increased sequence number. All nodes hearing this advertisement update their routing tables to incorporate the infinite metric to that particular destination until they hear a newly established route.

3.1.2 On-Demand Routing Protocols

On-demand routing protocols are the protocols in which the routes are acquired as and when they are needed. Whenever a source node has a packet to send to a destination, it initiates a route discovery to find the path to the destination. These protocols use less bandwidth compared to the proactive protocols, but the latency for application will increase. The route remains valid till the destination is reachable or until the route is no longer needed. This section discusses a few existing on-demand routing protocols.

Typical examples for on-demand routing protocols are AODV, TORA, and DSR and are explained in this section. As DSR is one of the protocols implemented in this thesis, it is explained in detail while the remaining protocols are explained briefly.

a) Dynamic Source Routing Protocol (DSR)

Dynamic Source Routing Protocol (DSR) [21] uses *dynamic source routing* to route packets in an ad hoc network. Source routing is a technique in which the source node determines the entire sequence of nodes a packet has to pass through, when it is being transmitted from source to destination. The source node puts the list of addresses of all nodes in the header of the packet, so that the packet is forwarded to the destination through those specified nodes. However source routing can be done statically or dynamically, *Dynamic Source Routing (DSR)* does it dynamically. This is done using a procedure called *route discovery*. Whenever a node has packet to send to some other node, the first node initiates the route discovery. Each node maintains a cache called *route cache* to store the routes it has gathered to different destinations. As the nodes in an ad hoc network move from place to another, some of the existing links break and the routes in the route caches of the nodes must be modified. This is done using a procedure called *route maintenance.* Route discovery and route maintenance are two main mechanisms used by DSR.

Route Discovery:

Whenever a node has a packet to send to another node, it checks its *route cache* first. If there is no existing route for the required destination node in the route cache, the source node broadcasts a *route request* with a *sequence number* and *destination address*. The *sequence number* identifies a particular route request. Each node maintains a table called *request table*, which maintains the information about all the route requests the node has received before. The source node makes a note of the route request is has sent in the request table.

When a node receives the route request, it checks its request table. If the request table already has an entry exactly the same as the current route request, the node simply ignores the route request. This prevents the route request packet from looping. If the entry is not present in the request table and the node is the destination of the route request, it sends a route reply back to the source node by simply copying the path from the route request packet. If the node is an intermediate node, it checks its route cache. If it has a route to the destination in its cache, the node sends a route reply to the source node by copying the path from its cache and the path from the route request packet. If the intermediate node does not have any available route to the destination in its route cache, it rebroadcasts the route request packet.

After the receiving the first route reply, the source node deletes the corresponding entry in the request table. Then it sends all the data packets destined to the destined node using the path retrieved from the route reply. If some intermediate node receives this route reply on the way to the source node, it saves the route in its route cache.

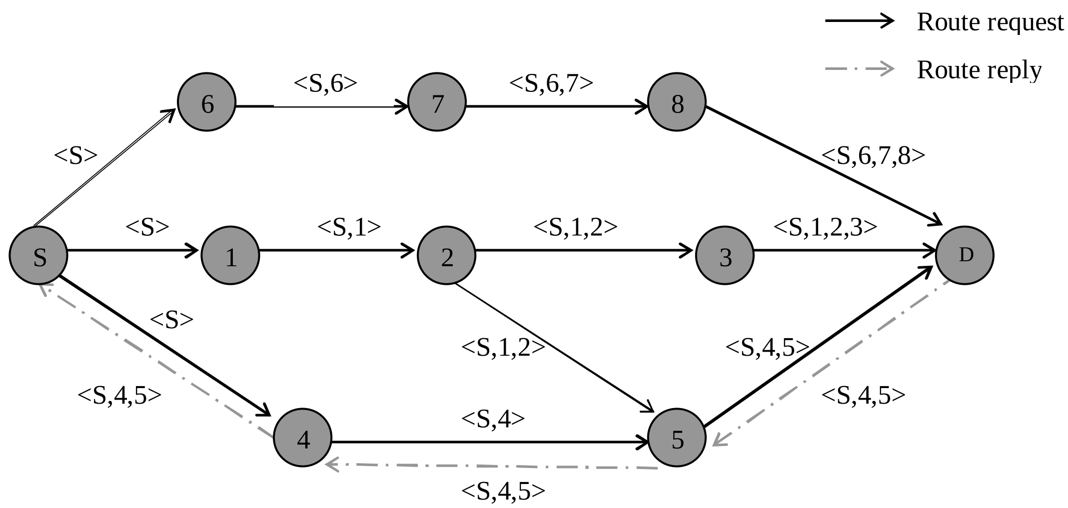

Figure 3.1 Route Discovery mechanism in DSR.

Figure 3.1 shows the propagation of route request message and the corresponding route reply message. Node ‘S’ is the source node and ‘D’ is the destination node. When ‘S’ broadcasts the route request packet, nodes 1,4, and 6 receive it. When node 2 rebroadcasts it, nodes 3, 5 receive it. But the node 5 has already received the route request from node 4. So, it drops the duplicate request received from node 2. Finally, the destination node receives the route request through <S, 4,5> first. So, it sends the route reply using that path.

Route Maintenance:

This procedure uses a new control message called *route error*, which will be used for maintaining the routes. Route maintenance is done with the help of the hop-by-hop acknowledgements used by the data link level or the end-to-end acknowledgements used by the transport or application layer. Whenever a node sends a data packet to the next hop, it waits for the acknowledgement from the receiving node. For some reason, if the node detects that the link to the next hop is in error, it sends a route error packet containing the two node addresses which from the link in error to the original sender of the data packet. When the source node receives the route error packet, it removes from the route cache all the routes containing the error link as a part of the route. If an intermediate node receives the route error packet, it also removes the routes containing that link from its cache and forwards the route error packet to the next hop node.

Promiscuous Mode Operation:

The nodes in an ad hoc network can operate in promiscuous mode, in which the wireless interface of the node will be able to listen to the packets destined to some neighboring nodes also. If a node overhears a route reply packet, it saves the route from the reply in its route cache, if it is not already present in the route cache. If it listens to a route error message, the node deletes all the routes from its route cache containing the error link. If the node overhears a data packet and the node has a route in its cache that is shorter than the source route contained in the packet, it sends a *gratuitous route reply* to the source of the packet. A gratuitous route reply is nothing but a route reply message, except that it is not sent in reply to a route request. The node includes the shorter path it has in the route reply message.

b) Ad hoc On-Demand Distance Vector Routing Protocol (AODV)

Ad hoc On-Demand Distance Vector Routing (AODV) [22] is the combination of DSR and DSDV. It follows DSR in route discovery and maintenance and follows DSDV in routing updates. In the case of link breakage, the information is sent to a neighbor, which recently has sent packets to some destination using that link. Route maintenance is done using *Hello* messages. Each node periodically sends *Hello* message. If the *Hello* messages are not received from a specific node is not received, then the link is suspected to be in error.

c) Temporally Ordered Routing Protocol (TORA)

Temporally Ordered Routing (TORA) [23] protocol is an on-demand routing protocol in which the nodes keep multiple routes to a destination. A node needing a route to a destination broadcasts a QUERY packet. The node receiving this packet broadcasts an UPDATE packet specifying its height. The nodes receiving this UPDATE packet set their height so that it is one greater than the height of the neighbor from which UPDATE packet was received. Thus, a directed acyclic graph is created. If the source node receives more than one UPDATE packet, it considers the packet received from the node with the lowest height. When a node detects a link breakage, it propagates an UPDATE packet with the height set to the local maximum. If it knows that there is some node with a height greater than this height, that link is updated. Otherwise, a new route is discovered as described above. When a node detects a network partition, it sends a CLEAR packet that refreshes the network status and removes all the invalid routes.

3.2 Energy Efficiency at the Network Layer

The *network layer* is responsible for routing packets, establishing the network service type (connection-less versus connection-oriented), and transferring packets between the transport and link layers. In a mobile environment, this layer has the added responsibility of congestion avoidance, rerouting packets and mobility management. This is done with the help of routing protocols. This section presents a discussion on the energy efficient routing algorithms developed for wireless ad hoc networks.

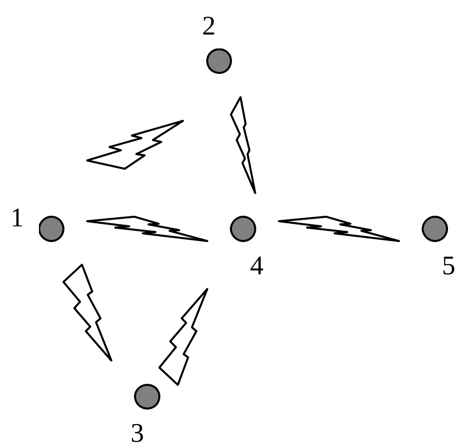

Figure 3.2 An example network topology.

Typical metrics used to evaluate ad hoc networks include shortest-hop, shortest-delay, location stability and link quality. These metrics may have a negative effect in wireless networks because they result in the over usage of a set of nodes, thus reducing network lifetime. For example, in the network shown in Figure 3.2, all the routes to the destination node 5 include node 4. So, the node gets exhausted quickly, thus reducing the overall network lifetime. Because of node mobility, the problem of routing becomes difficult in ad hoc networks. To optimize routes, frequent topology updates are required, while on the other hand, frequent topology updates result in higher message overhead. This section is organized as follows. Section 3.2.1 describes the metrics used to achieve energy efficiency. Section 3.2.2 presents a classification of energy efficient routing protocols and describes protocols that implement energy efficiency schemes at the routing layer.

3.2.1 Metrics for Power Aware Routing

In [24], routing of traffic is addressed with respect to energy consumption. The authors proposed five new metrics that result in energy efficient routes.

1. *Minimize energy consumed per packet*: This metric minimizes the average energy consumption of the nodes in the network. If the network is lightly loaded, this metric selects the path that is most probably the same as the path selected by the shortest path routing. Under heavy loads, this metric tries to avoid the congested areas while routing packets. This might increase the hop count sometimes. The disadvantage of using this metric for routing is that the variance in energy levels will be high, thus resulting in the death of some nodes.
2. *Maximize time to network partition*: For a given network topology, there exist a set of nodes whose removal causes the network to partition. All the routes between the two partitions go through these nodes. Thus, if the load is not divided properly between those critical nodes, there is possibility that some of these die

causing the network to partition. This reduces the overall network performance. This metric is of importance in battlefield networks.

3. *Minimize variance in node power levels:* This metric ensures that all the nodes in the network are equally used and none of the nodes is overused. It tries to run the nodes together as long as possible. The routing protocol using metric should be such that the nodes send packets to the neighbors with least numbers of packets waiting to be transmitted.
4. *Minimize cost per packet:* This metric increases the life of all the nodes in the network. The paths should try to avoid nodes with low energy levels. Such nodes should not lie on many paths. The advantage of this metric is that the battery characteristics can be incorporated into the routing protocol and it also reduces the variation in node costs.
5. *Minimize maximum node cost:* This metric minimizes the cost of a node when routing packets through it. This also minimizes the maximum node cost. Also, it reduces the variance in the node power levels, thus increasing the overall lifetime of the network.

3.2.2 Classification of Energy Efficient Routing Protocols

The energy efficient routing protocols can be classified into three types according to Yu et al. [25]. The classification is based on the energy consumption, which can be either *the active communication energy* required to transmit and receive packets or the *inactive energy* consumed when a mobile node stays idle but listens to the medium. *Transmission* *power control approach* and *load distribution approach* belong to the former category while the *sleep/power down mode approach* belongs to the latter category.

a) Transmission Power Control Approach

In the *transmission power control approach*, the active communication energy can be reduced by adjusting the node's power just enough to reach the receiving node, but not more than that. The protocols using this approach determine the optimal routing path that minimizes the total transmission energy required to deliver data packets to the destination. If the transmission power is controllable, it is advantageous to have the packet travel through more number of nodes, because the required power, $p(d) \propto d^2$, where '*d*' is the distance traveled. The protocols that use this approach include *Flow Augmentation Routing (FAR)* [26], *Online Min-Max Routing (OMM)* [27], *Maximum Transmission Power Routing (MTPR)* [28], *Distributed Power Control (DPC)* [29]. Table 3.1 summarizes the protocols belonging to this category.

Table 3.1 Energy efficient routing protocols that use transmission power control approach.

Protocol	***Summary***
FAR [26]	Uses the shortest cost path to the destination, includes node cost in the link cost
OMM [27]	Selects a path that maximizes the minimal residual power
MTPR [28]	Chooses a route with minimum total transmission power
DPC [29]	Minimize the overall spent in the end-to-end transmission by using appropriate power level to transmit the packet

i) Flow Augmentation Routing (FAR)

Flow Augmentation Routing (FAR) [26] protocol can be applied to either static networks or to the networks whose topology changes slowly. In FAR, at each iteration, each origin node $o \in O(c)$ of commodity c calculates the shortest path to each of its destination nodes in $D(c)$. Then the flow is augmented by an amount equal to λ times the actual amount. After the flow augmentation, the shortest cost paths are recalculated and the procedure is repeated until one of the nodes runs out of its initial battery energy. By doing this, we can obtain the flow which will be used at each node to properly split the incoming traffic.

ii) Online Max-Min Routing Protocol (OMM)

Online Max-Min Routing Protocol (OMM) [27] is applicable to the sparsely deployed networks. This protocol focuses on maximizing the lifetime of the network. The lifetime of the network can be taken as the time to the expiration of the first node. This protocol tries to optimize power consumption by selecting the path that minimum power consumption and the path that maximizes the minimal residual power of the nodes in the network. The algorithm tries to relax the minimal power consumption for the message to be $z\ P_{min}$ with parameter $z \geq 1$ to restrict the power consumption for sending one message to $z\ P_{min}$. This algorithm consumes at most $z\ P_{min}$ while maximizing the minimal residual power fraction.

iii) Maximum Transmission Power Routing (MTPR)

Maximum Transmission Power Routing (MTPR) [28] protocol minimizes the total transmission power consumed per packet and is not concerned with the remaining battery power of the nodes. The transmission power needed to send a packet from source to destination is directly proportional to the distance between the source and the destination. So, transmitting a packet to a node that is farther is consumes more energy compared to the transmission to a node that is nearer. Thus, in order to minimize the transmission power this protocol prefers routes with more number of hops than those with less number of hops. This increases the end-to-end delay in transmitting packet, but reduces the minimum transmission power to send the packet.

iv) Distributed Power Control (DPC)

In *Distributed Power Control (DPC)* [29], the sender uses an appropriate power level to transmit its packets by reducing channel interference and energy consumption, and the final path is selected by minimizing the overall power spent in the end-to-end transmission. This is primarily based on selecting a suitable transmit power level, which also reduces the interference. This transmit power is used as the link cost function in the path discovery and selection. This method assumes that each node can record the power level to transmit the packet (P_T) and the received power (P_R). A good choice for the transmit power P'_T would be

$$P'_T = P_T - P_R + S + Sec_{th}$$

Where S is the minimum power level required for a correct packet reception and Sec_{th} (Security Threshold) is a power margin introduced to take into account channel and interference power level fluctuations. The P'_T values are taken as the cost functions used by the routing algorithm to select a path. Thus, the algorithm tries to implement power saving mechanism.

b) Load Distribution Approach

The protocols using the *load distribution approach* try to balance the energy consumption in the network by using less frequently used nodes, instead of shortest paths. The route selected need not be the lowest energy route, but it ensures that the variance among the node energy levels is reduced. Examples include *Local Energy Aware Routing (LEAR)* [30], *Min-Max Battery Capacity Routing (MMBCR)* [28], *Conditional Max-Min Battery Capacity Routing (CMMBCR)* [28], *Request Delay Routing Protocol (RDRP)* [31], *Energy Aware AODV* [32], *Location-aided Power-aware Routing Protocol (LAPAR)* [33]. As the protocols implemented in this thesis are *Min-Max Battery Cost Routing (MMBCR)* [28], *Local Energy Aware Routing (LEAR)* [30], *Request Delay Routing Protocol (RDRP)* [31], they are explained in detail while the remaining protocols are explained briefly. Table 3.2 summarizes these protocols.

Table 3.2 Summary of energy efficient routing protocols using load-balancing approach.

Protocol	***Summary***
LEAR [30]	Route discovery procedure is modified to achieve energy balance in the network
MMBCR [28]	Route reply is handled in a different way
CMMBCR [28]	Combines MTPR & MMBCR
RDRP [31]	Route discovery procedure is modified
Energy Aware AODV [32]	Packets are routed around nodes with low power budgets
LAPAR [33]	Packets are forwarded based on the physical location of the destination

i) Min-Max Battery Cost Routing Protocol (MMBCR)

Min-Max Battery Cost Routing Protocol (MMBCR) [28] is similar to DSR except for a few modifications made in order to achieve energy efficiency. MMBCR tries to optimize the overall lifetime of the network, instead of the end-to-end delay. It takes into consideration the remaining battery capacity of the nodes in selecting the path to send packets from a source to a destination.

The MMBCR protocol works as follows. It is the same as DSR, except that the route reply is processed is differently. In DSR, the source node uses the first route reply to send packets to the destination node. It simply uses the shortest path. DSR never considers the energy levels of the nodes. But in MMBCR, the source node does not use the route reply immediately after its arrival. It waits for some time for a better route and then selects a path among all the existing paths, so that the variance of the energy levels of all the nodes in that path is the least and also all the nodes in that possess better energy level. A node is said to be *rich* if its remaining battery energy is greater than a *threshold* value. Otherwise, the node is considered *poor.* The protocol is implemented as described below.

<u>Protocol Implementation:</u>

Whenever a node has a packet to send to another node, it initiates the route discovery. The route discovery procedure is exactly the same as the route discovery mechanism used in DSR.

When a node receives a route reply and if the node is the destination of the route reply, it checks the route cache for any duplicate routes present for the same destination node. If there exists exactly the same route as the one contained in the route reply, the route reply is simply ignored. Otherwise, it inserts the route contained in the reply into the route cache. If all the nodes contained in the path are rich, it sends all the buffered packets for the destination using that route. If some of the nodes in the path are not rich, the node waits for duration of *DSR_MAX_WAITING_TIME* to receive all possible replies. After waiting for *DSR_MAX_WAITING_TIME*, the node selects the path whose critical node possesses the highest remaining battery level among the critical nodes of all the paths and sends all the buffered packets for the destination node using that path. If some intermediate node receives the reply, the node simply stores the route into its route cache.

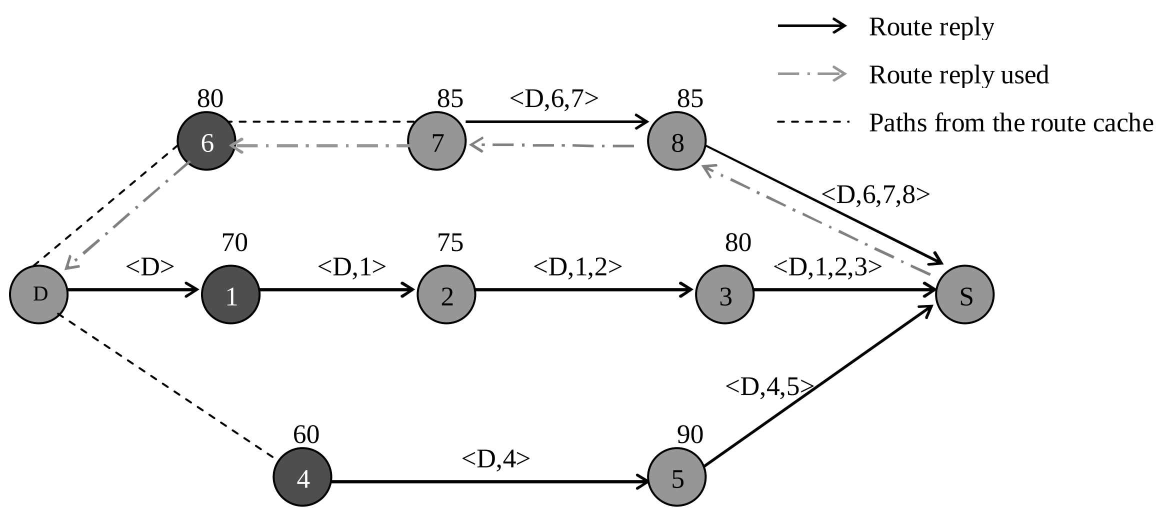

Figure 3.3 Route Discovery mechanism in MMBCR.

Figure 3.3 shows how this scheme works. The numbers above each node represent the remaining battery level of that particular node. Suppose the threshold value is 80. The source node receives the route reply packets through the paths shown above. Node ‘D’ replies through 1, 2, and 3. The intermediate nodes 4 and 7 also send route replies using the paths already present in their route caches. Thus, the source node receives three route replies. None of the paths contained in the replies has all the *rich* nodes. Each path has one or more *poor* nodes. So, according to MMBCR, the source node compares the remaining battery level of the critical node of each path and selects the path whose critical node has the highest energy. The nodes shaded in the dark gray color are the critical nodes. Among these, node 6 has the highest energy. So, the source node selects the path that has node 6 in it.

The *DSR_MAX_WAITING_TIME* is set to be equal to one backoff period. The backoff period is the amount of time for which the source node waits to reinitiate a route request, if it does not receive any route reply within that duration. Route maintenance is done as in the case of DSR using route error messages.

ii) Local Energy Aware Routing (LEAR)

Local Energy Aware Routing (LEAR) [30] is somewhat complicated compared to MMBCR. It uses additional control messages besides the control messages used by the DSR protocol. LEAR tries to optimize the lifetime of the network by considering a node's willingness to forward packets to other nodes. It takes the shortest path among multiple energy rich paths. An intermediate decides whether to forward a route request packet or not depending on its energy level. If its remaining battery level is greater than a threshold value, it forwards the packet. Otherwise, it drops the packet. Even though the concept of threshold is also used in MMBCR, LEAR uses dynamic threshold values whereas MMBCR uses fixed threshold values.

LEAR is also a DSR-based protocol. It works on the same grounds as DSR except for the use of additional control messages used to achieve energy efficiency. In LEAR also, when a node has a packet to send to some node and if does not know any route to the destination, the source node initiates a route discovery procedure. But if an intermediate is not energy rich, it will not forward the route request packet. It simply drops the packet. If all the intermediate nodes in all possible paths drop the route request packet, the route request will never reach the destination node, thus blocking the network. To avoid this, the new control messages *DROP_ROUTE_REQ*, *ROUTE_CACHE*, *DROP_ROUTE_CACHE*, and *CANCEL_ROUTE_CACHE* are used. The protocol works as follows. Each node maintains a *DROP_ROUTE_REQ_TABLE* to maintain the log of all the requests dropped previously.

<u>Protocol Description:</u>

The source node initiates the route discovery, whenever it needs a route to a destination. If the route request packet is received by the destination itself, it sends the route reply immediately. If some intermediate node receives the request and if the request is seen before, it is simply ignored. If the request is not seen before, the node makes an entry for the request in the request seen table. If an entry for the same request already exists in the *DROP_ROUTE_REQ_TABLE*, the node reduces its threshold by *delta*. If there exists a route for the destination in its route cache and if the node's energy is greater than the threshold, it initiates *ROUTE_CACHE* message along the path contained in the cache. Otherwise, it broadcasts the *DROP_ROUTE_REQ* message and makes an entry for that in the *DROP_ROUTE_REQ_TABLE*. If the node does not have any route for the destination and if its energy is greater than the threshold, it relays the route request packet. Otherwise, the node broadcasts the *DROP_ROUTE_REQ* message and makes an entry for that in the *DROP_ROUTE_REQ_TABLE*. Figure 3.4 shows the route discovery mechanism in LEAR.

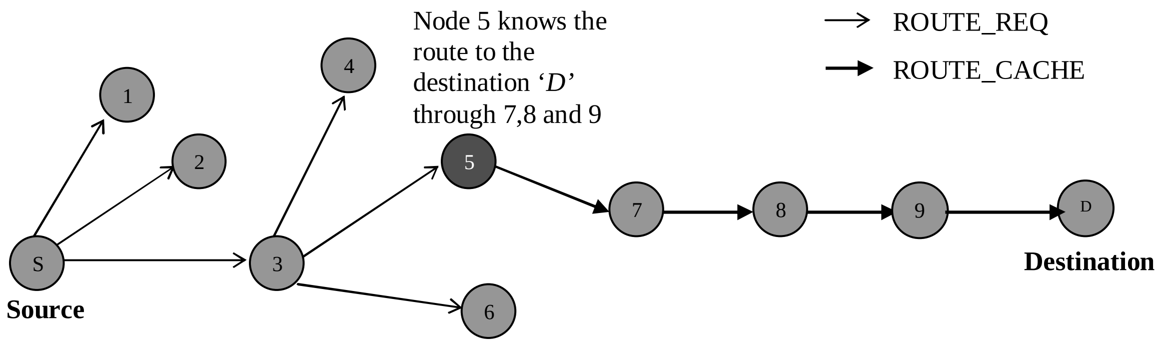

Figure 3.4 Route discovery mechanism in LEAR.

Handling of control messages:

Whenever a node receives a *DROP_ROUTE_REQ* message, it makes an entry in the *DROP_ROUTE_REQ_TABLE* and forwards the packet. When a node receives a *ROUTE_CACHE* message, the node checks its remaining battery level. If the battery level is greater than the threshold, the node forwards the *ROUTE_CACHE* message. Otherwise, it drops the *ROUTE_CACHE* message and sends the *DROP_ROUTE_CACHE* message back to the source and *CANCEL_ROUTE_CACHE* message to the nodes present in the path to the destination it has in its route cache. When a node receives the *DROP_ROUTE_CACHE* message, they make an entry in the *DROP_ROUTE_CACHE_TABLE* and relay the packet. This is showed in Figure 3.6.

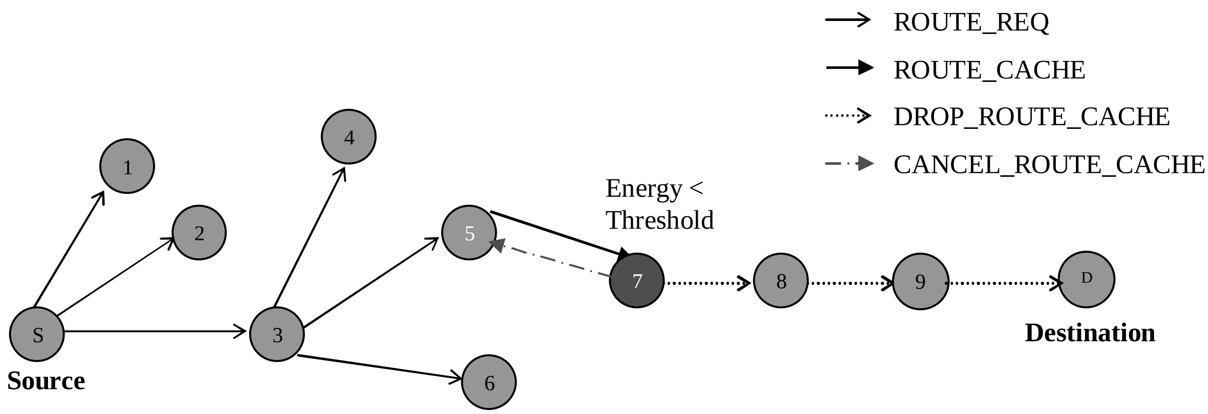

Figure 3.5 Route maintenance in LEAR.

If the source node does not receive any reply within the backoff interval, it reinitiates the route discovery with an increased sequence number. When a node receives the route request packet, it checks the *DROP_ROUTE_REQ_TABLE* and the *DROP_ROUTE_CACHE_TABLE*. If there is already an entry for the same source-destination pair in one of the tables, the node reduces the threshold by *delta* and acts accordingly. The idea behind this is that the node comes to know that some nodes are already getting depleted of the energy and it should not simply drop the packet. The destination node uses the route reply that arrived first.

iii) Request Delay Routing Protocol (RDRP)

Request Delay Routing Protocol (RDRP) [31] is also a DSR-based routing protocol designed to achieve energy balance among the nodes in an ad hoc network. This protocol tries to optimize the network lifetime. At the same time, it also uses the shortest path among all the energy rich paths. The implementation of this protocol is very simple compared to all the other protocols discussed till now.

The protocol works similar to DSR except that the intermediate nodes do not process the request immediately. The intermediate node holds the request packet for some duration *‘d’*, which is inversely proportional to the remaining battery energy level of the node. Each node maintains a request buffer to keep a record of all requests pending. This table keeps all the information about the request including the source and destination addresses, sequence number. The protocol works as follows.

The source node initiates the route request packet whenever it needs a route to some node. A node that receives the request packet checks if it is the destination of the request packet. If it is, it will send a reply back to the destination. If some intermediate node receives the request, the node inserts the request entry into the request table and the request buffer. The entry in the request table is just the sequence number of the request and is used to check if the request is a duplicate one. The entry in the request buffer contains entire information regarding the request. After inserting the entry in the tables, the node sets an event called *REQUEST_DELAY_EVENT* so that it gets an alert *‘d’* seconds after the time of entry into the request buffer. When the node gets the alert, it checks the request buffer to check which entry the alert is due to. Then the node processes that particular request according to the route cache information it has.

Handling the REQUEST_DELAY_EVENT:

Whenever a *REQUEST_DELAY_EVENT* for a node is generated, the node checks its request buffer. It checks for an entry that matches the source and destination address and the entry for which the delay timer has expired. If the node finds such an entry, it processes the corresponding request as the DSR protocol does. The destination node sends the reply as soon as it receives the request.

The value of *‘d’* is figured out from the remaining battery level of the node and is set to be inversely proportional to the battery level of the node. The idea behind this is that the destination node always receives the route request through energy rich nodes. As the delay used by a node to forward a packet is inversely proportional to the energy level of the node, the nodes that are not energy rich will never forward the request packet before the energy rich nodes. This way, the destination node receives the request through energy rich nodes first. As the destination responds to the route request that arrived first, it will always be the energy rich path among all the available paths. The protocol can be explained using an example shown below.

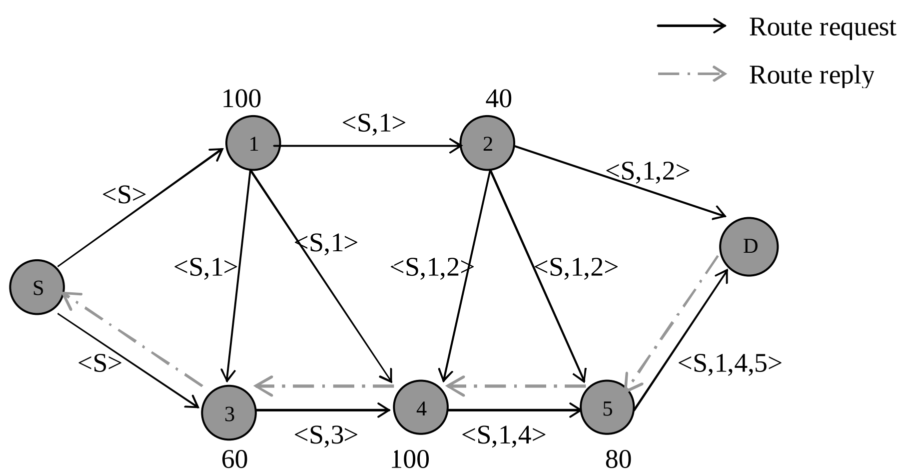

Figure 3.6 Example network topology for RDRP.

In Figure 3.6, the numbers below each node represent the remaining battery level of that node. When the source ‘S’ broadcasts the route request, nodes 1 and 3 receive it. According to RDRP, node 3 keeps the request packet for a longer time without processing it. Node 1 forwards the route request to nodes 2, 3, and 4. Node 3 ignores it as it is a duplicate request. Among nodes 2 and 4, node 4 is energy rich. So, it forwards the request earlier than 2 does. Node 4 forwards the request to 2 and 5. Node 2 ignores the duplicate request and node 5 forwards it to node ‘D’. Thus the route request packet is received through (S, 1, 4, 5) first. As we can observe from the figure, this is the energy rich path among all the paths in the network.

iv) Conditional Max-Min Battery Capacity Routing (CMMBCR)

The *Conditional Max-Min Battery Capacity Routing (CMMBCR)* [28] considers both the total transmission energy consumption of routes and the remaining power of nodes. The remaining battery capacity of the nodes is compared with a predefined threshold value. If all the nodes in a route have sufficient remaining battery capacity (i.e., above the predefined threshold), the route is considered to be rich. If there is more than one rich path among the available replies, the path with minimum transmission power is chosen. If none of the replies contain a rich path, the path whose critical node has the highest remaining battery capacity among all the critical nodes is chosen. Thus, CMMBCR is a combination of MTPR [28] and MMBCR [28].

v) Energy Aware AODV

Energy Aware AODV [32] uses a two-step approach to conserve energy of the nodes. In the first, the routing protocol tries to route the packets avoiding the nodes with low energy levels. In the second, the radio interfaces of the nodes turned off whenever they needed to be to conserve energy further. The nodes are classified into three zones depending on their remaining battery levels. Normal zone consists of nodes with battery energy levels greater that 20% of their initial energy. Normal zone consists of nodes with battery energy levels greater that 10%-20% of their initial energy. The nodes that have less than 10% of their initial energy are in the Danger zone.

The cost of routing packets through the Danger zone is the highest, next comes the cost of routing through the warning zone and the last is the Normal zone. If a node lies in the danger or warning zone and it has a large number of neighbors, the cost of routing packets through that node is directly proportional to the number of neighbors.

In order to ensure that a route is recalculated in the case of energy depletion of a node, the node sends a warning packet to the sources of all the paths using that node. This warning packet is propagated much like a route error packet in ordinary protocol, except that the routes are not erased. The source node initiates a route discovery as soon as it receives a warning packet.

vi) Location-Aided Power-Aware Routing Protocol (LAPAR)

In *Location-Aided Power-Aware Routing Protocol (LAPAR)* [33], the forwarding nodes make the routing of data packets based on the location of the neighbors. The forwarding node sends the packets to the neighboring node whose relay region covers the required destination. If the destination node falls into the relay regions of more than one neighbor, the algorithm makes a greedy choice to select the next hop to forward the packet depending on the distance between the source and the forwarding node and the destination node. The decisions are made locally and they depend on the location of the neighbors.

CHAPTER IV

PERFORMANCE EVALUATION OF ENERGY EFFICIENT ROUTING PROTOCOLS FOR MANETS

This chapter presents the results obtained by implementing the *Dynamic Source Routing Protocol (DSR)* [21], *Min-Max Battery Cost Routing (MMBCR)* [28], *Local Energy Aware Routing (LEAR)* [30], and *Request Delay Routing Protocol (RDRP)* [31] in the *GloMoSim 2.03* [34] simulator. GloMoSim is a scalable simulation environment for wireless and wired networks and is based on *Parsec* [35]. In this thesis, I have used the existing implementation for the DSR protocol and modified the code for the DSR protocol in order to implement MMBCR, LEAR, and RDRP. This chapter is organized as follows. Section 4.1 describes the simulation environment and various parameters used in the simulations. Section 4.2 explains the performance metrics used in this thesis. Section 4.3 presents the simulation results and discussion. These include standard deviation of the energy consumed by all the nodes (Section 4.3.1), peak-to-mean ratio of the energy consumption (Section 4.3.2), latency in delivering the packets (Section 4.3.3) and the packet delivery ratio (Section 4.3.4) with respect to pause time. Section 4.4 describes how the threshold value affects the performance of LEAR and MMBCR.

4.1 Simulation Environment

Table 4.1 Configuration file for the simulator.

# Duration of the simulation: 900 seconds SIMULATION-TIME 15M #Random seed SEED 3 # Terrain dimensions: 1500m x 300m TERRAIN-DIMENSIONS (1500, 300) #Number of nodes used in the simulation: 50 NUMBER-OF-NODES 50 #Node placement pattern: random NODE-PLACEMENT RANDOM #Node movement model: random-waypoint # pause times used: 0, 40, 80, 120, 200, 300, 600, # and 900 seconds # speed varies from 0 meters/sec to 20 meters/sec MOBILITY RANDOM-WAYPOINT MOBILITY-WP-PAUSE 0S MOBILITY-WP-MIN-SPEED 0 MOBILITY-WP-MAX-SPEED 20 #Other mobility parameters MOBILITY-POSITION-GRANULARITY 0.5 #Propagation limit (in dBm): -111.0 PROPAGATION-LIMIT -111.0 #Path loss model: two ray model PROPAGATION-PATHLOSS TWO-RAY #Noise figure NOISE-FIGURE 10.0	#Radio bandwidth (in bits per second): 2000000 RADIO-BANDWIDTH 2000000 #Packet reception model: SNR-BOUNDED RADIO-RX-TYPE SNR-BOUNDED RADIO-RX-SNR-THRESHOLD 10.0 #Radio transmission power used (in dBm): 7.874 RADIO-TX-POWER 7.874 #Antenna gain (in dB): 0 RADIO-ANTENNA-GAIN 0.0 #Sensitivity of the radio (in dBm): -91 RADIO-RX-SENSITIVITY -91.0 #Minimum power for received packet (in dBm) RADIO-RX-THRESHOLD -81.0 #MAC protocol used: IEEE 802.11 MAC-PROTOCOL 802.11 #No promiscuous mode was used PROMISCUOUS-MODE NO #Network protocol: IP NETWORK-PROTOCOL IP NETWORK-OUTPUT-QUEUE-SIZE-PER-PRIORITY 100 #Routing protocol: DSR ROUTING-PROTOCOL DSR #Statistics used APPLICATION-STATISTICS YES TCP-STATISTICS NO UDP-STATISTICS NO

#Temperature of the environment (in K): 290 TEMPARATURE 290.0 #Radio model used to receive and send packets RADIO-TYPE RADIO-ACCNOISE #Radio frequency (in Hertz): 2.4e9 RADIO-FREQUENCY 2.4e9	ROUTING-STATISTICS YES NETWORK-LAYER-STATISTICS NO MAC-LAYER-STATISTICS NO RADIO-LAYER-STATISTICS YES CHANNEL-LAYER-STATISTICS NO MOBILITY-STATISTICS NO

Table 4.1 (Contd.)

Table 4.1 shows the simulation parameters and their values. The transmission power of each node is set so that its radio range is 250 meters. Nodes move randomly and follow *random waypoint* model. The model is defined using the parameters the *minimum speed*, the *maximum speed*, and the *pause time*. Each node stays for the period of pause time seconds at the initial position and moves to a random direction at a random speed varying between the minimum and maximum speeds. The node repeats the same procedure after reaching the destination. The same process continues throughout the duration of the simulation. The random direction and speed are chosen depending on the seed number used in the simulation.

Also, the simulations used 10 *constant bit rate* (CBR) sources. The CBR source can be defined using the following parameters: number of packets to be sent per second, packet size, duration of the transmission, start time of the transmission, and end time of the transmission. In this thesis, each source used sends four 512-byte packets per second for the duration specified while defining the CBR source.

In this thesis, energy spent by the nodes while transmitting data is taken into consideration. The expression to evaluate the energy spent in transmitting a data packet is $E = i * v * t$ Joules, where i is the current used to transmit the packet, v is the voltage used to transmit the packet, and t is the duration of the transmission. In this, all the nodes are assumed to be equipped with IEEE 802.11b network interface cards. Also, the power consumption during the reception and idling periods is assumed to be zero.

4.2 Performance Metrics

The metrics used in this thesis to evaluate the performance of the protocols in terms of energy balance in the network are the *standard deviation* (SD) of the energy consumed by all the nodes in the network and the *peak-to-mean ratio* (PMR) of the energy consumption of the nodes in the network. Apart from achieving good energy balance in the network, a good network protocol should be able to deliver the data packet reliably and quickly. The metrics used to evaluate the general performance of a network protocol are the *packet delivery ratio* (PDR) and the average *latency* in delivering packets from the source to the destination.

Standard deviation (SD) of the energy consumed by the all the nodes in the network gives an estimate of the different between the energy levels of the nodes in the network. A network is completely balanced only if all the nodes in the network possess equal energy level. A high value of SD means that some of the nodes are being overused while some nodes are not used at all. This metric represents the energy balance in the network and can also be used to estimate the overall lifetime of the network. The better energy balance the network shows, the more the network lifetime is.

Peak-to-Mean Ratio (PMR) is another metric used to estimate the energy balance in the network. PMR is the ratio between the peak value of energy consumed among all the nodes in the network and the average value of the energy consumed by all the nodes in the network. SD gives an idea about the difference in the energy levels of the nodes in the network, but does not give the average energy level of the nodes in the network. It is possible that all the nodes in the network possess very low but equal energy levels, giving a low SD. In such cases, using SD itself as a metric to estimate the energy balance in the network is not a good idea. SD and PMR both can give an idea of the performance of a protocol with respect to energy balance. A value of PMR closer to 1 means that the energy balance in the network is good.

Packet Delivery Ratio (PDR) is the ratio between the number of packets transmitted by the source node and the number of packets received successfully by the destination node. PDR gives a measure of the reliability of the protocol.

4.3 Simulation Results

In this thesis, the energy consumption of nodes using DSR protocol is compared with that of the nodes using MMBCR, LEAR, and RDRP protocols. The results presented here are obtained by taking an average of the results from 80 simulation runs. The simulations are done at 8 values of pause time. At each value of pause time, the simulations are repeated 10 times using 10 different seed numbers. The same set of seed numbers are used with each pause time. The threshold value of energy used in the simulations for MMBCR and LEAR is equal to 99% of the initial battery level of the nodes (see section 4.4). This section presents the results in terms of SD, PMR, PDR, and the average latency in delivering data packets with respect to pause time. The results are presented at two different node movement speeds: 20 meters/sec and 1 meter/sec.

One important point should be noted before analyzing the results presented below. For the pause times below 100 seconds, there is a lot of irregular variation in the performance metrics considered. The reason for this behavior is that the nodes are in constant motion at low pause times and thus the network topology keeps changing quickly. In such situations, it is difficult to optimize the performance of the network because of frequent link breakage. So, the results are analyzed only for pause times above 100 seconds. Also, the energy measurements are taken excluding the source and destination nodes since those nodes should always be ON.

4.3.1 Standard Deviation (SD)

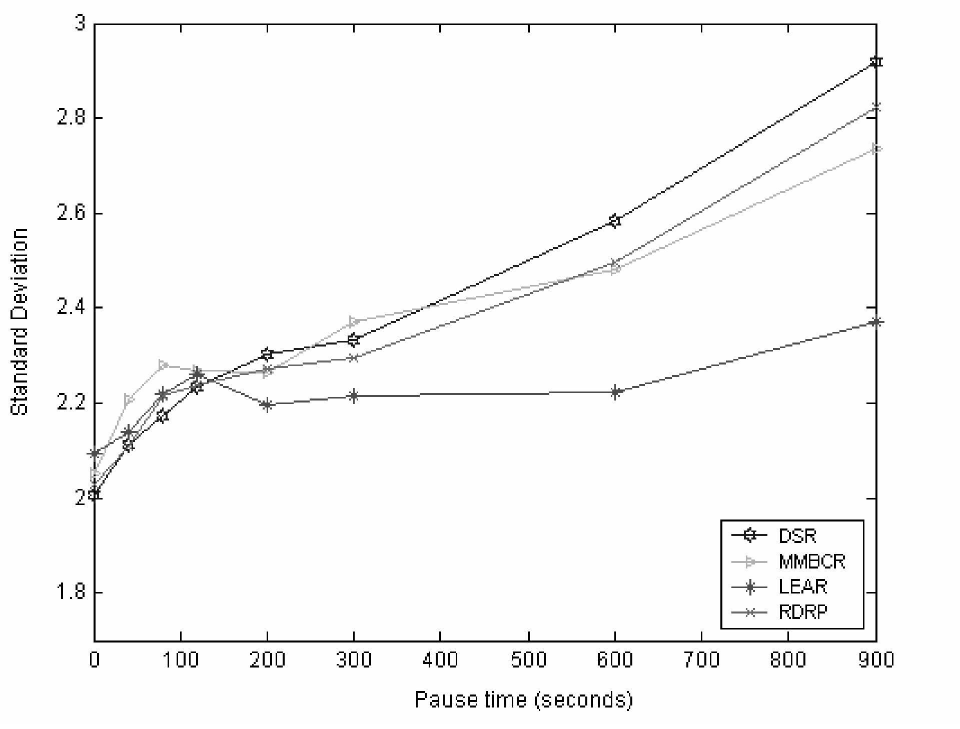

Figure 4.1 (a) Standard Deviation at node speed of 20 meters/sec.

Figure 4.1 (a) shows the plot of standard deviation of the energy consumed by all the nodes in the network versus pause time at node speed of 20 meters/sec. As we can see from the graph, all the energy efficient protocols (MMBCR, LEAR, and RDRP) give less standard deviation compared to the DSR protocol. If we observe each protocol independently, we can see that the standard deviation increases as the pause time increases. This is because the node movement is relatively less at higher pause times. So, the links do not break quickly making the nodes use the old routes and thus reducing the energy balance in the network. LEAR tries to avoid such routes by making the nodes check their energy level every time they transmit a packet. If the energy level of the node is less than the threshold, the node would not transmit the packet and it sends an error message saying that it cannot transmit the packet. The standard deviation does not increase drastically with LEAR, as that compared to other protocols.

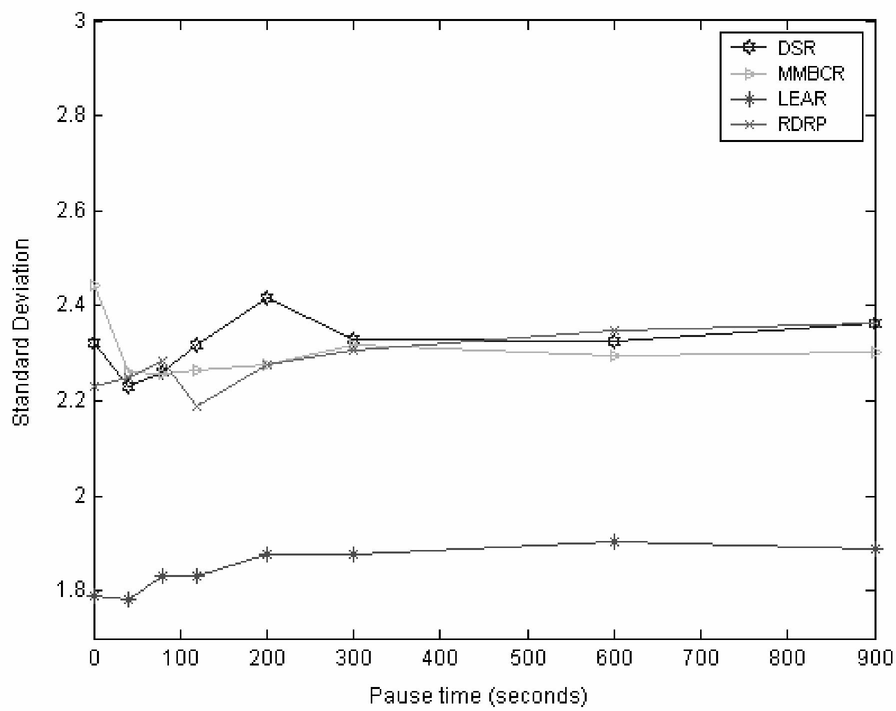

Figure 4.1 (b) Standard Deviation at node speed of 1 meter/sec.

Figure 4.1 (b) shows the plot of standard deviation versus pause time at node speed of 1 meter/sec. As the pause time increases, the standard deviation increases slightly in the beginning and it becomes constant after that. This is because the network acts more like a static network at higher pause times and the nodes move at very low speed. This causes more stable links and the energy efficient protocol does not affect the standard deviation much. As we said before, the nodes in LEAR check their energy level every time it passes a packet to the upper layer. That makes LEAR most energy efficient compared to the other energy efficient protocols.

4.3.2 Peak-to-Mean Ratio (PMR)

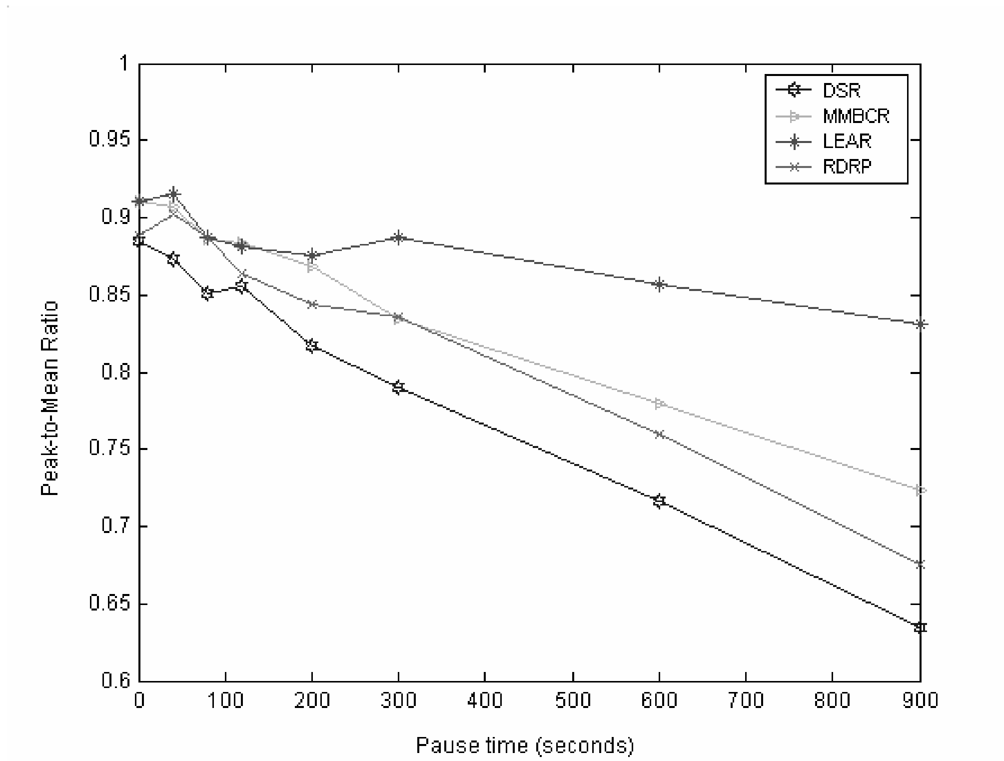

Figure 4.2 (a) Peak-to-Mean Ratio at node speed of 20 meters/sec.

Figure 4.2 (a) shows the plot of Peak-to-Mean Ratio (PMR) of energy consumed by all nodes in the network with respect to the pause time at node speed of 20 meters/sec. As explained before, the value of PMR equal to 1 represents a network, which has complete energy balance. The protocols LEAR, RDRP, and MMBCR show the PMR values closer to 1 compared to DSR. The value of PMR goes away from 1 as the pause time increases. This is because of the same reason explained for Figure 4.1. As the network becomes stable, the nodes keep using the previously obtained routes for a long time, thus overusing some nodes. This reduces the energy balance in the network, thus reducing the value of PMR.

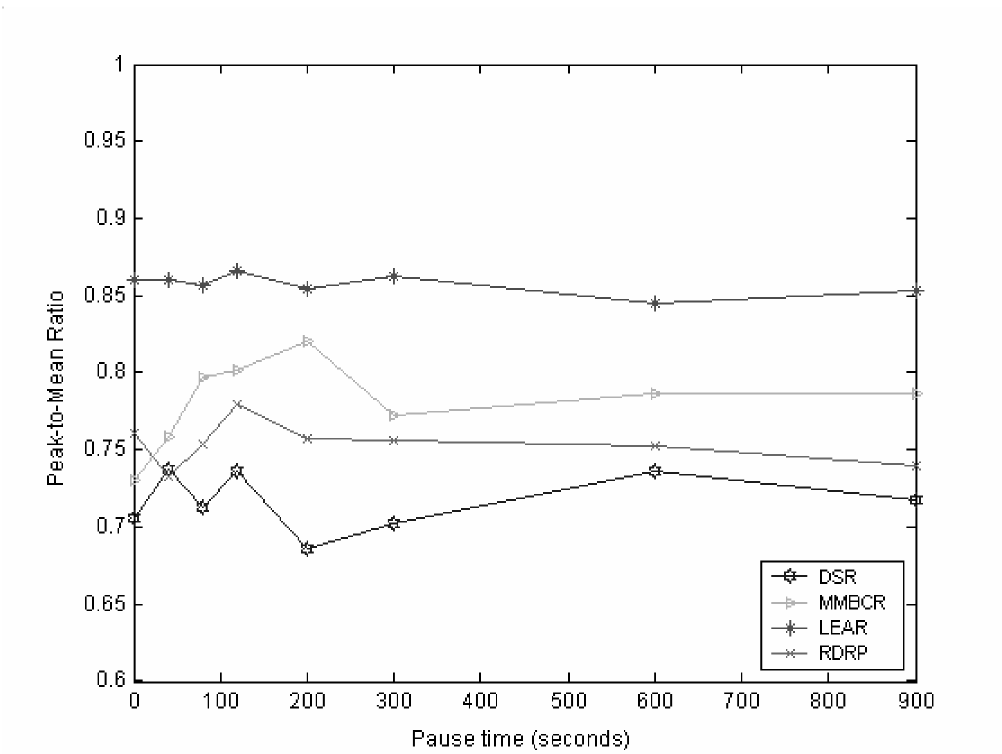

Figure 4.2 (b) Peak-to-Mean Ratio at node speed of 1 meter/sec.

Figure 4.2 (b) shows the plot of PMR versus pause time at node speed of 1 meter/sec. Again, the curves are almost flat at lower speed because of the increased network stability.

4.3.3 Latency

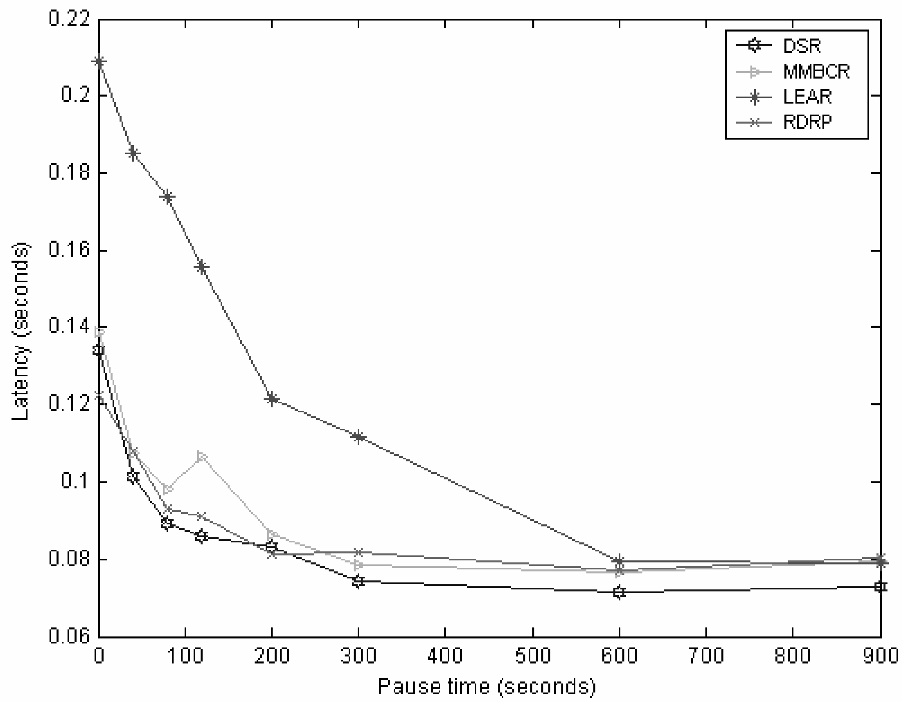

Figure 4.3 (a) Average Latency in delivering the packets at node speed of 20 meters/sec.

Figure 4.3 (a) shows the plot of average latency in delivering the packets with respect to pause time at node speed of 20 meters/sec. Apart from the irregular variations at the pause times below 100 seconds; DSR shows better performance compared to the energy efficient protocols. There is always a trade-off between the energy conservation and the delay and packet delivery ratio. As we have seen in chapter 3, the protocols MMBCR, LEAR, and RDRP use some scheme to achieve energy balance in the network. They do not use the route reply that is received first. Instead, they try to select the route that optimizes the energy balance in the network. In order to this, the nodes implementing these protocols wait till they detect that optimum route. This causes the latency in delivering the data packets. As we can see from the graph, the latency keeps on decreasing with the increase in pause time and goes flat after some time. The reason for this is that as the pause time increases, the network becomes steadier, reducing the number of link breaks and thus the nodes tend to use the routes previously obtained. We can also notice the highest latency with LEAR protocol. This is due to the complex route discovery procedure using more control messages in LEAR.

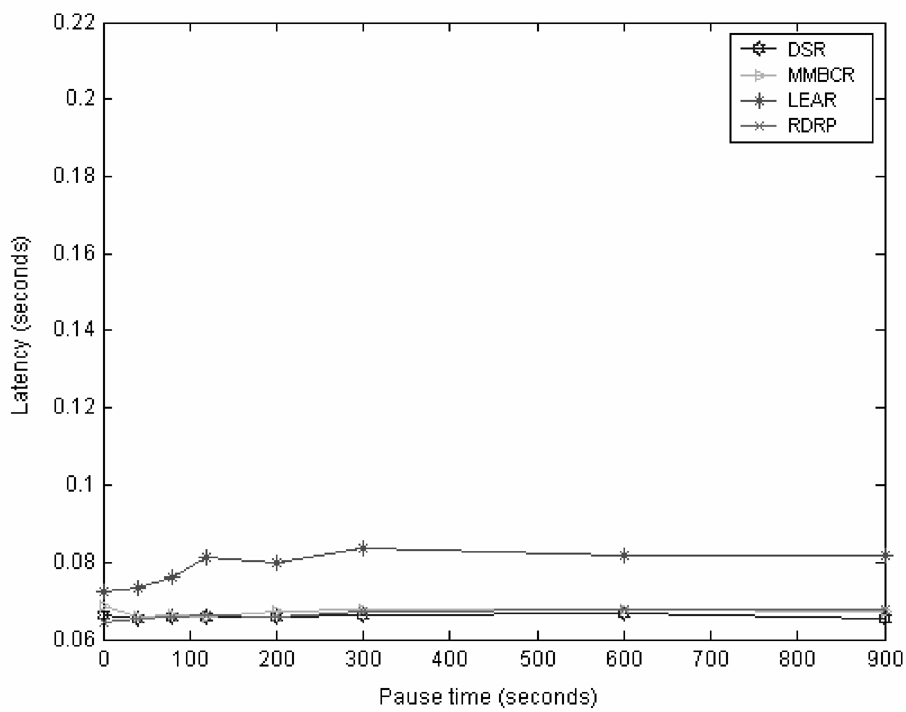

Figure 4.3 (b) Average Latency in delivering the packets at node speed of 1 meter/sec.

Figure 4.3 (b) shows the plot of average latency versus pause time at node speed of 1 meter/sec. As we can see from the graph, the latency is very low compared to that at higher speed. This is because the nodes do not move fast and discovering the routes would not take too long. This reduces the average latency in transmitting the packet. The protocols MMBCR and RDRP show the same performance as DSR, as these protocols do not discover a new route unless the existing link breaks. But, LEAR tries to keep the energy balance in the network by checking the node's energy level at every hop.

4.3.4 Packet Delivery Ratio (PDR)

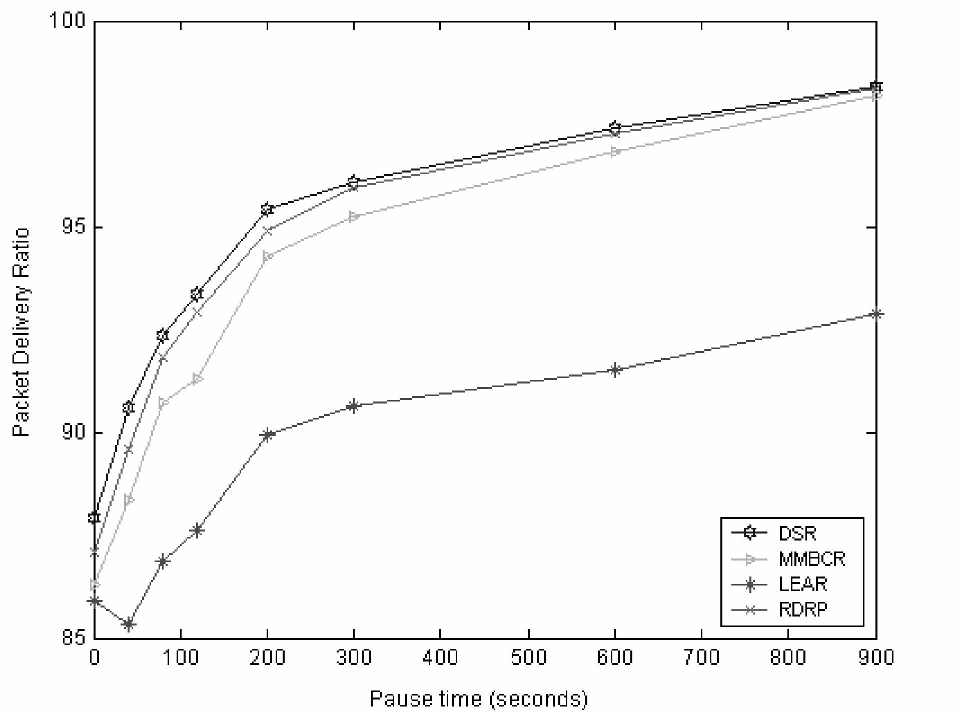

Figure 4.4 (a) Packet Delivery Ratio at node speed of 20 meters/sec.

Figure 4.4 (a) shows the plot of packet delivery ratio (PDR) versus pause time at node speed of 20 meters/sec. PDR is an important metric in assessing the performance of a protocol. The PDR increases with the increase in pause time. This is due to the stabilized network conditions. The protocols MMBCR and RDRP give a PDR almost equal to that of DSR, but the PDR of LEAR is significantly lower than that of DSR. This can be explained as follows. Even though MMBCR and RDRP implement energy efficient schemes at the routing layer, these protocols do not update the route until it is broken. In that case, the route need not always attain energy balance in the network. Unlike these two protocols, LEAR takes into consideration the willingness of the node to forward data packets at each hop. If the node's energy level is level is less than the threshold value, the packet is simply dropped before it is sent to the upper layer. This causes the significant reduction in the PDR for LEAR. But at the same time, we can see the significant improvement in the standard deviation with LEAR, thus improving the energy balance in the network.

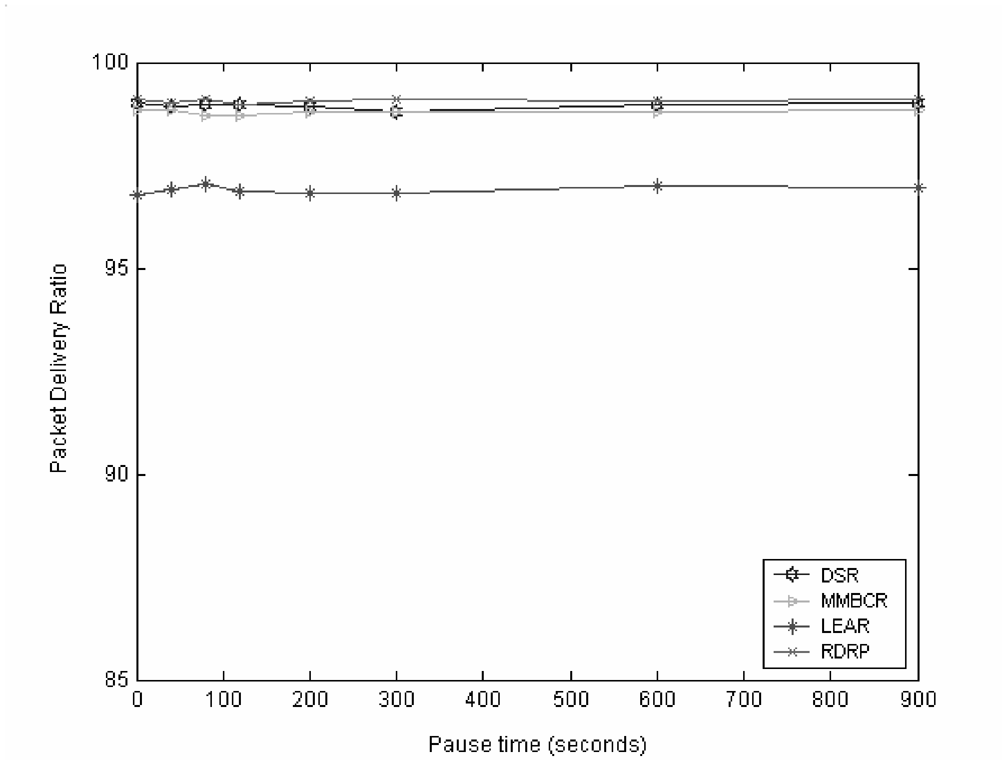

Figure 4.4 (b) Packet Delivery Ratio at node speed of 1 meter/sec.

Figure 4.4 (b) shows the plot of PDR versus pause time at node speed of 1 meter/sec. Again, the curves are more flat compared to that at higher speeds. Also, the protocols MMBCR, and RDRP give the same performance as DSR, while LEAR gives somewhat lesser PDR. This is due to the packets dropped by the nodes before passing it to the higher layer.

4.4 Effect of threshold value on the performance of LEAR & MMBCR protocols

In chapter 3, it is explained that the battery level of a node is compared to a threshold value while implementing the protocols MMBCR and LEAR. The value of the threshold chosen has a significant impact on the performance of the protocol. Table 4.2 shows how the value of the threshold affects the Standard Deviation offered by MMBCR and LEAR protocols. The first column in Table 4.2 shows the value of threshold as a percentage of the initial battery level of the node, and the columns 2 and 3 show the percentage improvement in the SD compared to the DSR protocol. As the value of the threshold increases, the protocols show better performance. LEAR shows regular improvement in the SD with increase in the threshold.

At the threshold of ‘0’, LEAR acts similar to DSR. As the threshold is increased, LEAR does not show any improvement in the performance till the threshold is increased above 50% of the initial battery level of the node. LEAR shows the best performance when the threshold is increased to 99% of the initial battery level of the node.

Table 4.2 Effect of threshold on the SD offered by MMBCR and LEAR.

Threshold (in % of the initial battery level of the nodes)	Percentage improvement in the Standard Deviation compared to that offered by DSR	
	MMBCR	**LEAR**
100 %	2.7 %	20 %
99 %	7 %	24 %
98 %	0 %	14 %
90 %	2.6 %	14 %
83 %	2 %	13 %
50 %	0 %	2 %
25 %	0 %	2 %
0 %	0 %	0%

MMBCR shows no improvement in the SD until the threshold is above 50% of the initial battery level of the node. MMBCR shows irregular variation when the threshold is 98% of the initial battery level. Also, MMBCR shows best performance when the threshold is 99% of the initial battery level of the node. The reason for no improvement in SD at lower threshold is that all the nodes are treated equally because they all possess energy levels higher than the threshold value. In such cases, the effect of using energy efficient protocols is less. Based on these results, the initial threshold value was kept at 99% of the initial battery level of the node for the simulations used in this thesis.

CHAPTER V

CONCLUSIONS

This thesis analyzes the performance of energy efficient routing protocols compared to a protocol with no energy efficient scheme. As we have seen in chapter 4, the energy efficient protocols show better performance in terms of energy conservation. We can observe from the results that there is always a trade-off between the energy conservation of a network protocol and the general performance in terms of PDR and latency. This study will be useful in selecting a routing protocol depending on the application. For applications such as sensor networks, the lifetime of a mobile node is very important. In such applications, energy efficient protocols play a significant role in enhancing the network lifetime. For applications where the reliable and quick delivery of data is important rather than the energy balance, ordinary routing protocols are preferred.

According to the simulation results with DSR, MMBCR, LEAR and RDRP; LEAR gives the best performance from the energy conservation perspective. LEAR shows improvement in the SD of up to 25% compared to DSR, while the protocols

MMBCR and RDRP show the improvement in SD by 6%-10% compared to DSR. Also, LEAR shows improvement in the PMR up to 32% compared to DSR, while the other two protocols show improvement in PMR of 8%-15% compared to DSR. The reason for the better performance of LEAR compared to MMBCR and RDRP is that LEAR takes the node's willingness to forward packets (both data and control packets) into consideration, while the other protocols do not update the energy efficient route till it is broken. Even though this causes additional delays, it achieves good energy balance in the network. This makes LEAR the best protocol in terms of energy conservation among all the protocols considered in this study.

In this thesis, only the energy efficient schemes at the routing layer are implemented and evaluated. There is also a lot of research being done in the other layers of the protocol stack. This study can be extended to design protocols with hybrid schemes with some energy efficient scheme implemented at the routing layer and also with energy efficient schemes at the other layers of the protocol stack to achieve improved energy efficiency.

BIBLIOGRAPHY

1. J. Schiller, “Mobile Communications”, *Addison-Wesley* (2000).
2. C. E. Perkins, “Ad Hoc Networking”, *Addison-Wesley* (2001).
3. IEEE, “Wireless LAN medium access control (MAC) and physical layer (PHY) Spec”, *IEEE 802.11 standard* (1998).
4. P. Karn, “MACA – A new channel access method for packet radio”, *ARRL/CRRL Amateur Radio Computer Networking Conference* (1990), pp. 134–140.
5. E. S. Jung, and N. H. Vaidya, “An Energy Efficient MAC Protocol for Wireless LANs”, *INFOCOM* (2002).
6. S. Singh and C.S. Raghavendra, “PAMAS: Power aware multiaccess protocol with signalling for ad hoc networks”, *Computer Communication Review* 28(3) (1998), pp. 5–26.
7. Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed Energy Conservation for Ad Hoc Routing”, *MobiCom* (2001), pp. 70-84.

8. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", *Wireless Networks* 8(5) (2002), pp. 481-494.
9. J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian, "Conserving Transmission Power in Wireless Ad Hoc Networks", *IEEE Conference on Network Protocols* (2001).
10. M. Zorzi and R. R. Rao, "Energy constrained error control for wireless channels", *IEEE Personal Communications* 4(6) (1997), pp. 27-33.
11. M. Zorzi and R.R. Rao, "Error control and energy consumption in communications for nomadic computing", *IEEE Transactions on Computers* 46(3) (1997), pp. 279–289.
12. P. Lettieri, C. Fragouli and M.B. Srivastava, "Low power error control for wireless links", *MobiCom* (1997), pp. 139-150.
13. E. Erkip, Y. Wang, D. Goodman, Y. Wu and X. Lu, "Energy Efficient Coding and Transmission", *IEEE Vehicular Technology Conference* (2001).
14. P. Lettieri and M. B. Srivastava, "Adaptive frame length control for improving wireless link throughput, range, and energy efficiency", *INFOCOM* 2 (1998), pp. 564–571.
15. M. Zorzi and R.R. Rao, "Energy efficiency of TCP in a local wireless environment", *Mobile Networks and Applications* 6 (3) (2001), pp. 265-278.
16. V. Tsaoussidis, A. Lahanas and H. Badr, "Wave & Wait protocol (WWP): low energy, high throughput for mobile IP-devices", *IEEE Conference on Networks* (2000).

17. W. Lilakiatsakun and A. Seneviratne, “Enhancing TCP energy efficiency for mobile hosts”, *IEEE Conference on Networks* (2002).

18. D. Bertozzi, A. Raghunathan, L. Benini and S. Ravi, “Transport protocol optimization for energy efficient wireless embedded systems”, *Design, Automation and Test in Europe Conference and Exhibition* (2003), pp. 10706-10713.

19. J. Flinn and M. Satyanarayanan, “Energy-aware adaptation for mobile applications”, *Symposium on Operating Systems Principles* (1999), pp 48-63.

20. C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Computer Communication Review* (1994), pp.234-244.

21. D. B. Johnson, D. A. Maltz, "Dynamic Source Routing in Ad Hoc Networks", *Mobile Computing* (1996), pp. 153-181.

22. C. E. Perkins, E. M. Royer, S. R. Das, "Ad Hoc On-demand Distance Vector Routing", *Internet Engineering Task Force Draft* (1999).

23. J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", *ACM/IEEE International Conference on Mobile Computing and Networking* (1998), pp 85-97.

24. M. Woo, S. Singh, and C. S. Raghavendra, “Power aware routing in mobile ad hoc networks (I)”, *ACM MobiCom* (1998), pp. 181-190.

25. C. Yu, B. Lee and H. Y. Youn, “Energy efficient routing protocols for mobile ad hoc networks”, *Wireless Communications and Mobile Computing* 3(8) (2003), pp. 959-973.

26. J-H. Chang, L. Tassiulas, “Energy Conserving Routing in Wireless Ad-hoc Networks”, *Conference on Computer Communications* (2000), pp. 22-31.

27. Q. Li, J. Aslam, D. Rus, “Online Power-aware Routing in Wireless Ad-hoc Networks”, *International Conference on Mobile Computing and Networking* (2001), pp. 97-107.

28. C-K. Toh, “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks”, *IEEE Communications* 39(6) (2001), pp. 138-147.

29. P. Bergamo, A. Giovanardi, A. Travasoni, D. Maniezzo, G. Mazzini, and M. Zorzi, “Distributed Power Control for Energy Efficient Routing in Ad Hoc Networks”, *European Wireless* (2002), pp. 237-243.

30. K. Woo, C. Yu, H. Y. Youn, B. Lee, “Non-Blocking, Localized Routing Algorithm for Balanced Energy Consumption in Mobile Ad Hoc Networks”, *International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (2001), pp. 117-124.

31. W. Yu, J. Lee, “DSR-based Energy-aware Routing Protocols in ad hoc networks”, *International Conference on Wireless Networks* (2002).

32. N. Gupta and S. R. Das, “Energy Aware On-Demand Routing for Mobile Ad Hoc Networks”, *International Workshop on Distributed Computing* (2002), pp. 164-173.

33. Y. Xue, B. Li, “A Location Aided Power Aware Routing in Mobile Ad Hoc Networks” *IEEE Globecom* (2001), pp. 2837-2841.

34. X. Zeng, R. Bargodia and M. Gerla, “GloMoSim: A library for the parallel simulation of large-scale wireless networks”, *Parallel and distributed Simulation Conference* (1998).

35. R. Bargodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song, “PARSEC: A Parallel Simulation Environment for Complex Systems”, *IEEE Computer* 31(10)(1998), pp. 77-85.