

## Power-Aware On-Demand Routing Protocols for Mobile Ad Hoc Networks

### 32.1 Introduction

### 32.2 MANET Routing Protocols

Proactive (Table-Driven) Routing Protocols, Reactive (On-Demand) Protocols, Hybrid Routing Protocols

### 32.3 Low Power Routing Protocols

Minimum Power Routing, Battery-Cost Lifetime-Aware Routing, Energy Conserving Techniques for Multi-hop Ad Hoc Networks, Energy-Aware Multicast Routing Algorithms

### 32.4 Power-Aware Source Routing

Cost Function, Route Discovery, Route Maintenance

### 32.5 Lifetime Prediction Routing

Basic Mechanism, Route Discovery, Route Expiration

**Error! Reference source not found.** Quantitative Evaluation of Source Routing Algorithms

Simulation Setup, Simulation Results

### 32.7 Conclusion

Morteza Maleki and Massoud Pedram

*University of Southern California, Los Angeles, CA*

### 32.1 Introduction

Wireless mobile networks may be classified into these two general categories:

*Infrastructure-based Networks:* Wireless networks often extend, rather than replace, wired networks, and are referred to as infrastructure networks. A hierarchy of wide area and local area wired networks is used as the backbone network. The wired backbone connects to special switching nodes called *base stations*. They are responsible for coordinating access to one or more transmission channel(s) for mobiles located within their coverage area. The end user nodes communicate via the base station using their respective wireless interfaces. Wireless LANs and WANs are a good example of this type.

*Mobile Ad hoc Networks (MANETs):* A MANET is composed of a group of mobile wireless nodes that form a network independently of any centralized administration, while forwarding packets to each other in a multi-hop manner. Since the mobile devices are battery-powered, extending the network lifetime has become an important objective. Researchers and practitioners have focused on power-aware design of network protocols for the ad hoc networking environment. Since each

mobile node in a MANET performs the routing functions for establishing communication among different nodes, the “death” of even a few nodes, due to energy exhaustion, might cause the disruption of service in the entire network. The main focus of this chapter is survey and design of power-aware unicast and multicast routing protocols and algorithms for wireless ad hoc networks with special attention to MANETs.

Metrics used by conventional routing protocols for the wired Internet, which is oblivious to an energy budget, typically do not need to consider any energy-related parameters. Thus, RIP [1] uses hop count as the sole route quality metric, thereby, selecting minimum-hop paths between the source and destinations. OSPF [2], on the other hand, supports additional link metrics such as available bandwidth, link propagation delay etc. These algorithms, however, may result in a rapid depletion of the battery energy in the nodes along the most heavily-used paths in the network. Routing protocols for wireless ad hoc environments contain special features to reduce the signaling overheads and convergence problems caused by node mobility and potential link failures. While these protocols do not necessarily compute the absolute minimum-cost path, they aim at selecting paths that have lower cost (in terms of metrics such as hop count or delay). Such protocols must be modified to yield energy-efficient routing solutions.

A large number of researchers have addressed the problem of energy-efficient data transfer in the context of multi-hop wireless networks. Existing protocols may be classified into two distinct categories. One category of protocols is based on minimum-power routing algorithms, which focus on minimizing the power requirements over end-to-end paths. A typical protocol in this category selects a routing path from a source to some destination so as to minimize the total energy consumption for transmitting a fixed number of packets over that path. Each link cost is set to the energy required for transmitting one packet of data across that link and Dijkstra’s shortest path algorithm is used to find the path with the minimum total energy consumption. These protocols traditionally ignore the power dissipated on the receiver side in a node, and therefore, tend to result in routing paths with a large number of short hops. A key disadvantage of these protocols is that they repeatedly select the least-power cost routes between source-destination pairs. As a result, nodes along these least-power cost routes tend to “die” soon by rapidly exhausting their battery energy. This is doubly harmful since the nodes that die early are precisely the ones that are most needed to maintain the network connectivity (and hence increase the useful service life of the network.)

A second category of protocols is based on routing algorithms that attempt to increase the *network lifetime* by attempting to distribute the forwarding load over multiple different paths. This distribution is performed by either intelligently reducing the set of nodes needed to perform the forwarding duties, thereby, allowing a subset of nodes to sleep over different periods of time, or by using heuristics that consider the residual battery power at different nodes and route around nodes that have a low level of remaining battery energy. In this way, they balance the traffic load inside the MANET so as to increase the battery lifetime of the nodes and the overall useful life of an ad hoc network. These protocols indeed constitute the state-of-the-art in power-aware network routing protocols and will be the focus of this chapter.

This chapter is organized as follows. Section 32.2 gives a brief classification of the broad domain of ad hoc routing protocols. Section 32.3 gives a brief literature review of research in power-aware ad hoc routing protocols. Section 32.4 describes the rationale and details of the Power-aware Source Routing (PSR) algorithm and likewise section 32.5 describes

the rationale and details of the proposed Lifetime Prediction Routing (LPR) algorithm. Section **Error! Reference source not found.** contains the experimental results comparing PSR and LPR with other popular ad hoc routing techniques.

## 32.2 MANET Routing Protocols

Routing protocols in ad hoc networks may be classified into three groups: *proactive* (table-driven), *reactive* (on-demand), and *hybrid*.

### Proactive (Table-Driven) Routing Protocols

These routing protocols are similar to and come as a natural extension of those for the wired networks. In proactive routing, each node has one or more tables that contain the latest information of the routes to any node in the network. Each row has the next hop for reaching to a node/subnet and the cost of this route. Various table-driven protocols differ in the way the information about change in topology is propagated through all nodes in the network.

The two kinds of table updating in proactive protocols are the periodic update and the triggered update [3]. In periodic update, each node periodically broadcasts its table in the network. Each node just arriving in the network receives that table. In triggered update, as soon as a node detects a change in its neighborhood, it broadcasts entries in its routing table that have changed as a result. Examples of this class of ad hoc routing protocols are the Destination-Sequenced Distance-Vector (DSDV) [4] and the Wireless Routing Protocol (WRP) [5]. Proactive routing tends to waste bandwidth and power in the network because of the need to broadcast the routing tables/updates. Furthermore, as the number of nodes in the MANET increases, the size of the table will increase; this can become a problem in and of itself.

DSDV, which is known not to be suitable for large dense networks, was described in [3]. A route table at each node enumerates all available destinations and the corresponding hop-count from the node. Each route table entry is tagged with a sequence number, which is created by a destination node. To maintain consistency of the route tables in a dynamically changing network topology, each node transmits table updates either periodically (*periodic update*) or when new significant information is available (*triggered update*). Routing information is advertised by broadcasting or multicasting. The packets are transmitted periodically and incrementally as topological changes are detected. Topological changes include movement of a node from place to place or the disappearance of the node from the network. Information about the time interval between arrival of the very first routing solution and the arrival of the best routing solution for each particular destination is also maintained. On the basis of this information, a decision may be made to delay advertising routes that are about to change, thus, reducing fluctuations in the route tables. The advertisement of possible unstable routes is delayed to reduce the number of rebroadcasts of possible route entries that normally arrive with the same sequence number.

### Reactive (On-Demand) Protocols

Reactive routing protocols take a lazy approach to routing. They do not maintain or constantly update their route tables with the latest route topology. Instead, when a source node wants to transmit a message, it floods a query into the network to discover the route to the destination. This discovery packet is called the *Route Request (RREQ)* packet and the mechanism is called Route Discovery. The destination replies with a *Route Reply (RREP)* packet. As a result, the source dynamically finds the route to the destination. The discovered route is maintained until the destination node becomes inaccessible or until the route is no longer desired.

The protocols in this class differ in handling cache routes and in the way route discoveries and route replies are handled. Reactive protocols are generally considered efficient when the route discovery is employed rather infrequently in comparison to the data transfer. Although the network topology changes dynamically, the network traffic caused by the route discovery step is low compared to the total communication bandwidth. Examples of Reactive routing protocols are the *Dynamic Source Routing (DSR)* [3][6], the ad hoc on-demand Distance Vector Routing (AODV) [7] and the Temporally-Ordered Routing Algorithm (TORA)[34]. The proposed power-aware routing algorithms belong to this category of routing algorithms. Since our approach is an enhancement over DSR, a brief description of DSR is warranted.

DSR, which is one of the widely-accepted reactive routing protocols, is entirely on demand with no periodic activity of any kind at any level within the network. This pure on-demand behavior allows the number of routing discovery packets for a set of communication patterns to scale to zero when all nodes are approximately stationary. This is due to the fact that if nodes are not moving about, all the routes employed by the current set of communication patterns will be discovered and will remain unchanged until the communications are completed. As nodes begin to travel or as communication patterns change, the routing packet overhead of the DSR automatically scales only to that which is needed to track the routes currently in use.

In DSR when a node wishes to establish a route, it issues a Route Request (RREQ) to all of its neighbors. Each neighbor broadcasts this RREQ, adding its own address in the header of the packet. When the RREQ is received by the destination or by a node with a route to the destination, a Route Reply (RREP) is generated and sent back to the sender along with the addresses accumulated in the RREQ header. The responsibility to assess the status of a route falls to each node in the route. Each node must ensure that packets successfully cross the link to the next node. If the start node does not receive an acknowledgement from the end node of a link on the path, it reports the error back to the source node and leaves it to the source to find and establish a new route. Since this process may consume a lot of bandwidth, DSR provides each node with a route cache to be used aggressively to reduce the number of control messages that must be sent. If a node has a cache entry for the destination when a route request for that destination is received at the node, it will use the cached copy rather than forwarding the request in the network. In addition, it promiscuously listens to other control messages (RREQs and RREPs) for additional routing data to add to its cache. DSR has the advantage in that no routing tables need to be maintained to route a given packet, since the entire route is contained in the packet header. However, tables are used to cache routes and enhance performance. The caching of any initiated or overheard routing data can significantly reduce the number of control messages being sent, thus drastically reducing the overhead.

The disadvantages of DSR are two-fold. DSR is not scalable to large networks. The Internet Draft acknowledges that the protocol assumes the diameter of the network is no greater than 10 hops. Additionally, DSR requires significantly more process resources than most other protocols. In order to obtain routing information, each node must spend much more time processing any control data it receives, even if that node is not the intended recipient. This is the ability of many network interfaces, to operate the network interface in "promiscuous" receive mode, including most current LAN hardware for broadcast media such as wireless. This mode causes the hardware to deliver every received packet to the network driver software without filtering, based on link-layer destination address. The promiscuous mode increases bandwidth utilization of DSR by reducing the number of control messages being sent out, though the use of promiscuous modes may increase the

power consumption of the network interface hardware. Depending on the design of the receiver hardware, and in such cases, DSR can easily be used without the optimizations that depend on the promiscuous receive mode, or can be programmed to only, periodically switch the interface into promiscuous mode. Use of promiscuous receive modes is optional in DSR.

### Hybrid Routing Protocols

Both the proactive and reactive protocols work well for networks with a small number of nodes. As the number of nodes increases, hybrid reactive/proactive protocols are used to achieve higher performance. Hybrid protocols attempt to assimilate the advantages of purely proactive and reactive protocols. The key idea is to use a reactive routing procedure at the global network level while employing a proactive routing procedure in a node's local neighborhood.

Zone Routing Protocol (ZRP) [3] is an example of the hybrid routing protocols. In ZRP, every node has a *zone* around itself, which includes nodes that are  $R$  hops away from that node.  $R$  is called the *zone radius*. ZRP limits the scope of proactive procedure to each node's zone. In this way, ZRP reduces the cost of frequent updates in response to continuously changing network topology by limiting the scope of the updates to the neighborhood of the change. The ZRP route discovery operates as follows. When a source node wants to find a route, it first checks whether the destination is within its zone. If so, the path to the destination is fetched from its table and no further route discovery is required. If the destination is not within the source routing zone, the source broadcasts a route request to its *peripheral nodes*, which are nodes in the border of the node's zone. The peripheral nodes, execute the same algorithm - checking whether the destination is within their zone. If so, a route reply is sent back to the source indicating the route to the destination. If not, peripheral nodes forward the route request to their peripheral nodes, which execute the same procedure.

## 32.3 Low Power Routing Protocols

The main focus of research on routing protocols in MANETs has been the network performance. There have been a handful of studies on power-aware routing protocols for MANETs. Presented below is a review of some of them.

### Minimum Power Routing

Reference [8] proposes a routing algorithm based on minimizing the amount of power (or energy per bit) required to get a packet from source to destination. More precisely, the problem is stated as:

$$Min_{\pi} \left\{ \sum_{(i,j) \in \pi} T_{ij} \right\} \quad (1)$$

where  $T_{ij}$  denotes the power expended for transmitting and receiving between two consecutive nodes  $i$  and  $j$  (a.k.a. cost of link  $(i,j)$ ) in route  $\pi$ .

This link cost can be defined for two cases:

- When the transmit power is fixed.
- When the transmit power is varied dynamically as a function of the distance between the transmitter and intended receiver. Each node chooses the transmission power level for a link so that the signal reaches the receiver node

with the same constant received power. To achieve this, clearly, links with larger distances require a higher transmission power than links with smaller distances.

For the first case all the nodes in the network use a fixed power for all transmissions, which is independent of the link distance. Since the power cost of transmitting and receiving is fixed, then the link cost is fixed and consequently equation (1) results in selecting a path with a minimum number of hops. In fact, assuming lossless links, a path with the minimum number of hops has a minimum number of transmissions and when the transmit power is fixed, then that path will also result in the least total power dissipation.

In general, for a network with 802.11b as Media Access Control (MAC) layer, energy consumption of each operation (receive, transmit, broadcast, discard, etc.) on a packet is given by [9]:

$$E(\text{packet}) = b \times \text{packet\_size} + c \quad (2)$$

where  $b$  and  $c$  are the appropriate coefficients for each operation. Coefficient  $b$  denotes the packet size-dependent energy consumption that depends on distance, wireless channel conditions and so on, whereas  $c$  is a fixed cost that accounts for acquiring the channel and for MAC layer control negotiation.

The link cost is the sum of all the costs incurred by the source and destination nodes. Traffic is classified as *broadcast* and *unicast* (i.e., point-to-point).

For unicast traffic, when receivers are in non-promiscuous mode operation, the energy cost of the link between sender and receiver may be calculated as follows:

$$T_{SD} = E_{S\_send}(\text{unicast\_packet}) + E_{D\_recv}(\text{unicast\_packet})$$

where  $S$  and  $D$  denote the sender and destination of the unicast packet.

In 802.11b, before sending a unicast packet, the source broadcasts a Request-To-Send (RTS) control message, specifying a destination and data packet size (duration of transmission). The destination responds with a Clear-To-Send (CTS) message. If the source does not receive the CTS, it may retransmit the RTS message. Upon receiving the CTS, the source sends the DATA and awaits an acknowledge (ACK) from the receiver. For unicast traffic with non-promiscuous mode operations the energy cost for all non-destination nodes that can hear the packets is nearly zero since non-destination nodes only consume energy to receive the RTS packet. After this step, they will be discarding packets or even turning off their receivers during the ongoing transaction.

For unicast traffic when receivers are in promiscuous mode operation, the link cost between the sender and destination pair may be calculated as follows:

$$T_{SD} = E_{S\_send}(\text{unicast\_packet}) + \sum_{r \in R_s} E_{r\_recv}(\text{unicast\_packet})$$

where  $R_s$  denotes the set of all nodes that can hear source  $S$ , which obviously includes destination  $D$ . Notice that  $T_{S,D}$  represents an *extended* link cost in the sense that it accounts for the receiver energy cost of the neighboring nodes of the source that can hear the packets sent along the link between the source and the intended destination. According to this link

cost function, assuming that all candidate paths have same hop-count, the “best” paths are those that traverse sparse areas of the network where the node density is low.

For broadcast traffic, the sender listens briefly to the channel and sends data if the channel is free. If the channel is busy, the sender waits and retries later. The broadcast cost may be calculated as follows:

$$T_S = E_{S\_send}(broadcast\_packet) + \sum_{r \in R_S} E_{r\_recv}(broadcast\_packet)$$

This is not a link cost, rather it is a node cost which is assigned to sender(s) of broadcast packets. Broadcast and multicast routing algorithms may make use of this node cost to construct power-aware broadcast or multicast routing trees. These categories of routing algorithms will be explained later in this chapter.

The question of how to make use of the variable transmission power level is more involved. Reference [10] proposes a local routing algorithm for this case. The authors assume that the power needed for transmission and reception is a linear function of  $d^\alpha$  where  $d$  is the distance between the two neighboring nodes and  $\alpha$  is a parameter that depends on the physical environment. The authors make use of the GPS position information to transmit packets with the minimum required transmit energy. The key requirement of this technique is that nodes in the MANET know the relative positions of themselves as well as all other nodes. However, this information may not be readily available. In addition, the GPS-based routing algorithm has two drawbacks. One is that the GPS cannot provide useful information about the physical environment (blockages, bit error rates, etc.) to the nodes. The second weakness is that the power dissipation of the GPS is an additional power draw on the battery source of the mobile node.

Reference [11] proposed a Minimum Transmission Energy (MTE) Multi-hop routing algorithm for wireless sensor networks. Assuming a first-order radio model for a wireless sensor node and assuming  $d^n$  energy loss due to channel transmission where  $n$  is between 2 and 4, the paper uses the following equations for calculating energy, sending and receiving  $k$  bit data over a distance  $d$ :

$$E_{Tx}(k, d) = E_{tx\_elec} * k + E_{amp} * k * d^n$$

$$E_{Rx}(k) = E_{rx\_elec} * k$$

where  $E_{tx\_elec}$  and  $E_{rx\_elec}$  are energy dissipated in the transmitter and receiver electronics and  $E_{amp}$  is energy dissipated in the transmit amplifier. If nodes A and B are separated by distance  $D$  (as shown in Figure 32.1), then MTE calculates the optimum number of relaying nodes,  $K_{opt}$ , that is required to send data from A to B with minimum transmission energy as follows:

$$K_{opt} = \left\lfloor \frac{D}{d_{char}} \right\rfloor \text{ or } \left\lceil \frac{D}{d_{char}} \right\rceil$$

where distance  $d_{char}$ , called the *characteristic distance*, is independent of  $D$  and is calculated as:

$$d_{char} = \sqrt[n]{\frac{(E_{tx\_elec} + E_{rx\_elec})}{(n-1)E_{amp}}}$$

**Figure 32.1:** Relaying nodes are inserted between nodes A and B to reduce the energy of sending a packet from A to B.

### **Battery-Cost Lifetime-Aware Routing**

The main disadvantage of the problem formulation of equation (1) is that it always selects the least-power cost routes. As a result, nodes along these least-power cost routes tend to “die” soon by rapidly exhausting their battery energy. This is doubly harmful since the nodes that die early are precisely the ones that are most needed to maintain the network connectivity (and hence increase the useful service life of the network.) Therefore, it may be more advantageous to use a higher power cost route if this routing solution avoids using nodes that have low remaining battery energy. This observation has given rise to a number of “battery-cost lifetime-aware routing” algorithms as described next.

*The min-sum battery cost routing* algorithm [12] minimizes the total cost of the route. More precisely, this algorithm minimizes a summation of the inverse of remaining battery capacities for all nodes on the routing path. One drawback of this algorithm is that it may select a rather short path containing mostly nodes with high remaining battery capacity but also a few nodes with low remaining battery capacity. The cost of such a routing solution may be lower than that of a path with a large number of nodes all having medium level of remaining battery capacity. However, the former routing solution is in general less desirable from the network longevity point of view because such a path will become disconnected as soon as the very first node on that path dies.

*The min-max battery cost routing* algorithm is a modification of the minimum battery cost routing to address the abovementioned weakness. This algorithm attempts to select a route such that has the cost of the most “expensive” link (i.e., one with the minimum remaining battery capacity) on that path is minimum. Thereby, this algorithm results in a more balanced use of the battery capacity of the nodes in the network. One drawback of this algorithm is that since there is no guarantee that paths with the minimum hop-count or with the minimum total power are selected, it can select paths that result in much higher power dissipation in order to send traffic from a source to destination nodes. This feature does actually lead in shorter network lifetime because in essence the average energy consumption per delivered packet of user data has been increased.

A *conditional min-max battery cost routing* algorithm was also proposed in [12]. This algorithm, which is a hybrid of the min-sum and the min-max battery cost routing algorithms, chooses the route with minimal total transmission power if there exists at least one feasible routing solution where all nodes in that route have remaining battery capacities higher than some pre-specified threshold value. However, if there is no such routing solution, then the min-max routing algorithm is employed to select a route.

Several experiments were reported in [12] to evaluate the effect of different battery cost-aware routing algorithms on the network lifetime. According to the reported results, the min-sum battery cost routing exhibits superior results compared to the min-max battery cost routing in terms of the expiration times of the nodes in the network. Conditional min-max routing showed better or worse results compared to either of the first two algorithms depending on how the threshold value was chosen.



*Maximum Residual Packet Capacity (MRPC)* was proposed in [13]. MRPC is conceptually similar to the conditional min-max battery cost routing. However, MRPC identifies the capacity of a node not only by the residual battery capacity, but also by the expected energy spent in reliably forwarding a packet over a specific link. In fact, the objective function of equation (1) is for a path with lossless links, however, for lossy links, the number of retransmissions in each link increases in proportion to the packet error rate of that link. Reference [13] proposed to rewrite the objective function of equation (1) for reliable minimum total transmission power routing on lossy links and ignoring power expended for receiving packets as follows:

$$Min_{\pi} \left\{ \sum_{(i,j) \in \pi} \frac{\rho_{ij}}{1 - e_{ij}} \right\} \quad (3)$$

where  $e_{ij}$  is the packet error rate of  $link(i,j)$  (assuming constant packet size) when the transmit power level of the link is  $\rho_{ij}$ . Notice that equation (3) is for the case of *hop-by-hop* retransmission where sender of each individual link provides reliable forwarding to the next hop by using localized packet retransmissions. Hop-by-hop retransmission may be contrasted to end-to-end retransmission where individual links do not provide link-layer retransmissions, and error recovery is achieved only via retransmissions initiated by the source node. For end-to-end retransmission, equation (3) is modified as follows [14]:

$$Min_{\pi} \left\{ \left( \sum_{(i,j) \in \pi} \rho_{ij} \right) \cdot \prod_{(i,j) \in \pi} \frac{1}{1 - e_{ij}} \right\}$$

Several experiments are reported in [13] to compare the routing method with different battery cost routings and minimum total transmission power routings. According to these results, although the first node dies sooner in the minimum total transmission power routings compared to the battery-cost routing algorithms, the last node dies later in the first case compared to the second case. MRPC, similar to other battery-cost routing algorithms, increases the expiration time of the first node while the death rate of the nodes is as smooth as the minimum total transmission power routing. However, performance of MRPC, like that of the conditional min-max battery cost routing, depends on a threshold value. This threshold value determines exactly when either the min-max battery cost routing or the reliable minimum total transmission power is applied for route selection.

Reference [15] describes a multi-path battery-cost routing algorithm to balance the energy consumption of nodes in a static wireless ad hoc sensor network. The routing has been designed for a network of stationary nodes whose task is to detect events inside a monitoring region. Nodes that detect an event (so-called source nodes) send their measurement data to specific destination(s) (so-called gateway) by using multi-hop routing. The paper proposes a *Maximal Residual Energy Path (MREP)* routing algorithm which has a min-max or min-sum objective function for selecting paths where the cost function for each link is as follows:

$$C_{i,j} = (F_i - \lambda \cdot \rho_{ij})^{-1}$$

where  $C_{ij}$  is the cost of link  $(i,j)$ ,  $F_i$  is the full-charge battery capacity of node  $i$ ,  $\rho_{ij}$  is transmit energy for sending a bit from node  $i$  to node  $j$ , and  $\lambda$  is an augmentation step size. Reference [15] also proposes a Flow Reduction (FR) algorithm. Firstly,

FR finds all possible paths from each source to a single gateway node (single commodity flow) or to several gateway nodes (multi-commodity flow.) We define a commodity as data exchanged between a specific source-destination pair. FR defines longevity of a path  $\pi$ ,  $L_\pi$ , as follows:

$$L_\pi(q) = \text{Min}_{i \in \pi} \tau_i(q)$$

$$\tau_i(q) = \frac{F_i}{\sum_{j \in N_i} \rho_{ij} \cdot \sum_{c \in C} q_{ij}^c} \quad (4)$$

where  $C$  is set of all commodities (i.e., data communications between various source-destination pairs),  $q_{ij}^c$  is the rate at which data is sent from node  $i$  to node  $j$  in commodity  $c$  with transmit power  $\rho_{ij}$ ,  $N_i$  is the set of all nodes that can be reached by node  $i$  with power level  $\rho_{i,j}$  (cf. Figure 32.2),  $\tau_i(q)$  is the lifetime of node  $i$ , and  $L_\pi(q)$  is the longevity of path  $\pi$  under a given flow  $q=\{q_{ij}\}$ . The goal of FR is to divide the traffic flow of a source node to a sink (gateway) node on all paths between that pair of source and sink nodes such that all paths have the same lifetimes. FR tries to achieve this goal by redirecting some flow of each commodity from the shortest path (which has minimum longevity) toward longest path (which has maximum longevity) and this is repeated separately for each commodity in several steps until all paths between a source and sink have the same longevity. By defining network lifetime,  $\tau$ , as the time when the first node dies, reference [15] has shown that the problem of finding maximum lifetime of a sensor network may be formulated as a linear programming problem as follows:

$$\text{Maximize } \tau$$

$$q_{ij}^{(c)} \geq 0, \forall i \in V, \forall j \in N_i, \forall c \in C$$

$$\sum_{j \in N_i} \rho_{ij} \sum_{c \in C} q_{ij}^{(c)} \leq \frac{F_i}{\tau}, \forall i \in V \quad (5)$$

$$\sum_{j: i \in N_j} q_{ji}^{(c)} + Q_i^{(c)} = \sum_{k \in N_i} q_{ik}^{(c)}, \forall i \in V - S^{(c)}, \forall c \in C$$

where  $Q_i^{(c)}$  denotes the information generation rate at source nodes to be sent to destination nodes (or sink nodes)  $S^{(c)}$  for each commodity  $c$ .  $V$  is set of all nodes and  $N_i$  was defined above. This linear programming can be solved in polynomial time. The solution to this linear programming problem provides the optimal network lifetime.

**Figure 32.2:** Node  $i$  sends out incoming traffic plus locally generated traffic toward nodes that can be reached from  $i$  (nodes in set  $N_i$ ).

### Energy Conserving Techniques for Multi-hop Ad Hoc Networks

It is known that an idle receiver listening for packets can consume almost as much as power as one doing active reception. More precisely, idle, receive, and transmit energy cost ratios for the transceiver part of a mobile node are 1, 2, 2.5 as per

reference [16] and 1, 1.2, 1.7 as per reference [17]. Clearly, energy consumed in idle state of the transceiver cannot be ignored. In addition reference [9] has shown that a major source of extraneous energy consumption is from *overhearing* (or eavesdropping). Radios have a relatively large broadcast range. All nodes in that range must receive each packet to determine if it is to be received locally or forwarded to some other node in the network. Although most of these packets are immediately discarded, they cause superfluous energy consumption in the mobile node. Since the network interface may often be idle or simply overhearing data, the energy dissipated at these states can be saved by turning the radio off when it is not in use. In practice, however, this approach is not straightforward: a node must arrange to turn its radio on not only to receive packets addressed to it, but also to participate in any higher level routing and control protocols. The need for power-aware routing protocols is particularly acute for multi-hop ad hoc networks.

**Power-Aware Multiple Access protocol with Signaling (PAMAS [18]):** PAMAS is a MAC-level protocol which avoids overhearing problem by powering off radios in any of the following cases:

- A node powers off if it is overhearing a transmission and does not have a packet to send.
- If at least one neighbor is transmitting and at least one neighbor is receiving a transmission, a node may power off. This is because, even if the node has a packet to transmit, it cannot do so because of fear of interfering with its neighbor reception.
- If all neighbors of a node are transmitting and the node is not a receiver, it powers itself off.

In PAMAS, nodes attempt to capture the communication channel by exchanging RTS/CTS packets. These packets contain duration of data packet transmission. A node can learn about the times that it can be sleeping (or turn off its radio transceiver) by listening to the RTS/CTS exchange. In PAMAS, this exchange takes place over a separate signaling channel. Thus, this exchange does not interfere with ongoing data transmission. It is possible that a new transmission starts when a node is asleep. In such a case, the node does not know about the duration of data transmission. To solve this problem, nodes probe the signaling channel to find out the length of remaining transmission. Although PAMAS avoids the overhearing problem, it does not address the problem of energy consumption when nodes are idle. Solutions to this latter problem are proposed in GAF and Span.

**Geography-informed Energy Conservation for Ad Hoc Routing ([19]):** Geographical Adaptive Fidelity (GAF) employs intelligent node scheduling techniques to conserve the energy. In MANETs, GAF is driven by this observation that when there is significant node redundancy in a MANET, multiple paths will exist between nodes, thus some intermediate nodes can be powered off to conserve energy while still maintaining the network connectivity.

GAF divides the whole area where the nodes are distributed to small virtual grid cells such that every node in each virtual grid cell can communicate with other nodes in that same cell. At any instant of time, exactly one node in each grid is active while all other nodes are in the power saving mode (sleep or discovery). As shown in Figure 32.3 nodes make transitions between *discovery*, *sleep* and *active* states. In the discovery state, which is the initial state, a node identifies all other nodes that are located in the same grid cell by exchanging discovery messages. A node goes to the active state,  $T_d$  seconds after it enters the discovery state. A node stays in the active state for  $T_a$  seconds after which it goes to the discovery state. A node which is in the discovery or active states enters the sleep state when it finds out that some other node in the same grid is

active and will thus handle routing. When transitioning to the sleep state, a node cancels all pending timers and powers down its radio. A node in the sleep state wakes up after an application-dependent sleep time  $T_s$ .

In GAF, nodes are ranked by several rules. An active node has a higher ranking than a node in the discovery state. For nodes that are in the same state, GAF gives higher ranking to nodes with a higher remaining battery capacity. Thus a node with higher energy resource has a greater chance to become active. In this way, GAF achieves its load balancing strategy. When nodes have high mobility, it is possible that an active node moves out of its grid cell and leaves the grid cell with no active node in it. This problem significantly increases the packet drop rate. To avoid this problem, each node uses the GPS information to determine its bearing and velocity, thereby estimates the time that it expects to stay in the current grid, and adds this expected time to the discovery message. Any node that enters the sleep state wakes up after a time equal to the minimum of  $T_s$  and the expected time for the active node to stay in the grid cell.

**Figure 32.3:** The state transition graph in GAF.

Assuming a static network and without accounting for the protocol overhead, the maximum increase in the network lifetime that is achieved by GAF is equal to  $(\frac{n.R^2}{5.A})$  where  $R$  is radio range of each node and  $n$  is total number of nodes distributed in an area  $A$ . In order for nodes in each grid cell to be able to communicate with nodes in the neighboring grid cells, the grid side length cannot be greater than  $\frac{R}{\sqrt{5}}$ . Thus, area  $A$  is divided to  $\frac{A}{(R/\sqrt{5})^2}$  virtual grid cells.

**Topology Maintenance for Energy Efficiency in Ad Hoc networks (Span [17]):** Span builds on the observation that when there is a region of dense nodes, only a small number of these nodes need to be on at any given time in order to forward traffic. Span thereby adaptively elects some nodes as *coordinators* in the network. Coordinators stay awake to maintain connectivity of the network and to route packets in the network. All other nodes go to sleep to save power. These nodes periodically check if they should wake up and become a coordinator. One possible way for some node  $x$  to become a coordinator is that two neighbors of  $x$  cannot communicate with one another directly or through one or at most two coordinators. In addition if node  $x$  has data to send out, it becomes coordinator during its data transfer. When a node decides to become a coordinator, it uses a slotting and damping technique to delay its announcement of the fact. The node picks a random slot and delays its announcement until that slot. The random delay helps keep away from contention when several nodes decide to become coordinator at the same time. The delay function is as follows:

$$delay_i = \left( \left(1 - \frac{R_i}{F_i}\right) + \left(1 - \frac{P_i}{\binom{N_i}{2}}\right) + \zeta \right) \cdot N_i \cdot \mu$$

where  $R_i$  is the remaining energy of the node,  $F_i$  is the full-charge battery capacity,  $P_i$  denotes the number of pairs of neighbors of  $i$  that cannot talk to one another unless through  $i$  itself,  $N_i$  denotes the number of neighbors of  $i$  (i.e., those nodes that can directly be reached from  $i$ .) Recall that  $\binom{N_i}{2}$  gives the number of pairs of neighbors of  $i$ . According to this equation, a node is more likely to pick an earlier time slot to become a coordinator if its ratio of remaining energy to full-charge battery capacity is high (close to 1.) The node is also more likely to pick an earlier time slot if it can help connect a large number of pairs of its neighbors that would be disconnected without its assistance.  $\mu$  is a random number that is picked uniformly in the  $[0,1)$  range whereas  $\delta$  is the link propagation delay. A node switches from the coordinator to a non-coordinator role if every pair of its neighbors can reach each other directly or through one or two other coordinators. However in order to balance the rate of energy consumption over all nodes, a node switches from a coordinator to a non-coordinator after some fixed period of time. In this way, it allows other nodes to become coordinators.

In Span, nodes make all their decisions based on their local information and, unlike GAF, no knowledge of geographical information is required. Nodes find out about their neighbors proactively by broadcasting HELLO messages. These HELLO messages contain the status of the sender (i.e., coordinator or non-coordinator role), a list of its current coordinators, and its current neighbors. The list of the coordinators and neighbors are used by each of the node's neighbors in coordinator election and withdrawal rules that were described above.

### Energy-Aware Multicast Routing Algorithms

The primary goal of the conventional multicast routing protocols and algorithms has been to reduce the route latency since most multicast applications tend to be delay-sensitive audio/video broadcasting. Therefore, most of the multicast routing protocols are designed to construct a multicast tree that minimizes the communication latency. Since the number of hops is a good heuristic metric for capturing this latency, a multicast tree with the minimum number of hops has been favored by most routing protocols ([20][21][22]). We call this tree the Minimum Hop-count Tree (MHT). As has been described, in MANETs, there are two other criteria that make routing design an even more complex task, i.e., mobility and power efficiency. The issue of mobility has extensively been addressed in the literature. In fact, the performance of multicast routing protocols has been evaluated in regard to their robustness to link failure due to the mobility ([20][21][23][24]). However, there has been little work accomplished on the development of a wireless multicast routing protocol in which power is key objective or constraint. More precisely, although there have been some studies on the construction of energy-efficient broadcast and multicast tree in ad hoc networks ([25][26]), most of these works require a global view of the network and cannot be applied in a distributed way where the nodes have only local knowledge.

**Minimum Energy Broadcasting:** The objective of the minimum energy broadcasting is to reach from a specific source to all other nodes in the network by using multi-hop transmission while consuming the minimum total transmission energy and assuming that nodes have variable transmission power. In MANETs, broadcasting takes place by flooding the network

from a specific source. Since the main use of flooding is in route discovery, it is important that flooding is done with the minimum total energy. Minimum energy broadcasting has been shown to be an NP-hard problem. Several heuristic algorithms for solving this problem have been proposed ([25]).

**Energy-Aware Multicast Routing:** The goal of energy-efficient multicast routing is to reach a subset of nodes (one-to-many cast) that we will refer to as multicast receivers, from a multicast source, such that we have maximum longevity of the paths between the source and the receivers. The problem of the energy-aware multicast tree is mathematically defined as follows. Consider a network graph  $G(V,E)$ , when  $V$  is set of nodes (or vertices) and  $E$  is set of edges in graph  $G$ . Let  $R_s$  denote the set of multicast receivers,  $s$  the multicast source, and  $c(u,v)$  the cost of edge  $(u,v)$ . The objective function may be stated as follows:

$$\text{Min } C(M) = \sum_{(u,v) \in M} c(u,v)$$

where  $C(M)$  is the cost of multicast tree,  $M$ , connecting  $s$  to  $R_s$ .

The edge cost function  $c(u,v)$  may represent the transmit power level needed for sending data from  $u$  to  $v$ . In this case, the abovementioned objective function results in a minimum total transmit power multicast tree. In addition,  $c(u,v)$  may be a battery-related cost of node  $u$  if the objective is to extend the lifetime of the network graph  $G$ . Figure 32.4 shows an example of multicast tree. In general, finding a minimum energy multicast tree is equal to finding a minimum Steiner tree that is known to be an NP-hard problem [27]. Two related works on developing heuristic energy-aware multicast (or broadcast) trees are as follows:

- Least-Cost shortest Path Tree (LPT): This is a tree obtained by superimposing all the least cost paths (or shortest paths) between the source and each multicast receiver.
- Broadcast Link-based MST (BLMST): This is a minimum spanning tree where the link cost is set to the transmission energy needed to sustain communication over that link.
- Multicast Incremental Power Tree (MIPT): This tree is obtained from the Broadcast Incremental Power (BIP) tree proposed in [26]. The BIP algorithm consists of the following steps:
  1. For all nodes  $i$  in the tree and all nodes  $j$  not in the tree, evaluate  $\rho'_{ij} = \rho_{ij} - \rho_i$ , where  $\rho_{ij}$  was defined earlier,  $\rho_i$  denotes the power level of node  $i$  (note that  $\rho'_{ij}$  provides the incremental cost associated with adding node  $j$  to the tree.) Initially the tree includes only the source node (i.e., the broadcast initiator node.)
  2. A pair  $(i,j)$  that results in the minimum value of  $\rho'_{ij}$  is chosen and node  $j$  is added to tree.

This procedure is continued until all intended destination nodes are included. The MIPT is generated by pruning the broadcast tree i.e., by eliminating all sub-paths that are not required to reach the multicast receivers.

**The Neighbor Cost Effect in Multicast Routing:** Assume that a multicast tree from the source to several receivers has been constructed. The packet flow is coming out from the source and is terminated at the leaves of the tree where the receivers are located. We will refer to those intermediate nodes of the tree that have more than one child in the tree as

multi-fanout nodes (e.g., node A in Figure 32.4). In MANETs, since the MAC layer does not have the ability of multicasting [9], there are two distinct methods to send out the packets from a multi-fanout node:

- Multiple unicast: The parent node sends unicast packets to every child node in the multicast tree separately,
- Single broadcast: The parent broadcasts the packets to all nodes in its immediate neighborhood (which may include nodes that are not in the multicast tree).

Reference [9] experimentally studied the power-optimal choice between these two methods. According to its results, the multiple unicast method results in much higher power consumption for the sender (parent node in the multicast tree). The following is empirical energy-cost measurement by [9] for broadcast and unicast send/receive packets:

	Unicast	Broadcast
Send ( $\mu\text{W}\cdot\text{sec}/\text{byte} + \mu\text{W}\cdot\text{sec}$ )	$1.9*\text{packet\_size} + 454$	$1.9*\text{packet\_size} + 256$
Receive ( $\mu\text{W}\cdot\text{sec}/\text{byte} + \mu\text{W}\cdot\text{sec}$ )	$0.5*\text{packet\_size} + 356$	$0.5*\text{packet\_size} + 56$

**Figure 32.4:** Neighbor cost effect in wireless networks.

These measurements have been completed on Lucent IEEE 802.11 2 MBPS WAVELAN PC card with 2.4 GHZ Direct Sequence Spread Spectrum.

Based on these results, a single broadcast method in multi-fanout nodes is more energy efficient. However, when using the single broadcast method, all the nodes that are in the radio range of the sender listen to the channel and receive the packet, thereby, unnecessarily consuming power in receiving the packet. As a result, these nodes will find the multiple unicast method to be more beneficial to them from a power dissipation viewpoint. Consequently, one must consider the power consumption cost of all neighbors of nodes that broadcast packets when calculating the cost of a multicast tree, in which multi-fanout nodes use a single broadcast method. This phenomenon, which we will refer to as the *neighbor cost effect*, makes the problem of finding a multicast tree with optimal cost quite complex. Regarding neighbor cost effect, the general objective function of the multicast tree problem is changed as follows:

$$C(M, t) = \sum_{(u,v) \in M} c(u, v) +$$

$$(if \deg(u) \geq 2 \text{ then } \sum_{v \in N_u \wedge j \notin M} c(u, v) \text{ else } 0)$$

where  $\deg(u)$  denotes degree of node  $u$  in multicast tree  $M$  (including incoming and outgoing edges) and  $N_u$  refers to the set of nodes that are in the radio range of node  $u$ .

Another issue concerning the single broadcast method of multi-fanout nodes is that the farthest child from the parent determines the broadcast transmission power of that transmitting node. For example, in Figure 32.4, the transmission power of node A is  $Max(\rho_1, \rho_2)$ . Considering the neighbor cost effect in multi-fanout nodes makes the multicast routing problem even more challenging. Recall that finding a minimum energy-cost multicast tree without considering the neighbor cost effect is equivalent to that of finding a minimum Steiner tree which is NP-hard. As a result, the problem of finding an energy-aware multicast tree with consideration of the neighbor cost effect is also an NP-hard problem.

There are many algorithms for finding a tree with near optimal cost ([28][29]). Although it is possible to modify some of these algorithms to account for the neighbor cost effect at multi-fanout nodes, this approach is ill advised in our context because these algorithms are too complex and require global information about the network connectivity graph in order to be applied. However, we are interested in finding solutions that can be deployed in an ad hoc network where nodes only have local knowledge about themselves and perhaps their neighboring nodes and must do the route discovery in a distributed, ad hoc manner (no global depository of information exists.) Furthermore, in ad hoc networks, the underlying network topology (connectivity graph) changes dynamically due to the mobility and link failure. Hence, ad hoc routing algorithms should be able to update their routes periodically. The routing update cost should be rather low.

## 32.4 Power-Aware Source Routing

### Cost Function

The objective of Power-aware Source Routing (PSR) [30] is to extend the useful service life of a MANET. This is highly desirable in the network since node death leads to a possibility of network disconnectedness, rendering other live nodes unreachable. Power-aware source routing solves the problem of finding a route  $\pi$  at route discovery time  $t$  such that the following cost function is minimized:

$$C(\pi, t) = \sum_{i \in \pi} C_i(t) \quad (6)$$

$$C_i(t) = \rho_i \cdot \left( \frac{F_i}{R_i(t)} \right)^\alpha \quad (7)$$

where  $\rho_i$  is transmit power level of node  $i$ ,  $F_i$  and  $R_i$  are full-charge and remaining battery capacities of node  $i$  at time  $t$  and  $\alpha$  is a positive weighting factor.

PSR uses a graded cost function as explained next. The exponent  $\alpha$  is a discrete function of the ratio of the remaining battery capacity over the full-charge battery capacity. As this ratio decreases and successively becomes less than a specified



set of threshold values,  $\alpha$  increases according to a fixed schedule. In this way, nodes with very low battery capacity contribute a much higher value to the total path cost. In other words, if a path from source to destination has some nodes with a very low residual battery, the cost of the path will be very high, and therefore, PSR will behave similar to the min-max battery cost routing. Figure 32.5 shows how PSR avoids routes, which include node(s) with low remaining energy. Routing path N1-N2-N5-N8 has the minimum hop-count from N1 to N8 and therefore it is selected by DSR. However, this route includes node N2 which has a very low remaining energy capacity, hence, PSR selects another route, N1-N3-N4-N7-N8.

In DSR, because the route selection is done based on a shortest path finding algorithm (i.e., it selects paths with the minimum number of hops), a selected path may become invalid only due to node movements. In contrast, in PSR, both the node mobility and the node energy depletion may cause a path to become invalid. Since the route discovery and route maintenance processes in PSR are slightly more complicated compared to their counterparts in DSR, these two steps will need to be described in detail. Also, since PSR is derived from DSR, the PSR description will often be contrasted with that of DSR.

**Figure 32.5:** PSR avoids routes consisting of nodes with low remaining battery capacity.

### Route Discovery

In DSR, activity begins with the source node flooding the network with RREQ packets when it has data to send. An intermediate node broadcasts the RREQ unless it gets a path to the destination from its cache or it has already broadcast the same RREQ packet. This fact is known from the sequence number of the RREQ and the sender ID. Consequently, intermediate nodes forward only the first received RREQ packet. The destination node only replies to the first arrived RREQ, since that packet usually takes the shortest path.

In PSR, all nodes except the destination calculate their link cost (cf. equation 7) and add it to the path cost in the header of the RREQ packet (cf. equation 6). When an intermediate node receives a RREQ packet, it starts a timer ( $T_r$ ) and keeps the cost in the header of that packet as *mincost*. If additional RREQs arrive with same destination and sequence number, the cost of the newly arrived RREQ packet is compared to the *mincost*. If the new packet has a lower cost, *mincost* is changed to this new value and the new RREQ packet is forwarded. Otherwise, the new RREQ packet is dropped. The destination waits for a threshold ( $T_r$ ) number of seconds after the first RREQ packet arrives. In that time, the destination examines the cost of the route of every arrived RREQ packet. When the timer  $T_r$  expires, the destination node selects the route with minimum cost and replies. Subsequently, it will drop any received RREQ. The reply also contains the cost of the selected path appended to it. Every node that hears this route reply adds this route along with its cost to its route cache table.

Although this scheme may somewhat increase the latency of the data transfer, it results in a significant improvement of network lifetime as will be shown later.

## Route Maintenance

Route maintenance is needed for two reasons:

- *Mobility*: Connections between some nodes on the path are lost due to their movement,
- *Energy Depletion*: The energy resources of some nodes on the path may be depleting too quickly.

In the first case, a new RREQ is sent out and the entry in the route cache corresponding to the node that has moved out of range is purged. In the second case, there are two possible approaches:

*Semi-global Approach*: The source node periodically polls the remaining energy levels of all nodes in the path and purges the corresponding entry in its route cache when the path cost increases by a fixed percentage. Notice that this results in very high overhead because it generates extra traffic.

*Local Approach*: Each intermediate node in the path monitors the decrease in its remaining energy level (hence the increase in its link cost) from the time of route discovery as a result of forwarding packets along this route. When this link cost increase goes beyond a threshold level, the node sends a route error back to the source as if the route was rendered invalid. This route error message forces the source to initiate route discovery again. This decision is only dependent on the remaining battery capacity of the current node, and hence, is a local decision.

PSR adopts the local approach that minimizes the control traffic. Furthermore, it assumes that all transmit power levels ( $\rho_{i,j}$ ) are constant. This enables PSR to separate the effect of mobility from that of energy depletion during route maintenance. More precisely, for each node  $i$  along a path  $\pi$ , we define a “delta cost” function as follows:

$$\Delta C_i(t_a) = C_i(t_a) - C_i(t_d) = \rho_i \cdot \left( \frac{F_i}{R_i(t_a)} \right)^\alpha - \rho_i \cdot \left( \frac{F_i}{R_i(t_d)} \right)^\alpha \quad (8)$$

where  $t_a$  denotes the time instance when this route entry is fetched from the cache table of node  $i$ ,  $t_d$  denotes the time instance at which  $\pi$  was added to the cache table of node  $i$ ,  $C_i(t_a)$  is the fractional cost contributed by node  $i$  to total cost of the path  $\pi$  at time  $t_a$  whereas  $C_i(t_d)$  is the fractional cost contributed by node  $i$  to total cost of the path  $\pi$  at time  $t_d$ .

Assuming that  $\alpha$  remains unchanged from time  $t_a$  to time  $t_d$ , then the condition for invalidating route  $\pi$  from the cache table of node  $i$  is:

$$\frac{\Delta C_i(t_a)}{C(\pi, t_d)} > \delta \quad (9)$$

where  $\delta$  is a user-specified threshold value.

This condition invalidates a path  $\pi$  in the cache table of node  $i$  if the change in the normalized cost of node  $i$  exceeds a threshold  $\delta$ . This metric appears to be a good way of capturing the dynamics of the node usage in MANETs. As the

remaining energy of a node decreases, the cost of the node increases. The node will force new routing decisions in the network by invalidating its own cache entries to various destinations. However, if a path was recently added to the cache table, the node will not force a new decision (route finding step) unless the node's remaining energy is depleted by a certain normalized amount, due to messages passing through that path. The effect of  $\delta$  on the performance of PSR is studied in detail in section 6.

It should be noted that we provision for the reuse of invalidated paths if node  $i$  was the source of the message and wanted to continue to send data via this path as follows. When node  $i$  has data to send to the destination, it looks up its route cache and chooses a route, if such a route can be found in the cache, irrespective of whether the route was invalidated or not. In this way, we avoid redundant route discoveries in the presence of an existing route. The invalidated cache is purged after a fixed time. The invalid entries are analogous to the victim buffer in the cache structure of general purpose processors. However, the same does not hold good for relaying data. If a cache entry is invalidated in a node and that node is asked to relay data/reply to the destination of that cache entry, then the node will send a route error back to the source. This reply will invalidate routing entries for all nodes on the trace path back to the source. The PSR function of intermediate nodes is shown in Figure 32.6 in pseudo code. The function of the destination node is similar to the intermediate node with the exception that it does not need to check for validation of the path when it refers to its cache because it is the end point for each possible path between that itself and the source.

## 32.5 Lifetime Prediction Routing

### Basic Mechanism

Lifetime Prediction Routing (LPR) [31] is an on-demand source routing protocol that uses battery lifetime prediction. The objective of this routing protocol is to extend the service life of MANET with dynamic topology. This protocol favors the path with the maximum remaining lifetime. We represent our objective function as follows:

$$\text{Max}_{\pi} \left\{ L_{\pi}(t) = \text{Min}_{i \in \pi}(\tau_i(t)) \right\} \quad (10)$$

where  $L_{\pi}(t)$  is lifetime of path  $\pi$  and  $\tau_i(t)$  is the predicted lifetime of node  $i$  at time  $t$ .

**Lifetime Prediction:** Each node tries to estimate its battery lifetime based on its past activity. This is achieved using a Simple Moving Average (SMA) predictor by keeping track of the last  $N$  values of residual energy and the corresponding time instances for the last  $W$  packets received/relayed by each mobile node. This information is recorded and stored in each node. We have carefully compared the predicted lifetimes based on the SMA approach to the actual lifetimes for different values of  $W$  and found  $W=10$  to be a good value.

Our motivation in using lifetime prediction is that mobility introduces different dynamics into the network. In [15] the lifetime of a node is a function of residual energy in the node and energy to transmit a bit from the node to its neighbors (cf. equation (4)). This metric works well for static networks for which it was proposed. However, it is very difficult to efficiently and reliably compute this metric when we have mobility since the location of the nodes and their neighbors constantly change.

**Figure 32.6:** Pseudo code for the key operations performed in the intermediate nodes of a path in the PSR.

PSR does not use prediction and only uses the remaining battery capacity. LPR is superior to PSR since LPR not only captures the remaining (residual) battery capacity but also accounts for the rate of energy discharge. This makes the cost function of LPR more accurate. This is true in MANETs since mobility can change the traffic patterns through the node, which thereby affects the rate of depletion of its battery. Also, recent history is a good indicator of the traffic through the node and hence we chose to employ lifetime prediction.

Our approach is a dynamic distributed load balancing approach that avoids power-congested nodes and chooses paths that are lightly loaded. This helps LPR achieve minimum variance in energy levels of different nodes in the network. As an example, consider the scenario shown in Figure 32.7. Here, node F has two flows going through it ( $D \rightarrow F \rightarrow$ ,  $B \rightarrow F \rightarrow$  and  $C \rightarrow F \rightarrow$ ). Now, if A wants to transmit data to E, the shortest path routing will use  $A \rightarrow F \rightarrow E$ . However, LPR will use  $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$  since E is very power-congested (as a result of relaying multiple flows) and the path passing through F will not be selected by LPR.

**Figure 32.7:** LPR avoids power-congested paths.

**Figure 32.8:** LPR avoids paths consisting of nodes with high energy depletion rate.

Figure 32.8 is an example that shows how different policies of DSR, PSR and LPR give different answers with the same scenario. While PSR avoids choosing a path which goes through node N6 because of low remaining energy, the path selected by LPR (N1-N3-N6-N7-N8) includes N6. The reason is that N6 has a low depletion rate and its estimated lifetime is high.

## Route Discovery

Route discovery in LPR is similar to PSR. In LPR, all nodes except the destination calculate their predicted lifetime,  $\tau_i$  (cf. equation (11)) and replace the *minlifetime* in the header with  $\tau_i$  if  $\tau_i$  is lower than the existing *minlifetime* value in the header.

$$\tau_i(t) = \frac{R_{r,i}(t)}{\frac{1}{W-1} \sum_{k=i-W+1}^i r_k(t)} \quad (11)$$

where  $R_{r,i}(t)$  denotes remaining energy at the  $i_{th}$  packet is being sent or relayed through the current node,  $r_k(t)$  is rate of energy depletion of the current node when the  $k_{th}$  packet was sent and is calculated by as the ratio of the difference between residual energies of the nodes for packets  $k-1$  and  $k$  and the difference between arrival times of these two packets and  $W$  is length of the history used for calculating the SMA.

When an intermediate node receives a RREQ packet, it starts a timer ( $Tr$ ) and keeps the min. lifetime in the header of that packet as *minlifetime*. If additional RREQs arrive with the same destination and sequence number, the cost of the newly arrived RREQ packet is compared to the *mincost*. If the new packet has a lower cost, *mincost* is changed to this new value and the new RREQ packet is forwarded. Otherwise, the new RREQ packet is dropped (cf. Figure 32.9).

In LPR, the destination waits for a threshold number ( $Tr$ ) of seconds after the first RREQ packet arrives. During that time, the destination examines the cost of the route of every RREQ packet that arrived. When the timer ( $Tr$ ) expires, the destination node selects the route with the minimum cost and replies. Subsequently, it will drop any received RREQs. The reply also contains the cost of the selected path appended to it. Every node that hears this route reply adds this route along with its cost to its route cache table. Although this scheme can somewhat increase the latency of the data transfer, it results in a significant power savings as will be shown later. A simple example of this process is illustrated in Figure 32.10. Here, the route A-B-C-D is chosen by LPR over the route A-E-D since the path lifetime of the former is in the 500s, which is greater than the latter.

LPR has a route invalidation timer that invalidates old routes. This helps in removing old routes. This also avoids over usage of particular routes in cases of low mobility.

**Figure 32.9:** Pseudo code of functions performed in an intermediate node as it is executing the LPR algorithm.

**Figure 32.10:** The route setup process in LPR.

### Route Expiration

Route maintenance is needed for two reasons:

- Connections between some nodes on the path are lost due to their movement
- Change in the predicted lifetime.

In the first case, a new RREQ is sent out and the entry in the route cache corresponding to the node that moved out of range is purged. The following policy is adopted to tackle the second situation:

Once the route is established, the weakest node in the path (the node with minimum predicted lifetime at path discover time) monitors the decrease in its battery lifetime. When this remaining lifetime decrease goes beyond a threshold level, the node sends a route error back to the destination as if the route was rendered invalid. The destination sends this route error message to the source. This route error message forces the source to initiate route discovery again. This decision is only dependent on the remaining battery capacity of the current node and its discharge rate in the short history, and hence is a local decision. LPR adopts this local approach because this approach minimizes control traffic. Figure 32.11 shows an example of the route expiration process.

More precisely, node  $i$  generates a route error at time  $t$  when the following condition is met:

$$\tau_i(t_0) - \tau_i(t) \geq \delta \quad (7)$$

where  $t$  denotes the current time,  $t_0$  is time of the route discovery, and  $\delta$  is threshold value.

**Figure 32.11:** (a) Node C sends route error to destination node D (b) Node D sends route error to source A to invalidate the whole path.

## 32.6 Quantitative Evaluation of Source Routing Algorithms

### Simulation Setup

We used the event driven simulator ns-2 [32] along with the wireless extensions provided by CMU [33]. The simulation consists of a network of 20 nodes confined in a 1000X1000 m<sup>2</sup> area. Random connections were established using CBR traffic (at 4 packets/second) such that each node has chance to connect to every other node. Packet size was 512 bytes and each simulation was executed for 20000 sec. The initial battery capacity of each node is 100 units. Nodes followed a random waypoint mobility model with a specific max velocity and no pause time. Each packet relayed or transmitted consumes a fixed amount of energy from the battery as given by equation (2);  $a$  and  $b$  are constants.

The key parameters of study are the *network lifetime*, *node lifetime* and *RMS of energy consumption* ( $E_{RMS}$ ) in the network. We vary the speed and radio transmission range and study their effects on these metrics.

### Simulation Results

The network lifetime is defined as the time taken for a fixed percentage of the nodes to die due to energy resource exhaustion. Network lifetime of DSR, PSR and LPR are compared for a given scenario. Here, the speed of each node is 10 m/s and radio transmission range is 125 m. Figure 32.12 shows the time instances at which a certain number of nodes have died when simulating LPR, PSR and DSR. Note that in the Figure 32.12 node death of all 20 nodes is not shown since some nodes are still alive at the end of the simulation. Some of these nodes are, however, rendered unreachable since many of the nodes have exhausted their energy and hence cannot reach other nodes consistently.

As can be seen, the first node in DSR and PSR dies about 20% earlier than in the case of LPR. Similarly, in DSR 5 nodes die approximately 32% earlier than LPR and 27% earlier than LPR in the case of PSR.

Due to the dynamic nature of the path cost function of PSR (and LPR), a discovered path cannot remain valid for a long time. This is because these connections, if maintained for a long period of time, may exhaust the energy of some nodes on that path. However, discovered paths are in the cache and can be accessed whenever they are required in DSR and (as implemented in ns-2) only mobility can invalidate these cache entries. In addition, cache invalidation is very expensive for the network, since the route is reconstructed by flooding the network. This is handled in PSR as described in the next paragraph.

When the path is discovered, every node puts its remaining energy and path cost in the cache entry. Intermediate nodes check for validity of this path by computing the cost difference as in equation (9). Here,  $\delta$  (the threshold) is a metric that decides how often we invalidate the cache. This threshold affects the performance of PSR. If the threshold is very high, we do cache invalidation very rarely, and might end up over-exercising some nodes in the path. If it is very low, the cache invalidation rate is very high and may lead to unnecessary flooding in the network. The effect of varying this threshold is shown in Figure 32.13.

**Figure 32.12:** Number of dead nodes in DSR, PSR and LPR as a function of the elapsed time.

**Figure 32.13:** Effect of the threshold value,  $\delta$ , for the PSR path invalidation step on the network lifetime.

Since LPR outperforms PSR in terms of results in a longer network lifetime, we have selected LPR to compare it with DSR for the rest of simulation.

To increase the lifetime of the network, *the variance of the residual energy of the nodes should be minimized*. Figure 32.12 is not very informative in this regard. A histogram of the snapshots of the energy consumption in each of the nodes at different time instances would be more informative. Figure 32.14 shows what this histogram may look like, at three time instances. Initially, all nodes have zero energy consumption. As time increases variance of energy consumption or remaining energy of nodes increases, but the rate of increasing for LPR is more than DSR. One of the ways to compare such histograms would be to look at the RMS of the remaining energy ( $E_{RMS}$ ) at different time instances. It provides information about the total energy consumed and spread of consumed (residual) energy. Figure 32.15 shows the evaluation of  $E_{RMS}$  as a function of time for DSR and LPR before any node dies out. The effect of mobility on  $E_{RMS}$  can also be seen in this figure. A linear estimation of  $E_{RMS}$  is shown for ease of comparison. As can be seen LPR is always better than DSR in terms of  $E_{RMS}$  value. This graph is in agreement with our expectations. However, as the velocity of node movement increases, the rate of energy consumption in the network goes up. This is expected since higher velocity of movement implies more route discoveries being performed and as a consequence higher energy consumption in the network. Also, as the node mobility increases, the difference between DSR and LPR decreases. This could be attributed to two reasons:

- LPR makes use of the fact that DSR overloads certain nodes and has a big variance between remaining energies of the nodes. As mobility increases, the amount of overhead (control packets for route discovery) increases for both



DSR and LPR. As a consequence, there is less room for LPR to balance the energy consumption among the nodes in the network and extend its network lifetime.

- Because there are more route discoveries, no paths are overused even by DSR. As a consequence, DSR also achieves load balancing to an extent, decreasing the gain seen by LPR.

Packet delivery ratio is defined as the number of delivered data packets to the number of generated data packets in all nodes. Note that the number of generated packets is the “expected” number of generated packets. We generate as many as 200,000 data packets during the simulation. They are generated between random sources and destination pairs at random times. Many of these might not have reached their intended destination because of the lack of existence of a route between the source and destination for various reasons. Also, the network lifetime clearly affects this ratio. If the network was alive for longer time, it implies that more data traffic goes through since we establish random connections throughout the time of simulation.

In Figure 32.16 we can see that for lower velocities of node movement, LPR has a greater ratio of delivered packets. However, as the mobility increases, this ratio goes down. The intuition for why LPR does not perform as well in higher velocities has been presented above.

**Figure 32.14:** Distribution of energy consumed at three different time instances for LPR and DSR.

**Figure 32.15:** Evaluation of  $E_{RMS}$  for different velocities of node movement.

**Figure 32.16:** Packet delivery ratio vs. velocity of node movement.

The transmission range is another parameter that can affect the performance of routing protocols because it changes the connectivity of the network. We changed the transmission range to see the effect of the degree of connectivity on our metric (cf. Figure 32.17). We assume the same transmission power for all nodes in a simulation. The node transmit range was assigned two different values (125, 200 m) for the simulations. We make the following observations based on this figure:

- When the transmission range increases, each node covers more nodes. In other words, when a node sends a unicast or broadcast packet, more nodes will receive packets and they consume power in their receiver. Hence, each transmission has a lot of power overhead for the network. As a result, when the range increases, nodes discharge faster.
- The number of hops per route decreases by increasing the transmission range. Hence, nodes have less participation in relaying packets resulting in lower activity for each node and slower discharge of its battery capacity.

When range increases from 125 to 200, the dominant effect is the first and the charge rate of the nodes increases drastically. Both of those effects reduce the effect of the LPR scheme and as can be seen, the difference between LPR and DSR decreases such that when the range is 200 the difference is not clear. To reduce the cost of the power due to the second effect, one way is to shut down the non-destined nodes in the range of a transmitting node.

**Figure 32.17:** Effect of transmission range on the  $E_{RMS}$  (node velocity is 5 m/s).

In LPR, route discovery process needs more control packets to be propagated in the network since it needs to compare all possible paths between a source and a sink and selects a path with maximum lifetime. To show the overhead of LPR on the network we have measured the ratio of the number of control packets to the number of delivered packets in the network. This normalizes the overhead of the routing protocol to the goodput (number of received packets) in the network. Figure 32.18 shows this ratio for LPR and DSR for different velocities of node movement and for 380 UDP connections. As the velocity of movement increases, routes are valid for a shorter time and more route discoveries are done in the network resulting in more control packets and more difference between LPR and DSR. LPR increases the ratio of the control packet to transmit a packet less than 4%. The increase in the size of a control packet in DSR to that of the LPR is approximately 1/10 and the overhead in energy for sending such a packet increases by approximately 0.4%. Hence, the additional energy overhead of LPR for route discovery is small.

**Figure 32.18:** The ratio of control packets to delivered packets as a function of velocity of node movement for LPR and DSR for 380 UDP connections.

## 32.7 Conclusion

One of the main design constraints in mobile ad hoc networks (MANETs) is that they are energy constrained. Hence, network routing algorithms must be developed to consider energy consumption of the nodes in the network as a primary objective. In MANETs, every node has to perform the functions of a router. So if some nodes die early due to lack of energy so that the network becomes fragmented, then it may not be possible for other nodes in the network to communicate with each other. This chapter presented power-aware source routing and lifetime prediction routing protocols for MANETs whose aim is to maximize the network lifetime (which is typically defined as the duration time after which a fixed percentage of the nodes in the network “die” as a result of energy exhaustion.) This goal of extending the network lifetime was accomplished by finding routing solutions that tend to minimize the variance of the remaining energies of the nodes in the network. Although these power-aware network routing protocols and algorithms tend to create additional control traffic, simulations reported in this chapter show that they improve the network lifetime by more than 20% on average.

## References

1. C. Hedrick, “Routing Information Protocol,” *RFC 1058*, <http://www.faqs.org/rfcs/rfc1058.html>, June 1988.
2. J. Moy, “OSPF version 2.0,” *RFC 2328*, <http://www.faqs.org/rfcs/rfc2328.html>, Apr. 1998.
3. C. E. Perkins, “Ad Hoc Networking,” Addison Wesley, 2001.

4. C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. of ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications*, pp. 234-244 Oct. 1994.
5. Murthy and J.J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications Journal, Special issue on Routing in Mobile Communication Networks*, vol. 1, no. 2, pp. 183-197, 1996.
6. D. B. Johnson, D. A. Maltz, Yih-Chun Hu and Jorjeta G. Jetcheva, "The Dynamic Source Routing for Mobile Ad Hoc Wireless Networks," <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt>, *IETF Internet draft*, Nov. 2001.
7. C. E. Perkins, E. M. Belding-Royer, and S. Das, "Ad Hoc On Demand Distance Vector (AODV) Routing," *IETF Internet draft*, draft-ietf-manet-aodv-12.txt, Nov. 2002, Work in Progress.
8. S. Singh, M. Woo and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad hoc Networks," *Proc. of Mobile Computing and Networking (Mobicom)*, pp. 181-190, 1998.
9. L.M. Freeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," *Proc. of IEEE Infocom*, pp. 1548-1557, Apr. 2001.
10. Stojmenovic and X. Lin, "Power-aware localized routing in wireless networks," *Proc. IEEE Trans. on Parallel and Distributed Systems*. Vol. 12, No. 11, pp. 1122-1133, May 2001.
11. W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", *Proc. of the 33<sup>rd</sup> Annual Hawaii International Conference on System Sciences*, pp. 3005-3014, Jan. 2000.
12. C.K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad hoc Networks," *IEEE Communication Magazine*, pp. 138-147, June 2001.
13. A. Misra and S. Banerjee, "MRPC: Maximizing Network Lifetime for reliable routing in Wireless Environments," *Proc. of IEEE Wireless Communications and Networking Conference*, pp. 800-806, 2002.
14. S. Banerjee and A. Misra, "Minimum Energy Paths for Reliable Communication in Multi-hop Wireless Networks," *Proc. of MobiHoc 2002*, June 2002.
15. J-H Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad Hoc Networks," *Proc. of Infocom*, pp. 22-31, 2001.
16. O. Kasten. Energy Consumption. ETH-Zurich, Swiss, Federal Institute of Technology. Available at: [http://www.inf.ethz.ch/~kasten/research/bathtub/energy\\_consumption.html](http://www.inf.ethz.ch/~kasten/research/bathtub/energy_consumption.html), 2001.
17. B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Proc. of Mobile Computing and Networking (Mobicom)*, pp. 85-96, July 2001.
18. S. Singh and C. Raghavendra, "PAMAS: Power-aware Multiple Access Protocol with Signaling for Ad Hoc Networks," *ACM Computer Communication Review*, 28(3):5-26, July 1998.
19. Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *Proc. of Mobile Computing and Networking (Mobicom)*, pp. 70-84, July 2001.
20. S.-J. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," *IETF Internet draft*, draft-ietf-manet-odmrp-02.txt, Jan. 2000.
21. J. J. Aceves and E. Madruga, "The Core-Assisted Mesh Protocol," *IEEE JSAC*, vol. 17, no. 8, pp. 1380-1394, Aug. 1999.
22. E. Royer and C. Perkins, "Multicast Ad hoc On-Demand Distance Vector (MAODV) Routing," *IETF Internet Draft*, draft-ietf-manet-maodv-00.txt.
23. S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, "A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols," *Proc. of IEEE Infocom*, pp. 565-574, Mar. 2000.
24. M. Gerla, C.-C. Chiang, and L. Zhang, "Tree Multicast Strategies in Mobile, Multi-hop Wireless Networks," *ACM/Kluwer Mobile Networks and Applications*, vol. 4, no. 3, 1999.
25. M. Cagalj, J. Phubaux and C. Enz, "Minimum-Energy Broadcast in All-Wireless Networks: NP-Completeness and Distribution Issues," *Proc. of Mobile Computing and Networking (Mobicom)*, Sep. 2002.
26. J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "On the Construction of energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Proc of Infocom*, pp. 585-594, 2000.
27. A. Goel, and K. Munagala, "Extending Greedy Multicast Routing to Delay Sensitive Applications," *Journal of Algorithmica*, vol. 33, no. 3, pp. 335-352, 2002.
28. M. Parsa, Q. Zhu and J.J. Garcia-Luna-Aceves, "An Iterative Algorithm for Delay-Constrained Minimum-Cost Multicasting," *IEEE/ACM Trans. on Networking*, vol. 6, No.4, pp. 461-474, Aug. 1998.
29. J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably good performance-driven global routing," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 739-752, 1992.

30. M. Maleki, K. Dantu and M. Pedram, "Power -aware Source Routing in mobile ad hoc networks," *Proc. of Int'l Symp. On Low Power Electronics and Design (ISLPED)*, pp. 72-75, Aug. 2002.
31. M. Maleki, K. Dantu and M. Pedram, "Lifetime Prediction Routing in Mobile Ad Hoc Networks," *Proc. IEEE Wireless Communications and Networking Conf.*, Mar. 2003.
32. NS-2 Manual.<http://www.isi.edu/nsnam/ns/doc/index.html>.
33. CMU Monarch Extensions to ns. <http://www.monarch.cs.rice.edu/>.
34. V.D.Park and S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification," *IETF Internet draft*, draft-ietf-manet-tora-spec-01.txt, Aug.1998, Work in progress.