Chapter 6

ENERGY CONSERVATION

Robin Kravets
Department of Computer Science
University of Illinois at Urbana-Champaign
rhk@uiuc.edu

Cigdem Sengul
Department of Computer Science
University of Illinois at Urbana-Champaign
sengul@uiuc.edu

Abstract Energy is a limiting factor in the successful deployment of ad hoc networks since nodes are expected to have little potential for recharging their batteries. In this chapter, we investigate the energy costs of wireless communication and discuss the mechanisms used to reduce these costs for communication in ad hoc networks. We then focus on specific protocols that aim to reduce energy consumption during both active communication and idle periods in communication.

Keywords: Communication-time energy, idle-time energy, power control, topology control, energy-aware routing, suspend/resume scheduling, power management.

Introduction

The limited energy capacity of mobile computing devices has brought energy conservation to the forefront of concerns for enabling mobile communications. This is a particular concern for mobile ad hoc networks where devices are expected to be deployed for long periods of time with limited potential for recharging batteries. Such expectations demand the conservation of energy in all components of the mobile device to support improvements in device lifetime [11] [10] [25] [38] [42] [35]. In wireless networks, there is a direct tradeoff between the amount of data an application sends and the amount of energy consumed by sending that data. Application-level techniques can be used to reduce the amount of data to send, and so the amount of energy consumed. However, once the application decides to send some data, it is up to the network to try to deliver it in an energy-efficient manner. To support energy-efficient communication in ad hoc networks, it is necessary to consider energy consumption at multiple layers in the network protocol stack. At the network layer, intelligent routing protocols can minimize overhead and ensure the use of minimum energy routes [7] [19] [41] [58] [60] [61]. At the medium access control (MAC) layer, techniques can be used to reduce the energy consumed during data transmission and reception [14] [30] [45] [31] [44] [70]. Additionally, an intelligent MAC protocol can turn off the wireless communication device when the node is idle [26] [34] [56] [57] [65] [69] [72] [35].

Communication in ad hoc networks necessarily drains the batteries of the participating nodes, and eventually results in the failure of nodes due to lack of energy. Since the goal of an ad hoc network is to support some desired communication, energy conservation techniques must consider the impact of specific node failures on effective communication in the network. At a high level, achieving the desired communication can be associated with a definition of network lifetime. Current definitions of network lifetime include: 1) the time when the first node failure occurs [5], 2) the fraction of nodes with non-zero energy as a function of time [22] [67] [68], 3) the time it takes the aggregate delivery rate to drop below a threshold [8], or 4) the time to a partition in the network. In the context of any of these definitions, it may also be useful to consider node priority in the definition of lifetime. For example, the network lifetime could be defined as the time the first high priority node fails. In general, one static definition of lifetime does not fit all networks. In this chapter, we do not discuss the impact of the definition of network lifetime or node failures due to depleted batteries on the communication in the network. Instead, we present approaches to energy conservation that minimize energy consumption for communication in ad hoc networks. However, these approaches can be tuned to support the desired communication and the definition of network lifetime as needed by the specific ad hoc network.

Energy conservation can be achieved in one of two ways: saving energy during active communication and saving energy during idle times in the communication. The first targets the techniques used to support communication in an ad hoc network and is typically achieved through the use of energy-efficient MAC and routing protocols. The second focuses on reducing the energy consumed when the node is idle and not participating in communication by placing the node in a low-power state. In this chapter, we first define the costs associated with communication in ad hoc networks and then discuss the use of communication-time and idle-time energy conservation.

Card	Transmit	Receive	Sleep
Cisco Aironet 350	2.25W	1.35W	75mW
Linksys wpc11	1.42W	462mW	297mW
ORiNOCO 11b	1.43W	925mW	45mW
Socket Low Power SDIO	924mW	49.5mW	49.5mW
Mica2 Mote	89.1mW	33mW	$< 1\mu A$

Table 6.1. Transmit, receive and sleep mode energy costs for selected wireless cards.

6.1 Energy Consumption in Ad Hoc Networks

In general there are three components to energy consumption in ad hoc networks. First, energy is consumed during the transmission of individual packets. Second, energy is consumed while forwarding those packets through the network. And finally, energy is consumed by nodes that are idle and not transmitting or forwarding packets. To understand how and when energy is consumed in ad hoc networks, it is necessary to consider these costs for data packets forwarded through the network and for control packets used to maintain the network. To lay the groundwork for discussing energy efficient communication protocols in ad hoc networks, we define these costs for communication and introduce energy-saving mechanisms used by many protocols.

6.1.1 Point-to-Point Communication

The basis for all communication in ad hoc networks is the point-to-point communication between two nodes. At each node, communication impacts energy consumption in two ways. First, the wireless communication device consumes some base energy when it is activated and idle (see Table 6.2. Note that specifications for most current wireless devices do not provide a differentiation between idle and receive costs). Second, the act of transmitting a packet from one node to another consumes energy at both nodes. Transmission energy is determined by the base transmission costs in the wireless card (see Table 6.1) and the transmit power level at the sender (see Table 6.2). Reception energy depends on the base reception costs in the wireless card and the processing costs for reception (see Table 6.1). The amount of time needed for the packet transfer determines the amount of time the card must be active, and so directly determines the energy consumed by the base card costs for both transmission and reception. This time is determined by two factors: the control overhead from packet transmission and the rate at which the packet is transmitted.

The per-packet control overhead is determined by the mechanisms of the medium access control (MAC) protocol. Depending on the chosen protocol, some energy may be consumed due to channel access or contention resolution. For example, in IEEE 802.11 [26], the sender transmits an RTS (ready to send) message to inform the receiver of the sender's intentions. The receiver replies

Card	Transmit Power Levels
Cisco Aironet 350	100, 50, 30, 20, 5 mW
Socket Low Power SDIO	max 25 mW

Table 6.2. Transmit power levels for selected wireless cards with power control capabilities.

Card	Rates
IEEE 802.11 b,g	11, 5.5, 2, 1 Mbps
IEEE 802.11 a,g	54, 48, 36, 24, 18, 12, 9, 6 Mbps
Mica2 Motes	12 Kbps

Table 6.3. Transmission rates for selected wireless card types.

with a CTS (clear to send) message to inform the sender that the channel is available at the receiver. The energy consumed for contention resolution includes the transmission and reception of the two messages. Additionally, the nodes may spend some time waiting until the RTS can be sent and so consume energy listening to the channel. In this chapter, we focus on the use of RTS/CTS-based protocols. While it has been shown that such protocols may not be optimal for throughput [37], there is no widely accepted alternative for communication in mobile ad hoc networks.

Once channel access and contention resolution have determined that a packet may be sent, many wireless network cards provide multiple rates at which the data can be transmitted, which determines the time needed to send the data (See Table 6.3). The specific transmission rate used is determined by a number of factors, including the signal-to-noise ratio (SNR) and the target reliability of the transmission [19] [41] [58] [60]. In general, the signal strength at the receiver, which determines the SNR, varies directly with the sender's transmit power level and varies inversely with the distance between the sender and the receiver. This relationship can be formulated as:

$$ReceiveSignal \propto TransmitPower/Distance^n, \quad (1.1)$$

where the path loss exponent n varies from 2 to 6 [51], although is most commonly used as 2 or 4. For the receiver to correctly receive the packet, the SNR must be over a certain threshold. As long as the receive SNR is maintained above this threshold, the transmit power level at the sender can be reduced, directly reducing energy consumption at the sender. The adaptation of the sender's transmit power level is called *power control* and is the main tool used to conserve energy during active communication. For the remainder of this chapter, we use power level to mean transmit power level.

Finally, energy is consumed to compensate for lost packets, generally via some number of retransmissions of the lost packets. While reliability is generally the domain of the transport layer, the MAC layer in most wireless devices compensates for some packet failure by retransmitting the packet up to some retransmit limit number of times before considering the packet lost. For current energy conserving protocols, this cost is only considered by protocols that aim to avoid low quality channels and so avoid needing to retransmit packets.

6.1.2 End-to-End Communication

End-to-end communication in ad hoc networks is supported by all nodes participating in route maintenance and data forwarding. Therefore, network-wide energy consumption includes any control overhead from routing protocols, including route setup, maintenance and recovery, as well as the impact of the chosen routes on the energy consumed at the intermediate nodes to forward data to the receiver. The choice of a specific route is determined by the metrics used in the routing protocol. Initial protocols use hop count as a primary metric [29] [47], although delay often implicitly impacts route choices [29]. More recent protocols suggest the use of extended metrics such as signal strength [12], stability [63] and load [36] [46], all of which impact performance and so implicitly impact energy consumption [18]. Energy can also be used explicitly to choose routes that minimize energy consumption [54] [64] or avoid nodes with limited energy resources [58] [33]. Additionally, when a route breaks, it is essential to use energy-efficient mechanisms to find a new route, avoiding a reflooding of the network whenever possible. At the network layer, *energy-efficient routing protocols* combine these techniques with power control for additional energy conservation during active communication.

6.1.3 Idle Devices

A wireless communication device consumes energy when it is idle or listening to the channel (See receive costs in Table 6.1). Such idle costs can dominate the energy consumption of a node, especially if there is not much active communication. Idle-time energy conservation can be achieved by suspending the communication device (i.e., placing it in a low-power mode). Low-level management of *device suspension* is generally handled in the MAC layer. Such power-save modes monitor local communication to determine when a device can be suspended (i.e., no immediate communication) and when it should be awake to communicate with its neighbors. While energy is conserved in these power-save modes, there is a limitation placed on the communication capacity of the network since all communication to and from the node is suspended. Higher layer *power management protocols* trade off energy and performance by determining when to transition between power-save mode and standard active mode.

6.1.4 Energy Conservation Approaches

Once all of these costs are understood, two mechanisms affect energy consumption: power control and power management. If these mechanisms are not used wisely, the overall effect could be an increase in energy consumption or reduced communication in the network. The remainder of this chapter is broken into two sections. We first present techniques for communication-time energy conservation, focusing on the impact of power control and energy-efficient routing. We follow this with a presentation of idle-time energy conservation techniques, looking at both low level suspend/resume mechanisms and higher level power management.

6.2 Communication-Time Energy Conservation

The goal of communication-time energy conservation is to reduce the amount of energy used by individual nodes as well as by the aggregation of all nodes to transmit data through the ad hoc network. Two components determine the cost of communication in the network. First, direct node-to-node transmissions consume energy based on the power level of the node, the amount of data sent and the rate at which it is sent. The amount of data is determined by the application and the rate is determined by the characteristics of the communication channel. Although the transmission rate can also be adapted by the sender [23], we do not consider such rate control in this chapter. However, the power level can be controlled by the node to reduce energy consumption. Such *power control* must be performed in a careful manner since it can directly affect the quality and quantity of communication in the network. Second, energy is consumed at every node that forwards data through the network. Such costs can be minimized using *energy-aware routing* protocols. This section first discusses the use of power control and its impact on communication in ad hoc networks. We then present power control protocols and energy-aware routing protocols that aim to minimize energy consumption for communication in the network.

6.2.1 Power Control

Current technology supports power control by enabling the adaptation of power levels at individual nodes in an ad hoc network. The power level directly affects the cost of communication since the power required to transmit between two nodes increases with the distance between the sender and the receiver. Additionally, the power level defines the communication range of the node (i.e., the neighbors with which a node can communicate), and so defines the topology of the network. For devices capable of power control, the power level can be adapted up to a *transmit power level threshold,* as defined by the capabilities of the device (see Table 6.2). This threshold defines the maximum energy cost for communication. Due to the impact on network topology, artificially limiting the power level to a *maximum transmit power level* at individual nodes is called *topology control.* Topology control protocols adapt this maximum within the constraints of the threshold to achieve energy-efficient communication by limiting the maximum cost of a transmission. The impact of power control on communication is twofold. First, adjusting power levels affects channel reservation. Second, power control determines the cost of data transmission.

During channel reservation, the power level directly defines the physical range of communication for a node and the physical area within which channel access control must be performed. Given the shared characteristics of wireless communication channels, any node within transmission range of the receiver can interfere with reception. Similarly, the sender can interfere with reception at any node within its transmission range. Therefore, MAC layer protocols coordinate all nodes within transmission range of both the sender and the receiver. In the context of RTS/CTS-based protocols, the channel is reserved through the transmission of RTS and CTS messages. Any other node that hears these messages backs off, allowing the reserving nodes to communicate undisturbed. The power level at which these control messages are sent defines the area in which other nodes are silenced, and so defines the spatial reuse in the network [20] [24] [37] [62]. Since topology control determines the maximum power level for each node in the network, topology control protocols that minimize power levels increase spatial reuse, reducing contention in the network and reducing energy consumption due to interference and contention.

The use of power control can result in nodes with different maximum power levels. While utilization of heterogeneous power levels increases the potential capacity of the network, it increases the complexity and degrades the effectiveness of the control protocols. Therefore, it is necessary to understand these trade-offs to decide whether to allow heterogeneous power levels or to require all nodes to use the same maximum power level.

In a random uniformly distributed ad hoc network where traffic patterns are optimally assigned and each transmission range is optimally chosen, the maximum achievable throughput is $O(\frac{1}{\sqrt{n}})$ for each node, where n is the number of nodes in the network [21]. When a homogeneous, or common, power level is used (i.e., without optimal heterogeneous power level assignments), the achievable throughput closely approaches this optimum [32]. Therefore, common power can be effective in such networks. However, the results for common power in uniformly distributed networks are not applicable to non-uniformly distributed networks [20]. To maintain connectivity in a network where nodes are clustered, the common power approach converges to higher power levels than the heterogeneous approach, sacrificing spatial reuse and energy.

While heterogeneous power levels can improve spatial reuse, the mechanisms used for channel reservation are compromised, resulting in asymmetric links

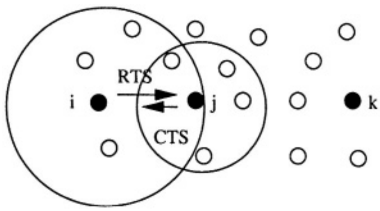

Figure 6.1. Node j's power level is less than node i's and communication is not possible.

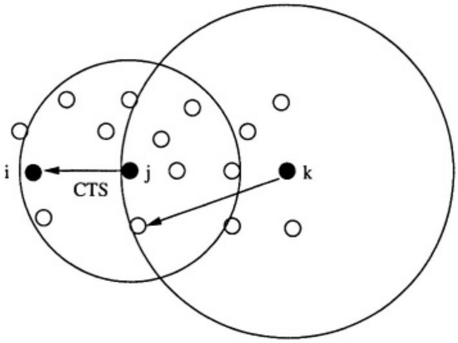

Figure 6.2. Node j's CTS does not silence node k, and so node k can interfere with node j, since node k's power level is higher.

(see Figure 6.1) and in more collisions in the network [30]. For a homogeneous network where all nodes transmit with identical power levels, RTS/CTS-based protocols, such as IEEE 802.11, achieve contention resolution while limiting the occurrence of collisions. However, in a heterogeneous network where each node is capable of transmitting with different power levels, collisions may occur if a low-power node attempts to reserve the channel with an RTS message that is not heard by high-power neighbors that are close enough to disrupt communication [48] (See Figure 6.2). Therefore, control message transmission should use the threshold power level, leaving little potential for additional spatial reuse. PCMA [43] suggests the use of a second channel to transmit a busy tone, allowing senders to monitor the strength of the busy tone signal to dynamically determine a maximum power level that would not interfere with ongoing communication. However, PCMA was designed in the context of single hop wireless networks and it is yet unclear how to apply it to multihop wireless networks. Although channel reservation for nodes with heterogeneous power levels has not yet been solved in the context of ad hoc networks, future protocols may enable better channel reservation. Therefore, we discuss topology control protocols for both homogeneous and heterogeneous networks.

Once the communication range of a node has been defined by the specific topology control protocol, the power level for data communication can be determined on a per-link or even per-packet basis. If the receiver is inside the communication range defined by the specific topology control protocol, energy can be saved by transmitting data at a lower power level determined by the distance between the sender and the receiver and the characteristics of the wireless communication channel [19] [41] [58] [60]. When limited to the transmission of data messages, we call such transmit power control *transmission control.* In the context of RTS/CTS-based protocols, transmission control can easily be used to limit power level adaptation to the transmission of data, leaving control message transmission at the maximum power level [19].

Although reducing the power level only during data transmission directly reduces the transmission energy consumption, it can cause more collisions in the network [30] [48]. If the same power level is used for both control and data messages, nodes that miss the control message exchange still back-off during the data transmission since they sense a busy channel. If the the data is sent at a lower power level, nodes that miss the control message exchange may not sense a busy channel and so could unintentionally interfere with the data transmission. To compensate for these collisions, PCM [30] uses the threshold power level to send the RTS and CTS messages and uses the minimum power level necessary to transmit the ACK. However, to send the DATA, PCM alternates between short transmissions at the threshold and longer transmissions at the minimum power level. These "pulses" at the threshold power level indicate to other nodes that there is active communication and the channel is already reserved. While saving energy by sending most of the data message at a lower power level, PCM does not enable any extra spatial reuse.

Senders can use transmission control with very little overhead. Transmission control can be supported in a fully localized manner since it only needs information about the state of the communication channel between the sender and the receiver. For example, in the context of an RTS/CTS-based protocol, the receiver can return the observed signal strength of the RTS in the CTS packet [27] [1]. The sender can use the received signal strength along with the original power level for the RTS to determine an optimal data power level [19] [41] [58] [60], Energy-aware routing protocols can then use these optimized data transmission costs to find minimum cost routes through the network.

6.2.2 Topology Control

Topology control aims to reduce the maximum power level at individual nodes to minimize energy consumption and maximize spatial reuse while maintaining connectivity in the network. However, aggressive topology control can create a network that is easily partitioned by the loss or failure of one node. Fault tolerance can be improved by requiring the topology control protocol to find a graph where multiple node failures are required to cause a partition. Additionally, the majority of topology control protocols are designed for static networks, limiting their ability to maintain the network topology in the presence of mobility.

Topology control protocols can be divided into two types: *common power* and *heterogeneous power*. Common power protocols find the common maximum power level for all nodes and heterogeneous protocols choose a maximum power level for each node. We first present both common and heterogeneous topology control protocols and then discuss the impact of mobility on all protocols.

Common Power. When all nodes share the same maximum power level, this common power should be chosen as small as possible to limit the maximum energy consumption and to achieve high spatial reuse. A common power that is too high increases the number of neighbors at a given node, which increases the number of nodes that can cause interference at that node, increasing energy consumption and reducing spatial reuse. On the other hand, if the common power is too low, the network may be disconnected, limiting effective communication in the network.

Given a discrete set of power levels $[P_{min}, P_{thresh}]$, if the network is connected when all nodes use the threshold power level (i.e., P_{thresh}), COMPOW [45] finds the smallest common power *P* that ensures the network remains connected. For each power level *P*, *R*(*P*) is the set of nodes that are connected to a distinguished node when all nodes use common power. Thus, *R(P)* is the reachable set for a common power *P*. Since the network is connected at P_{thresh}, $R(P_{thresh})$ is the maximal reachable set. COMPOW finds the minimum power level P_{common} that maintains this maximal reachable set (i.e., $R(P_{thresh}) = R(P_{common})$). To find each *R(P),* COMPOW runs one proactive routing protocol at each power level up to P_{thresh} to populate the neighbor sets for each node at each power level. The result is a minimum common power that achieves connectivity. However, there is no fault tolerance built into COMPOW and the failure of a critical node can partition the network.

If the network is not connected at P_{thresh}, COMPOW finds the minimum power level that maintains connectivity for every connected component of the network. In a network where nodes are clustered, the common power must be chosen to connect the clusters to each other and therefore may converge to a higher power level (see Figure 6.3). The CLUSTERPOW power control protocol [31] addresses this problem by choosing per packet power levels so that intra-cluster communication uses lower common power and only inter-cluster communication uses higher power levels (see Figure 6.3). This use of multiple power levels at the same time to reach different clusters is a step towards heterogeneous power control approaches, which are discussed next.

Heterogeneous Power. Allowing each node to pick its own maximum power level increases spatial reuse in the network and so increases network capacity. Heterogeneous power topology control protocols use local information to determine which links must be part of the network to maintain connectivity and set the power levels to ensure the presence of those links. We discuss four approaches to heterogeneous power topology control: Connected MinMax [50], Enclosure [52], Cone-Based [66] and Local Minimum Spanning Tree [40].

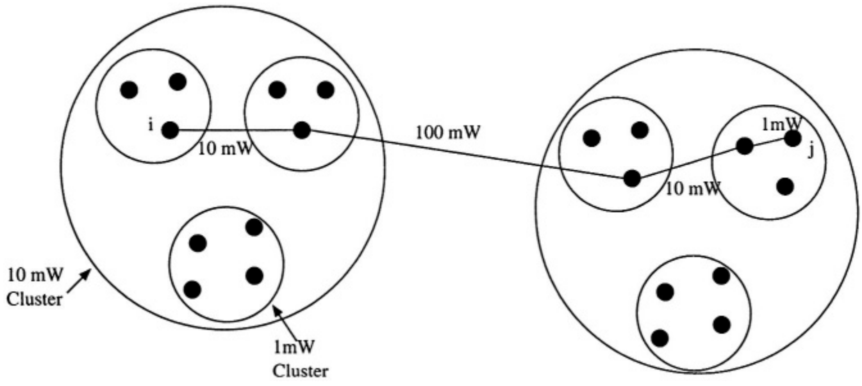

Figure 6.3. COMPOW computes a common power level of 100mW for the network, which shows that a common power level is not appropriate for non-homogeneous networks. With CLUSTERPOW, the network has three clusters corresponding to 1mW, 10mW and 100mW. The 100 mW cluster is the whole network. A 10mW-100mW-10mW-1mW route is used for node i to reach node j.

Connected MinMax Power. In the first approach, the problem of adjusting the power level of individual nodes to create a desired topology is formulated as a constrained optimization problem with connectivity and bi-connectivity as constraints and maximum power level as the optimization objective [50]. The goal of the MinMax Power algorithm is to find the minimum energy needed to maintain a connected (or bi-connected) topology by minimizing the power level of the node with the maximum power level.

The multihop wireless network is represented as $M = (N,L)$, where N is the set of nodes and L is the set of coordinates of node locations. This algorithm requires knowledge of node locations for correct operation, A *least-power* function $\lambda(d)$ defines the minimum power level required to transmit to a distance d, based on current channel conditions. $\lambda(d)$ is defined as:

$$\lambda(d) = \gamma(d(l_i, l_j)) + S, \tag{2.1}$$

where γ is a monotonically increasing propagation function of the geographical distance between the location of node i, l_i, and the location of node j, l_j, and S is the receiver threshold, which determines the threshold signal strength needed for reception. S is assumed to be a known fixed cost for all nodes and, therefore, $\lambda(d)$ does not include the effects of channel fading and shadowing.

The MinMax Power algorithm finds a minimum energy topology that maintains connectivity in the network. For this optimization, a network forms a graph $G = (V, E)$, where V is the set of vertices corresponding to nodes and E is the set of edges corresponding to bi-directional links between nodes based on the maximum power level of the nodes. To improve fault-tolerance, the

MinMax Power algorithm can support more than minimum connectivity. A graph is k-vertex connected if and only if there are k vertex-disjoint routes between every pair of vertices. Therefore, the minimum power level assignment problem to achieve a connected $(k = 1)$ and bi-connected $(k = 2)$ multihop wireless network is formulated as follows [50]:

- *Connected MinMax Power:*

  Given a multihop wireless network $M = (N, L)$ and a least-power function $\lambda(d)$, find a per-node minimal assignment of power levels p such that M is 1-connected and $MAX_{i \in N}(p(i))$ is a minimum (i.e., the maximum power level assigned to any node $i \in N$ is minimized).

- *Bi-connectivity Augmentation with Minimum Power:*

  Given a multihop wireless network $M = (N, L)$, a least-power function $\lambda(d)$ and an initial assignment of per-node power levels p such that M is connected, find the per-node power level increase $\delta(i)$ such that the resulting graph is bi-connected (i.e., given a connected network, find the $p(i) + \delta(i)$ for each node $i \in N$ that makes the network bi-connected).

Given a static network and the location and least power function for all nodes, the above problems can be solved using the following polynomial (greedy) algorithms [50]. To find the power levels that connect the network, the CONNECT algorithm iteratively merges connected components until the whole network is connected. Initially, each node is an individual component. Node pairs are selected in non-decreasing order of their mutual distance. If the nodes are in different components, the power level of each node is increased to reach the other. This is continued until the whole network is one single component. Given a connected network and the power level assignments from the CONNECT algorithm, redundant links can be removed to ensure per-node minimums. The augmentation of a connected network to a bi-connected network is done via the BICONN-AUGMENT algorithm, which determines the bi-connected components in the network via a depth-first search. Node pairs are selected in non-decreasing order of their mutual distance and only joined if they are in different bi-connected components. This is continued until the whole network is bi-connected.

The Connected MinMax Power algorithm achieves the goal of a connected (or bi-connected) network that minimizes energy consumption. However, the algorithm has several limitations. First, both the CONNECT and BICONN-AUGMENT algorithms are centralized and require global information to construct the topology. Second, the construction requires location information, which can be expensive to collect and disseminate. Finally, the propagation model is quite simple and does not reflect the real characteristic of wireless communication such as shadowing or fading.

Enclosure Algorithm. The second approach uses a local optimization algorithm to find per-node maximum power levels that achieve minimum energy consumption [52]. This approach was designed for networks with a specific sink or master node that all other nodes want to communicate with. In this context, the enclosure algorithm focuses on multiple source - single destination communication.

To determine the maximum power level, each node creates a bounded region, called an enclosure, which defines the node's immediate neighborhood. All nodes inside the enclosure are direct neighbors and all nodes outside the enclosure are reached indirectly through neighbors. The enclosure is determined by finding relay regions associated with each neighbor, where indirect communication through neighbors with nodes in those regions is more power-efficient than direct communication. To calculate the enclosures, every node first broadcasts its location information at the threshold power level. A transmitting node i collects these broadcasts to determine the relay region *R* for each potential relay node j as follows:

$$R_{i\to j} \equiv \{(x,y)|P_{i\to j\to(x,y)} < P_{i\to(x,y)}\}, \tag{2.2}$$

where $P_{i\to j\to(x,y)}$ is the power level required to transmit from node i to a node at location (x, y) through the relay node j, and $P_{i\to(x,y)}$ is the power level required to transmit directly from node i to the node at location (x, y). The power consumption $P_{i\to j\to(x,y)}$ includes both transmit and receive power costs:

$$P_{i\to j\to(x,y)} = td^n_{i,j} + td^n_{j,(x,y)} + c, \tag{2.3}$$

where t is the minimum receive threshold at the receiver, d is the distance between i and j, path loss exponent $n \geq 2$ and c is the receive cost at the relay node j.

After determining all relay regions, node i can compose its enclosure and select its direct neighbors as those nodes that are not in any of the relay regions it has calculated. Node i then chooses a maximum power level that maintains connectivity to all of its neighbors. Figure 6.4 illustrates node i and five nodes $\{j, k, l, m, n\}$ it has discovered through the broadcast messages. Node i computes the relay regions for each of these nodes. The three regions computed for nodes j, k and l are illustrated in the figure. The bounded region around i is the enclosure of i. Node m falls in the relay region of node l (Relay Region 3 in the figure) and node n falls in the relay region of j (Relay Region 1 in the figure). Therefore nodes m and n do not belong to the enclosure of node i, which only maintains links to nodes j, k and l as its neighbors.

The *enclosure graph* of the network includes links that belong to all enclosures of all nodes in the network. The minimum power topology, which is a spanning tree with the master site as its root, is a sub-graph of the enclosure graph. A distributed Bellman-Ford shortest path algorithm [9] with

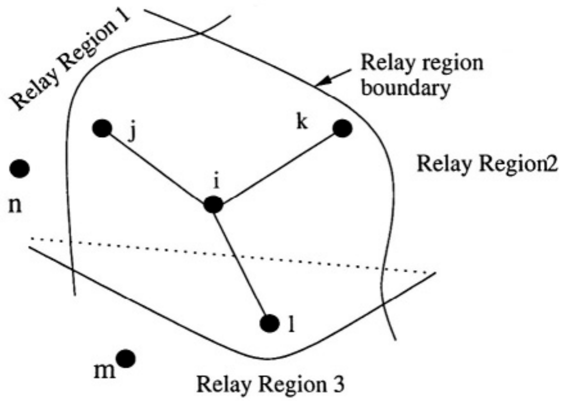

Figure 6.4. Enclosure of node i. Node i computes the relay regions of nodes j, k, and l. Relay Regions 1, 2 and 3 (corresponding to nodes j, k, and l respectively) specify the enclosure of node i. Node i maintains links only to nodes j, k and l. Nodes m and n are not contained in node i's enclosure, and therefore, are not its neighbors.

power consumption as the cost metric is used to find the minimum power paths from each node to the master site. The minimum-power topology is computed by simply removing all links from the enclosure graph that are not part of an energy-efficient shortest path.

The enclosure algorithm builds a strongly connected graph using only local information. It is guaranteed that there exists a path from any node i to any other node j, since the location of node j, (x_j, y_j) falls into the relay region of some neighbor of node i. However, the minimum power topology is computed for one destination and so cannot provide minimum energy communication for arbitrary communication between any two nodes. This approach could be extended to support such arbitrary communication by constructing minimum power topologies for all destinations. Additionally, nodes must be able to acquire their location information, since the algorithm must be able to determine the distances between nodes. Furthermore, the enclosure algorithm uses a fixed channel propagation model based on these distances to compute the relay regions. Such simple channel models do not capture the effect of noise levels at receivers, which may affect nodes differently. The use of this channel model will either be overly optimistic, causing some links to break, or overly pessimistic, causing some nodes to use a higher power level than necessary and wasting energy.

Cone-Based Topology Control. The third approach, cone-based topology control (CBTC), divides the space around each node into "cones" and attempts to create a link to at least one neighbor in every cone [66]. First, each node per-forms *neighbor discovery* by broadcasting discovery messages with increasing power levels until it has reached at least one neighbor in every cone of α degrees. This approach is limited to environments where the node can determine the direction of the sender when receiving a message. If no neighbor is reached in a particular cone even when transmitting at the threshold power level, that cone is left empty. A node's maximum power level is chosen as the minimum that maintains at least one neighbor in each cone, excluding the cones that had no neighbors at any power level. Figure 7.2 illustrates neighbor discovery by the cone-based algorithm for $\alpha = \pi/2$. In the figure, node i sets its power level to P_{max}, which maintains neighbors in cones I, II and IV. Since node j is out of receive threshold range, cone III is empty.

Once the initial power levels have been determined, nodes perform *redundant edge removal,* removing the edges that use more power than an indirect route. Specifically, node i removes an edge to node j if there exists a node k and:

$$p(i,j) + p(j,k) < p(i,k), \tag{2.4}$$

where $p(i,j)$ denotes the power required to send from node i to node j. From a performance point of view, a node should have as few neighbors as possible to reduce the contention and interference in its neighborhood. Therefore, it is desirable to remove some edges even if a direct transmission consumes less power than an indirect transmission. Therefore, Equation 2.4 is extended to:

$$p(i,j) + p(j,k) \leq q \times p(i,k), \tag{2.5}$$

where $q \geq 1$ is a constant that determines the threshold for edge removal even if a direct transmission is more power-efficient (for $q > 1$).

The resulting network constructed by the cone-based algorithm is connected for $\alpha \leq 5\pi/6$, if it is connected when all nodes transmit at the threshold power level [39]. Additionally, if $\alpha \leq 2\pi/3$, asymmetric edges can be removed while still maintaining network connectivity. This is not true for the case when $\alpha > 2\pi/3$, which requires adding a reverse edge for each asymmetric edge to preserve connectivity. We refer the readers to [39] for detailed proofs.

The cone-based algorithm depends only on directional information and does not assume that nodes have location information. However, current techniques for estimating direction without using location information require nodes to be equipped with multiple directional antennas, which can be more complex and consume more energy than a single antenna. Additionally, the CBTC algorithm only supports minimum connectivity and therefore, any node failure may partition the network. A recent CBTC algorithm [3] constructs a k-connected topology if $\alpha = 2\pi/3k$. In such networks, for each $p < k$, failure of p nodes does not disconnect the network.

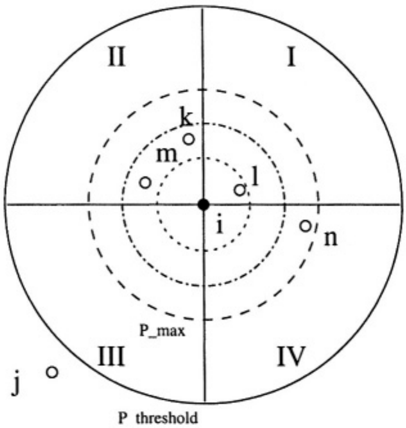

Figure 6.5. Neighbor discovery in the cone-based algorithm, $\alpha = \frac{\pi}{2}$. Node i adjusts its power level to P_{max} to reach all neighbors in all cones. Although, cone III (due to node j being outside the $P_{threshold}$ range), node i does not unnecessarily adjust P_{max} to $P_{threshold}$.

Local Minimum Spanning Tree. The final approach [40] uses purely local information to build a minimum energy spanning tree of the network. Connectivity is only maintained between two nodes if the link between them is part of the spanning tree. The Local Minimum Spanning Tree (LMST) algorithm is composed of three phases: *Information collection, Topology construction* and *Transmit power level determination.*

For the information collection phase, nodes determine their local topology, where local is defined by reachability at the threshold power level. All nodes periodically announce their location by broadcasting HELLO messages at the threshold power level. These HELLO messages are used to define the graph *G(V, E),* where *V* is the set of all nodes, *E* is the set of links, and a link exists between two nodes if they can reach each other at the threshold power level. Each node collects the HELLO messages to determine its *visible neighborhood, NV(G),* where $NV_i(G)$ is defined as the set of nodes that node i received a HELLO message from. Locally, node i maintains the graph $G_i = (V_i, E_i)$, where G_i is the induced subgraph of *G(V, E)* such that $V_i = NV_i$ and E_i is the set of all links in *G* with both endpoints in V_i.

In the topology construction phase, each node i builds a local minimum spanning tree for its visible neighborhood using Prim's algorithm [9]. Specifically, a power efficient minimum spanning tree is built using G_i as the base graph. The weight of each edge is assigned to be the distance between the nodes. Although the weight of an edge in G_i should ideally be the power level required between the nodes, the weight can be approximated as the distance between the nodes since power consumption is an increasing function of distance. At the end of the topology construction phase, node i selects node j as its neighbor if the link to node j is is part of the minimum spanning tree. Finally, each node determines the specific power level needed to reach all of its neighbors by measuring the receive power of the periodically broadcast HELLO messages.

The result of running the LMST algorithm is a directed graph G_0, which may contain unidirectional links if two nodes do not both select each other as neighbors. Figure 6.6 illustrates an example where the topology derived using LMST contains such unidirectional links. There are 6 nodes in $V = \{i, j, k, l, m\}$, where $d(i, j) = d < d_{thresh}$ and $d(i, n) < d_{thresh}$. Nodes k, l and m are outside the threshold transmission range of node i. Therefore, $NV_i = \{j, n\}$. On the other hand, all nodes are in the threshold transmission range of node j and so, $NV_j = \{i, k, l, m, n\}$. Node i maintains links to both nodes j and m as its neighbors since both of these links are part of its local minimum spanning tree (see the solid lines in Figure 6.6). However, node j only keeps node k as its neighbor based on its local minimum spanning tree (see the dashed lines in Figure 6.6). Therefore, the link between node i and node j is unidirectional. However, there exists a route from node j to node i though other nodes.

If the underlying network topology, G, is connected, the unidirectional topology found by the LMST algorithm, G_0, is also connected. In G_0, either two nodes i and j are directly connected, as in G, or there is a minimum energy route from node i to node j. Therefore, G_0 is strongly connected (i.e., it is guaranteed that there exists a route from every node i to every node j). To eliminate the need to deal with unidirectional links, which break some existing routing protocols, a bidirectional topology can be constructed by either deleting all unidirectional edges, G_0^-, or adding reverse edges where unidirectional edges exist, G_0^+ (see Figure 6.7). Both G_0^- and G_0^+ preserve the connectivity of G_0. Since all links in G_0 also exist in G_0^+, it follows that G_0^+ preserves the connectivity of G_0. Similarly, the removal of unidirectional links does not affect the existence of a route between any two nodes. Since each node uses its own local minimum spanning tree to determine its neighbors, a unidirectional link can exist between node i and node j if node j found a more energy efficient route to node i through other nodes in its own visible neighborhood. Since all links are assumed to be symmetric, removal of the unidirectional link simply forces node i to use the energy-efficient route found by node j, maintaining the connectivity in the network. However, there exists a tradeoff between the two choices. While G_0^- is a simpler topology and is more efficient in terms of spatial reuse, G_0^+ provides more routing redundancy.

The LMST algorithm constructs a connected network topology using only local information. On the other hand, each node must be equipped with the ability to gather its location information. Another limitation is that the channel propagation model assumes symmetric channel conditions at both ends of the

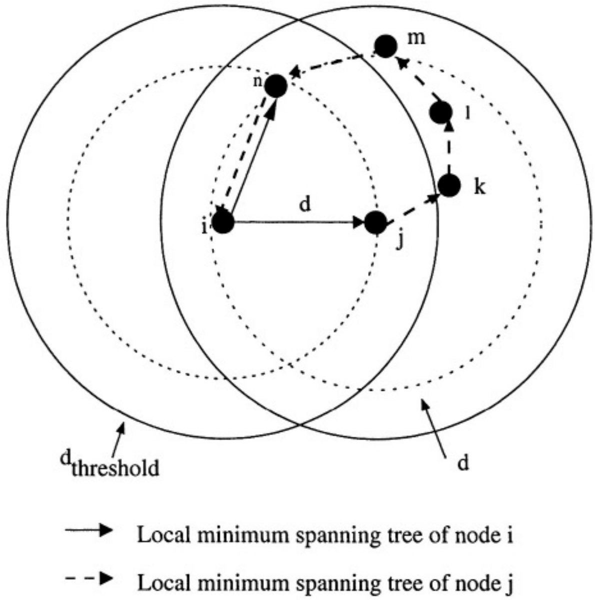

Figure 6.6. An example of unidirectional links using LMST. There are 6 nodes, $V = \{i, j, k, l, m, n\}$. The visible Neighborhood of node i is $NV_i = \{j, n\}$ and the neighbors of node i are nodes j and n. The visible Neighborhood of node j is $NV_j = \{i, k, l, m, n\}$ and only node k is its neighbor. Therefore, $i \rightarrow j$ but $j \nrightarrow i$.

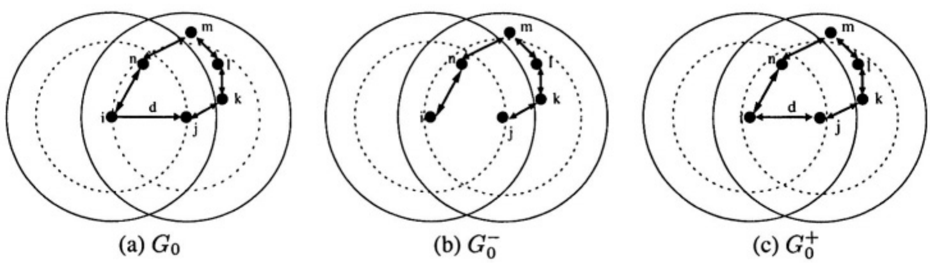

Figure 6.7. Example topologies created by the LMST algorithm

link. Given this assumption, the power level to reach a neighbor can be determined from the receive power of the HELLO messages from the neighbor. However, in practice, the noise levels at different nodes results in asymmetric conditions, limiting the effectiveness of this model.

Mobility. Since most if not all of the nodes in an ad hoc network are expected to be mobile, the topology is expected to change dynamically, implying that a new minimum energy topology must be found. The impact of node movement on a network using minimum energy topology control can be captured by looking at the movement of a single node. If the node moves closer to other nodes, communication can still be supported. However, if the node movement results in a smaller neighborhood for a node i (i.e., node i could now use a lower power level to reach all nodes in its neighborhood), node i may not know about this change and continue using an unnecessarily high power level. If the node moves away (i.e., outside of the current range of the nodes it is communicating with), the network may be partitioned. All of the protocols discussed in this section find minimum energy topologies for a given graph defined by the location of nodes in the ad hoc network. In this section, we discuss how each of the protocols deals with mobility.

COMPOW [45] should recompute the common power for the network each time a node moves to support energy efficient communication and to avoid partitions. To avoid having to determine when these changes occur, COMPOW relies on routing updates generated by the proactive routing protocol to learn about such changes and to determine a new common power. However, proactive protocols are known for their poor performance and lack of convergence in the presence of mobility and therefore, COMPOW can only handle limited mobility.

For the Connected MinMax approach [50] two distributed heuristics are introduced to support mobility. In LINT (Local Information No Topology), each node is configured with three parameters: the desired node degree, d_d, the maximum node degree, d_h and the minimum node degree, d_l. Each node periodically checks its active neighbors and adjusts its power level to stay within these thresholds. In particular, the node reduces its power level if the degree is higher than d_h and increases its power level if the degree is lower than d_l. The magnitude of the power change is a function of d_d and the current degree (i.e., the further apart the current degree and the desired degree, the higher the power change). A significant limitation of LINT is that it may not provide a connected network. LILT (Local Information Link-State Topology) tries to address this problem by exploiting global topology information available from routing protocols. Initially, all nodes transmit with the threshold power level, which results in a maximally connected network. After this initialization, power levels are adjusted based on the desired node degrees, similar to LINT. Additionally, if nodes detect a disconnection in the network via route updates, they increase their power levels to the threshold power level again. However, LILT, similar to LINT, cannot guarantee network connectivity during convergence, especially in a highly mobile environment.

In the Enclosure algorithm [52], each node periodically re-computes its enclosure to find the enclosure graph of the network. The frequency of enclosure computations should be chosen to be frequent enough to accommodate energy cost changes. However, if enclosures are computed too often, unnecessary energy may be consumed. The chosen frequency of enclosure updates must address this trade-off for energy efficient operation of the network. However, the choice of an appropriate update frequency is not addressed in [52].

To deal with mobility, CBTC uses a simple neighbor discovery protocol: each node uses beaconing messages (i.e., HELLO messages) to announce that it is still alive. The beacon includes the node ID and the power level of the beacon. A neighbor is considered to have moved away (or failed) if no beacons are received from this neighbor within a certain time interval, *T*. Each node reconfigures its neighborhood if there are any α-gaps (i.e., at least one of the α-degree cones is empty) or as new nodes are discovered. However, network connectivity is not guaranteed in the presence of frequent topology changes. Additionally, the choice of the time interval for HELLO messages and the time interval *T* is not addressed in [39].

LMST [40] must rebuild the local minimum spanning trees in the presence of mobility. To this end, the interval between two information exchanges (i.e., two HELLO messages) is determined by a probabilistic model. Based on the knowledge of the number of nodes in the network and the maximum node speed, a node computes the probability that a new node joins its neighborhood or that a neighboring node leaves its neighborhood. These two probabilities define the probability that the visible neighborhood of a node changes. A threshold update interval can be chosen to accommodate the expected changes. However, due to its probabilistic nature, LMST may not guarantee connectivity at all times.

In summary, these topology control protocols can only deal with limited mobility and do not guarantee connectivity in the presence of high mobility in the network.

6.2.3 Energy-Aware Routing

Routing protocols for ad hoc networks generally use hop count as the routing metric, which does not necessarily minimize the energy to route a packet [16]. Energy-aware routing addresses this problem by finding energy-efficient routes for communication. At the network layer, routing algorithms should select routes that minimize the total power needed to forward packets through the network, so-called *minimum energy routing*. However, minimum energy routing may not be optimal from the point of view of network lifetime and long-term connectivity, leading to energy depletion of nodes along frequently used routes and causing network partitions. Therefore, routing algorithms should evenly distribute forwarding duties among nodes to prevent any one node from being overused (i.e., *capacity-aware routing*). Hybrid protocols explore the combination of minimum energy routing and capacity-aware routing to achieve energy efficient communication while maintaining network lifetime.

Minimum Energy Routing. The routing metric used by minimum energy routing is the per-hop minimum power level $P(i, j)$ needed for node i to reach node j. The total power level for route r, P_r, is the sum of all power levels $P(i,j)$ along the route:

$$P_r = \sum_{i=0}^{D-1} P(n_i, n_{i+1}), \tag{2.6}$$

where nodes n_0 and n_D are the source and destination, respectively.

Minimum total transmission power routing (MTPR) [54] [15] [60] [61] [58] finds a minimal power route s such that:

$$P_s = \min_{r \in A} P_r, \tag{2.7}$$

where A is the set of all possible routes. Based on a given minimum energy topology that defines the maximum power level for all nodes, MTPR finds the minimum energy routes optimizing the power level for each hop. In contrast, PARO [19] is a minimum energy routing protocol ad hoc networks that discovers minimum energy routes on demand. PARO assumes that all nodes are located within direct transmission range of each other and that a source node initially uses the threshold power level to reach the destination. Each node capable of receiving the packet determines if it should intervene and forward the packet to the destination itself to reduce the energy needed to transmit the packet. Although, PARO is designed for one-hop ad hoc networks, the optimization can be used by any pair of communicating nodes, which allows extending PARO to multi-hop networks.

Given this definition of minimal power routing, both MTPR and PARO favor routes with more hops (i.e., more shorter hops vs. fewer longer hops). Since the power level, and so the transmission energy consumption, depends on distance (proportional to d^n), the energy consumed using many short hops may be less than the energy consumed using fewer longer hops [19] [41] [58] [60]. However, the more nodes involved in routing, the greater the end-to-end delay. Additionally, a route consisting of more hops is likely to be unstable due to the higher probability of the movement or failure of intermediate nodes. Furthermore, both protocols ignore the energy consumed at the relay nodes to receive the packets. Based on these observations, the routes found by MTPR and PARO may not be efficient. To overcome these problems, the energy consumed when receiving the packet should be included into the routing metric [64] [52], which is likely to result in the use of shorter routes. An even more accurate metric should include the total energy consumed in reliably delivering the message to its destination (e.g., the energy cost of link-layer retransmissions) [4]. In particular, it is essential to avoid links with relatively high error rates to reduce the energy consumed to reliably transmit packets.

Capacity-Aware Routing. Assuming all nodes in the network are equally important, no node should be used for routing more often than other nodes.

However, if many minimum energy routes all go though a specific node, the battery of this node is drained quickly and eventually the node dies. Therefore, the remaining battery capacity of a node should be used to define a routing metric that captures the expected lifetime of a node, and so, the lifetime of the network.

Given c_i^t, the battery capacity of node i at time t, the function $f_i(c_i^t)$ captures the cost to forward packets for a node i. This cost can be defined as the inverse of the remaining battery capacity and modeled as [58] [60]:

$$f_i(c_i^t) = \frac{1}{c_i^t}. \tag{2.8}$$

The battery cost metric for route r at time t, R_r, can then be determined as:

$$R_r = \max_{i \in r} f_i(c_i^t). \tag{2.9}$$

Therefore, the desired capacity-aware route s, where A is the set of all possible routes satisfies:

$$R_s = \min\{R_r | r \in A\}. \tag{2.10}$$

It must be noted that the choice of $f_i(c_i^t) = \frac{1}{c_i^t}$ does not consider the effect of the traffic load on the node battery capacity. To this end, *drain rate* is proposed as a metric to measure the *energy dissipation rate* at a given node [33]. The *Minimum Drain Rate (MDR)* algorithm determines the battery cost metric of route r, R_r, as:

$$R_r = \min_{i \in r} \frac{c_i^t}{DrainRate_i}, \tag{2.11}$$

and capacity-aware route s satisfies:

$$R_s = \max\{R_r | r \in A\}. \tag{2.12}$$

Incorporating the battery cost into the routing protocol prevents a node from being overused. However, there is no guarantee that minimum energy routes are found by the routing protocol. Therefore, capacity-aware routing may consume more energy to route traffic, which can reduce the lifetime of the network.

Hybrid Solutions (Minimum Energy/Maximum Capacity). Hybrid solutions try to find minimum energy routes while maximizing the lifetime of the network. To this end, *Conditional Max-Min Battery Capacity Routing (CMM-BCR)* [64] follows minimum energy routing as long as some routes between the source and the destination have sufficient remaining battery capacity (i.e., above a certain threshold). The battery capacity of a route r, R_r^c, is:

$$R_r^c = \min_{i \in r} c_i^t, \tag{2.13}$$

and minimum energy routing is followed as long as:

$$R_r^c \geq \delta \textit{ for any } r \in A. \tag{2.14}$$

If all routes are below the energy threshold δ, capacity-aware routing is used to determine the route to choose. The benefit of such an approach comes from the fact that capacity-aware routing is only used when critical nodes in the network have low battery levels. The efficiency of the CMMBCR depends on the energy threshold δ. However, it is not straightforward how to determine δ.

The *Conditional Minimum Drain Rate (CMDR)* protocol [33] limits route choices for MTPR to routes only containing nodes with a lifetime higher than a given threshold (i.e., $\frac{c_i^t}{DrainRate_i} \geq \gamma$). If no such route exists, CMDR switches to the MDR scheme. To overcome the difficulty of selecting a value for δ in CMMBCR, CMDR uses γ, which is an absolute time value based on the current traffic conditions.

The *max-min* $z \cdot P_{min}$ algorithm [41] minimizes energy consumption and maximizes the minimum residual energy of the nodes. If the minimum energy route has energy consumption P_{min}, routes with higher minimum residual energy can be used as long as the energy consumption is less than $z \cdot P_{min}$. The z-factor, similar to CMDR, is computed based on the minimum lifetime of the nodes.

All three of the above algorithms find minimum energy routes when nodes have sufficient residual energy and switch to capacity-aware routing as the battery capacity of the nodes decreases or the lifetime decreases beyond a predefined threshold. In contrast, the cost metric of a link (i, j) can be chosen to represent both the transmission power cost of the link and the initial and residual energy of node i [7] [60]. Specifically, link cost, $cost_{i,j}$, can be computed as [7]:

$$cost_{i,j} = e_{ij}^{\alpha}(c_i^t)^{-\beta}(c_i^0)^{\theta}, \tag{2.15}$$

where e_{ij} is the energy used to transmit and receive on the link, c_i^t is the current capacity of node i, c_i^0 is the initial capacity of node i and α, β and θ are non-negative weights. The link cost function computed in this fashion emphasizes the energy expenditure term when nodes have high battery capacity. As the residual energy of the nodes decreases, the battery capacity term is more emphasized.

To avoid depletion of nodes along common minimum energy routes, another approach is to occasionally use sub-optimal routes [55]. Basically, possible routes between a source and destination are used with a probability based on the energy metric in Equation 2.15.

6.3 Idle-time Energy Conservation

Effective idle-time energy conservation necessarily spans all layers of the communication protocol stack. Each layer has access to different types of information about the communication in the network, and thus, uses different mechanisms to support energy conservation. MAC layer protocols can save energy by suspending the communication device during short-term idle periods in communication (i.e., operate in a power-save mode). Such fine-grained control requires integrated knowledge of transitions between device suspend and resume in the MAC protocol to insure the communicating nodes are both awake. The delay overhead from waking up a suspended device can negatively impact communication in the network and so power-save modes should not always be used. Power management protocols integrate global information based on topology or traffic characteristics to determine transitions between active mode (i.e., never suspend) and power-save mode.

6.3.1 Communication Device Suspension

When not transmitting, a wireless communication device is continuously listening for incoming transmissions. This listening cost can be quite high since a node must try to receive a packet to see if there is actually a packet being transmitted to it or any other node. If there are currently no transmissions destined for a given node, this listening wastes significant amounts of energy [18] [35]. In wireless communication devices, the cost of listening is only slightly lower than the actual cost of receiving, since listening requires minimal processing overhead compared to receiving [13]. Table 6.1 only lists receive costs since most specifications do no include idle costs. However, measurements show a significant difference between idle and receive costs, depending on the specific device [18].

Listening costs can be reduced by shutting off the device or placing the device in a low-power state when there is no active communication [26] [35]. The low-power state turns off the receiver inside the device, essentially placing the device in a suspended state from which it can be resumed relatively quickly. In general, the suspend costs for most current devices are low enough that the overhead from staying in a suspended state is minimal. In a completely off state, the device consumes no energy. However, the time it takes to resume a device from a completely turned off state can be prohibitively long (i.e., on the order of hundreds of milliseconds) and may even consume extra energy to re-initialize the card. The choice about whether to use suspension or whether to turn off the device must include information about the expected communication patterns for a node. Given that all nodes in an ad hoc network participate in routing and forwarding, we mainly focus on suspending the device.

The goal of any device suspension protocol is to only remain awake when there is active communication for a node and otherwise suspend. In general, active communication is defined to be communication, unicast, multicast or broadcast, that originates from or is destined to a node. However, many ad hoc routing protocols take advantage of the fact that all communication in an ad hoc network is inherently broadcast and snoop on communication in their neighborhood to populate their routing tables. Allowing a node to suspend its device limits the node's ability to snoop on communication between neighboring nodes. To date, there has been little evaluation of the impact of device suspension on route caching.

When a communication device is suspended, the node is effectively cut off from the rest of the network. While it is relatively simple to resume the device when the node has packets to send, the challenge comes from dealing with packets destined to a node with a suspended communication device. Two major problems arise when the destination device is suspended. First, sending nodes must know when the receiving node's communication device is suspended. If the receiver is active, the sender should transmit immediately. If the receiver is suspended, the sender should buffer the packet and wait until the receiver wakes up to transmit it. Packets destined to a suspended device experience delay on the order of the length of time the device spends suspended before it resumes to check for pending transmissions. If the receiver is suspended for too long, the sender's buffers could fill up, eventually causing the packet to be dropped. If the sender thinks a suspended receiver is active, the sender tries to transmit the packets with no success, unnecessarily wasting its own energy and potentially dropping the packets because it thinks the receiver is not in range. Second, the suspended device can only guess when there are packets destined for the node. If it resumes when there are none, it again wastes energy listening to an idle channel. If it waits too long to resume, pending packets are unnecessarily delayed or even dropped.

Since both the sender and receiver must be awake to transmit and receive, it is necessary to ensure an overlap between awake times for nodes with pending communication. We discuss two types of protocols used to manage the suspend/resume cycles for individual nodes: *periodic resume* and *triggered resume.*

Periodic Resume. A simple technique for managing the suspend/resume cycles for nodes is to allow the nodes to suspend most of the time and to resume periodically to check for pending transmissions. If no packets are destined for a node when it checks, the node can again suspend its device. If a node has some packets destined for it, it remains awake until there are no more packets or until the end of the cycle. Nodes can be notified of pending transmissions through an out-of-band control channel (i.e., part of the channel is reserved for

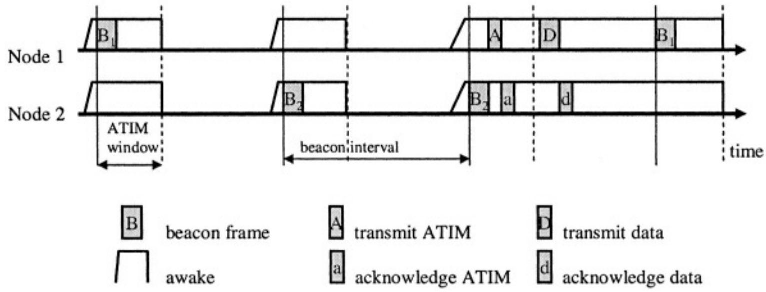

Figure 6.8. IEEE 802.11 Power-save Mode

control messages) [26] [17] [65] or through in-band signaling [71]. The success of a periodic resume approach is to ensure that nodes that want to communicate with each other are awake at the same time to coordinate the notification and ensure that the receiver remains awake to receive the pending transmission. We discuss two approaches to periodic resume: *synchronous* and *asynchronous.*

Synchronous. If the periods of all nodes in the ad hoc network are synchronized, all nodes are guaranteed to have overlapping awake times, and so can easily have overlapping out-of-band channels for notification. Such a synchronized solution is specified in IEEE 802.11 Power Save Mode (PSM) [26], which also provides low-level support for buffering packets for sleeping nodes and synchronizing nodes. In power-save mode, all nodes in the network are synchronized to wake up at the beginning of a beacon interval (see Figure 6.8). To maintain synchronization, beacon messages are sent at the beginning of every beacon interval. Synchronization in single-hop networks only requires the transmission of one beacon per interval. To determine if and when to send a beacon, nodes use a random backoff algorithm. Any node that hears another node's beacon before it sends its own cancels its beacon and synchronizes to that beacon. This algorithm has been proposed for use in multiple-hop wireless networks. While there is the possibility that the algorithm will not converge in dynamic environments, the randomness of the algorithm should enable convergence in many environments.

Broadcast, multicast or unicast packets to a power-saving node are first announced during the period when all nodes are awake. The announcement is done via an ad hoc traffic indication message (ATIM) inside a small interval at the beginning of the beacon interval called the ATIM window. Channel access and contention resolution for communication during the ATIM window follow the same rules as during normal communication. A node that receives a directed ATIM during the ATIM window (i.e., it is the designated receiver) sends an acknowledgement and stays awake for the entire beacon interval waiting for the packet to be transmitted. A node with a pending ATIM that overhears an ATIM acknowledgement from that node need not send another indication since it already knows that the node will remain awake to receive. Broadcast/multicast packets announced in the ATIM window are not directed to a specific node and so are not acknowledged. However, both broadcast and multicast indication messages cause all nodes (or just the nodes in the multicast group) to stay awake for the entire beacon interval. Since the neighbor sets of two nodes is not likely to be exactly the same, nodes send broadcast/multicast indications even if they have already heard one during the current ATIM window.

Immediately after the ATIM window, nodes can transmit buffered broadcast, multicast or unicast packets addressed to nodes that are known to be awake (e.g., nodes that have acknowledged a previously transmitted ATIM). Following the transmission of all announced packets, nodes can continue to transmit packets destined to nodes that are known to be awake for the current beacon interval. The state of a node (i.e., awake or suspended), can be determined by snooping ATIM acknowledgements or by snooping control messages during active communication.

Figure 6.8 shows the interactions between two nodes using IEEE 802.11 PSM in an ad hoc network. During the first two beacon intervals, no packets are pending for either node. The two nodes randomly send beacon messages to maintain synchronization. In the third interval, node 1 has a packet to send to node 2 and so sends a directed ATIM, which is acknowledged by node 2. After the ATIM window has ended, node 1 knows that node 2 is awake and sends the packet using normal channel access rules.

In protocols like IEEE 802.11 PSM that use an out-of-band channel to announce pending transmissions, the throughput of the network is limited to the amount of data that can be announced in the channel. Essentially, if a node cannot send an indication message to wake up its destination, it must buffer its packets until the next beacon interval. If this continues to happen, the node's buffer eventually fills up and packets are dropped.

Asynchronous. Given the expected mobility of nodes in ad hoc networks, clock synchronization may be difficult to maintain. If the clocks are not well synchronized, nodes may not be awake to hear each other's notification messages. However, it is possible to allow nodes to use asynchronous cycles if there is a guarantee that communicating nodes' awake times overlap [17] [28] [65] [71]. The basic idea behind such *asynchronous wake up protocols* is that nodes stay awake longer so there is a guaranteed chance to listen for pending communication from other nodes. Asynchronous approaches can be used with and without notification messages (e.g., ATIMs in IEEE 802.11). All approaches

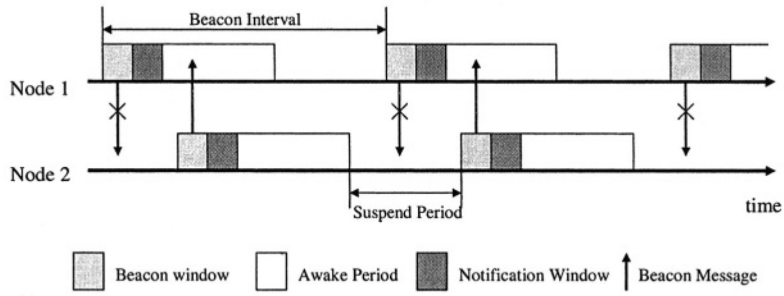

Figure 6.9. Mis-matched Beacon Intervals. Node 2 can never hear the ATIM from node 1.

discussed in this section use beacon messages to inform listening nodes of the beaconing node's presence and of the start of its awake period.

If notification messages are used, the notification window (e.g., the ATIM window in IEEE 802.11 PSM) of the transmitting node must overlap with the awake period of its neighbor node for which it has a packet to transmit. In these approaches [17] [28] [65] [71], each interval is divided into an awake period and a suspend period. Beacon and notification messages are still sent at the beginning of every awake period. To guarantee the overlapping of notification windows and awake periods for nodes with pending communication, awake periods must be at least half of the beacon interval. In other words, every node is awake at least half of the time. However, this change alone does not guarantee overlap. For example, in Figure 6.9, node 2 always misses node 1's beacons. This problem can be fixed by either having the notification window be at the beginning of even periods and at the end of odd periods [28] [65] (see Figure 6.10), or by having two notification windows, one at the beginning of a period and one at the end [17] (see Figure 6.11). Both approaches ensure that at least every other notification window overlaps with a neighbor's awake period. However, requiring a node to remain awake at least half of the time limits the amount of energy that can be saved by these approaches.

The amount of awake time can be reduced in one of three ways. First, a node can remain fully awake once every T beacon intervals [28] (see Figure 6.12). This approach reduces the amount of time a node must remain awake, but increases the delay to transmit to a suspended node. A message could be delayed up to T times the length of the beacon interval before the node can receive a notification message.

The second approach improves on the first by increasing the number of beacon intervals in the cycle but also increasing the number of fully awake intervals [28] [65]. Additionally, the number of beacon messages is reduced by only requiring beacon messages during awake intervals. Essentially, each n^2 intervals, a node stays fully awake $2n-1$ intervals. These $2n-1$ intervals must

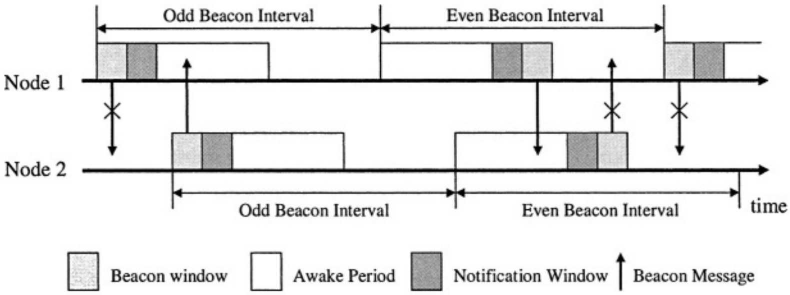

Figure 6.10. Alternating odd and even cycles ensure that all nodes can hear each other's notification messages.

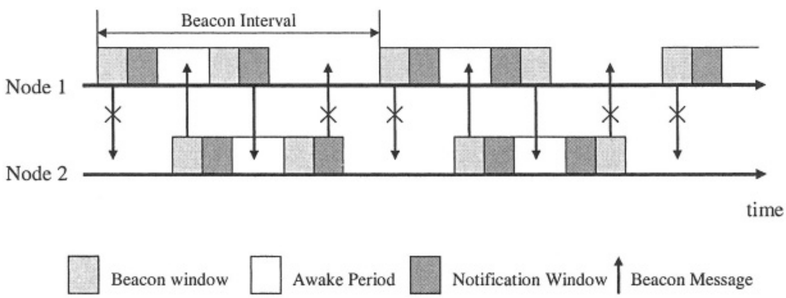

Figure 6.11. Using two notification windows guarantees overlap.

form a quorum, ensuring a non-empty overlap set between any two neighbors. If the n^2 intervals are arranged as a 2-dimensional $n \times n$ array, each host can pick one row and one column of entries as awake intervals (i.e., $2n - 1$) (see Figure 6.13). No matter which row and column are chosen, two nodes are guaranteed to have at least two overlapping awake intervals, guaranteeing the chance to hear each other's notification messages. For example, if $n = 4$, node i chooses row 0 and column 1 and node j chooses row 2 and column 2, they both stay awake during intervals 2 and 9 (see Figure 6.14). This approach improves the average delay to wake up a node since nodes are guaranteed at least two overlapping awake intervals per cycle. However, in the worst case, the overlapping intervals could be right next to each other, resulting in a potential delay up to the length of the whole cycle.

The third approach eliminates the need for notification messages, although still requires beacon messages during awake periods. In this approach, each nodes cycles through a pattern of awake and suspend periods [71]. Every node uses the same pattern, although they may be offset from each other in time. Any pattern of any length can be used as long as it guarantees sufficient overlapping awake intervals between any two nodes. If the number of overlapping intervals is 1, a feasible pattern can be found if the cycle length is a power of a prime number. Other cycle lengths require more overlapping slots. For example, consider a cycle of seven slots to achieve one overlapping slot per pair of nodes. Figure 6.15 shows seven nodes, each with the same pattern, but offset from each other by one slot. This pattern of (awake, awake, suspend, awake, suspend, suspend, suspend) guarantees that every node has at least one overlapping awake interval with every other node, ensuring that each pair of nodes has the opportunity to communicate at least once per cycle. The synchronization between nodes is not required for correctness. We can see in Figure 6.16 that if the nodes' slots are not synchronized, they are still guaranteed to hear each other's beacon messages once per cycle. If one slot is not sufficient to transmit all pending packets, the receiving node listens for the in-band signals in an augmented MAC layer header and remains awake during the next slot to receive the remaining buffered packets. The delay imposed by this approach depends on the number of overlapping awake intervals per cycle.

While asynchronous wake up removes any overhead from maintaining synchronization in the network, a node may spend significantly more time awake than in a synchronous approach. Additionally, all current approaches incur more delay than a synchronous approach. One major drawback of asynchronous wake up is that broadcast support is only provided if the awake periods of all nodes within transmission range of the sender overlap. One approach to solving this problem is to transmit the broadcast message multiple times. However, it is unclear what impact this will have on total energy consumption or on communication in the network. Routing protocols are a particular concern since

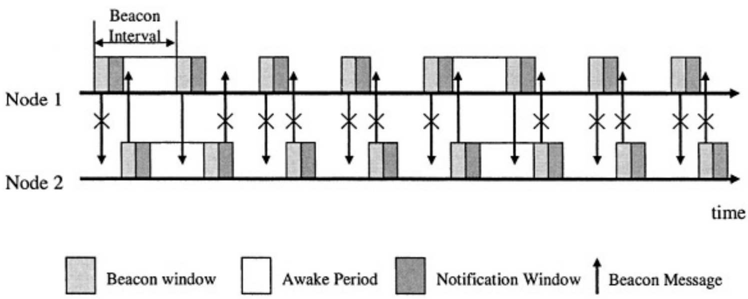

Figure 6.12. Nodes remain awake once every T intervals (T = 4). However, communication is delayed up to T times the length of the beacon interval

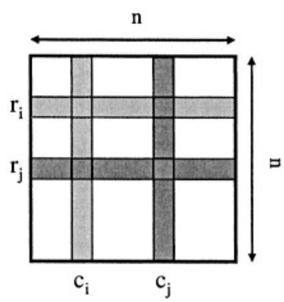

Figure 6.13. Nodes remain awake once $2n - 1$ every n^2 intervals. Nodes each choose one row and one column (i.e., node i chooses row r_i and column c_i and node j chooses row r_j and c

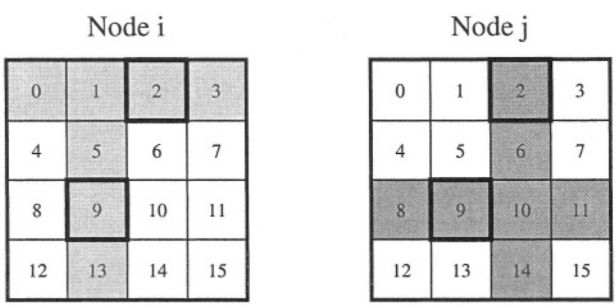

Figure 6.14. Node i chooses row 0 and column 1 and node j chooses row 2 and column 2. Both stay awake during intervals 2 and 9 ($n = 4$).

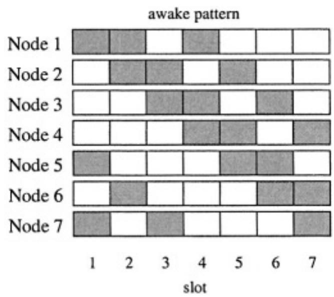

Figure 6.15. Slot allocations determine when each node remains awake. This figure shows an example slot allocation that guarantees at least one overlapping slot between any two nodes.

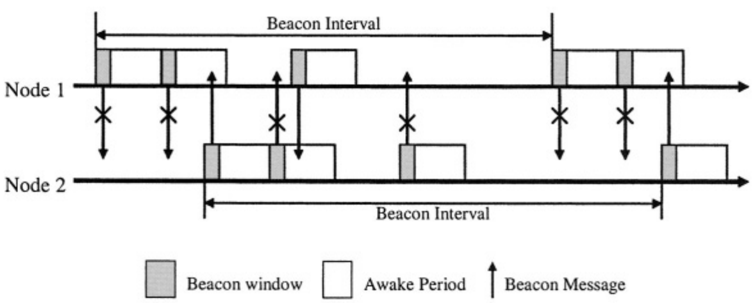

Figure 6.16. Nodes with offset slots are guaranteed to hear each other's beacon messages at least once per cycle..

they typically discover and maintain routes by broadcasting requests through the network.

Triggered Resume. To avoid the need for periodic suspend/resume cycles, a second control channel can be used to tell the receiving node when to wake up, while the main channel is used to transmit the message [1] [49] [53] [56] [57]. To be effective, the control channel must consume less energy than the main channel and also must not interfere with the main channel. For example, transmitting in the 915Mhz [49] [56] or using RFID technology [1] does not interfere with IEEE 802.11, and both consume significantly less energy.

RTS [57] or beacon messages [53] [56] are sent using the control channel to wake up intended receivers, which first respond in the control channel and then turn on their main channel to receive the packet. After the packet transmission has ended, the node turns its radio off in the main channel. Similar to IEEE 802.11, sleeping nodes with traffic destined for them are woken up. However, the decisions about when a node should go back to sleep can be based on local information. The out-of-band signaling used by triggered resume protocols avoids the extra awake time needed by asynchronous periodic resume protocols. Triggered resume protocols like PAMAS [57] and Wake-on-Wireless [56] assume that the radio in the control channel is always active, avoiding the clock synchronization needed by synchronous periodic resume protocols such as IEEE 802.11. Additional savings can be achieved on the control channel using any of the periodic resume approaches. For example, STEM [53] uses a synchronized periodic resume protocol, saving energy in the control channel at the cost of requiring node synchronization.

Triggered resume protocols do not provide mechanisms for indicating the power management state of a node, and so senders assume a receiver is suspended by default. Essentially, the power management state is only maintained on a per-link basis between nodes with active communication. Therefore, it is possible that a sending node experiences the delay from waking up a receiver node, even if the receiver is already awake due to recent communication with a third node.

The limitations of triggered resume protocols come from the complexity of requiring two radios on one node. First, two radios are certainly more expensive than one. Although, if dual radio approaches become popular, the extra cost could become less significant. Second, the characteristics of the wireless communication channel of the two radios can differ significantly in terms of transmission range and tolerance to interference. There is no guarantee that the main channel is usable even if the control radio can successfully transmit to the receiver, causing the receiving node to resume and the sending node to try to transmit needlessly. Similarly, a usable main channel is not accessible if the control channel is not usable, needlessly preventing communication from occurring.

6.3.2 Power Management

In ad hoc networks, suspending a node's communication device can impact communication at multiple layers of the protocol stack. At the MAC layer, uncoordinated suspension between two nodes can prevent the nodes from communicating. At the routing layer, a node that is suspended could be miscategorized as having moved away and so cause a route to break, incurring unnecessary route recovery overhead. Additionally, current device suspension protocols place limitation on the amount of data that can be supported in the network.

If the coordination of suspend and resume states between communicating nodes causes too many packets to be dropped or delayed, the suspension of devices can actually end up consuming more energy [2] [34] [72]. Similarly, if not enough data can be supported in the network, the suspension of devices can limit the effectiveness of the network. Communication in the network can be improved by allowing higher layer decisions about if a device should ever use power-saving techniques. In this context, a node can be in one of two power management modes: active mode and power-save mode. In active mode, a node is awake and may receive at any time. In power-save mode, a node is suspended most of the time and resumes periodically to check for pending transmissions, as described in the previous section. The role of a power management protocol is to determine when a node should transition between active mode and power-save mode.

Packets traversing an ad hoc network can experience difficulties from power management at every hop, impacting the routing protocols and the productivity of the network [72]. The major challenge to the design of a power management protocol for ad hoc networks is that energy conservation usually comes at the cost of degraded performance such as lower throughput or longer delay. Essentially, the goal of power management is to let as many nodes use power-save mode as possible while maintaining effective communication in the network. A naive solution that only considers power savings of individual nodes may turn out to be detrimental to the operation of the whole network.

Power Management and Routing. The particular decisions about when a node should be in a power-save mode affect the discovery of routes as well as the end-to-end delay of packets. Similar to ad hoc routing protocols, power management schemes range from proactive to reactive. The extreme of proactive can be defined as always-on (i.e., all nodes are in active mode all the time) and the extreme of reactive can be defined as always-off (i.e., all nodes are in power-save mode all the time). Given the dynamic nature of ad hoc networks, there must be a balance between proactiveness, which generally provides more efficient communication, and reactiveness, which generally provides better power saving. In this space, we discuss three approaches to using power management in ad hoc networks: *reactive, proactive,* and *on-demand.*

Reactive Power Management. A pure power saving approach (i.e., always-off) can be considered as the most reactive approach to power management. However, a network that relies solely on MAC layer power management such as IEEE 802.11 can be highly inefficient even though some communication is still possible [72]. In an always-off network, all nodes must be woken up before any communication can occur, causing increased delay for both control (e.g., route request or route reply) and data packets. Additionally, all transmissions must be announced (e.g., via an ATIM). If the resources for announcement (e.g., the ATIM window size), cannot support the load in the network, queues fill up and packets get dropped. In a lightly loaded network, an always-off approach can generally support the traffic with little or no drops, although there is still an increased delay. However, in a heavily loaded network, the announcements become a bottleneck and little or no effective communication occurs.

Proactive Power Management. A proactive approach to power management provides some persistent maintenance of the network to support effective communication. Since routing protocols operate at the network layer, proactive power management schemes can take advantage of topological information to ensure that a specific set of nodes stays awake to provide complete connectivity for routing in the ad hoc network [5] [6] [8] [22] [67] [68]. We call this type of approach *topology management.* This differs from topology control, since topology control determines the topology for all nodes while topology management determines which nodes participate in routing in the network.

One approach to topology management is to create a connected dominating set (CDS), where all nodes are either a member of the CDS or a direct neighbor of one of the members [59] (see Figure 6.17). In general CDS-based routing, nodes in the CDS serve as the "routing backbone" and all packets are routed through the backbone. In a CDS-based power management protocol, all nodes on the CDS remain active all the time to maintain global connectivity (e.g., GAF [68] and Span [8]). All other nodes can choose to use power-save mode or even turn off completely. GAF creates a virtual grid and chooses one node in every grid location to be part of the backbone and remain awake (see Figure 6.18). All other nodes turn completely off. Span takes a slightly different approach and uses local message exchanges to allow a node to determine the effect on its neighbors if it stays awake or uses a low-power mode like IEEE 802.11 PSM. Both Span and GAF assume that sources and destinations are separated from pure forwarding nodes. In the case of mixed source/destination/forwarding nodes scenarios, the specification of both protocols is incomplete. Neither

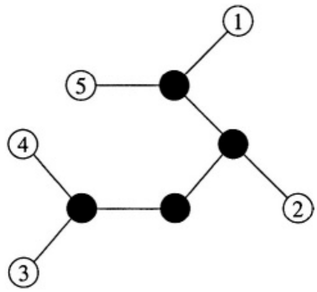

Figure 6.17. Example Connected Dominating Set. The black nodes form the CDS. Nodes 1-5 are all only one hop away from a node in the CDS.

protocol has a mechanism for signaling the data sink for incoming transmissions. In Span, it is unclear whether the election of coordinators should consider the fact that some nodes may be required to be turned on as data sources or destinations.

By taking advantage of route redundancy in dense ad hoc networks, topology management approaches save energy by turning off devices that are not required for global network connectivity. The challenge to topology management comes from the need to maintain the CDS, generally through local broadcast messages that may consume a significant amount of energy [18], especially since broadcast messages wake up all nodes for some amount of time. Additionally, the nodes chosen to participate in the CDS are periodically rotated to prevent any one node from having its battery depleted. This rotation essentially results in the formation of a new CDS, resulting in unnecessary overhead if the CDS does not change. The final limitation to these approaches comes from the fact that regardless of whether or not traffic is present in the network, all the backbone nodes must be active all the time. Essentially, even if there is no traffic in the network, some nodes are still active and consuming significant amounts of energy.

On-Demand Power Management. In response to the limitations of both reactive and proactive power management, on-demand power management eliminates the need to maintain any nodes in active mode if there is no traffic in the network by tying power management decisions to information about which nodes are used for routing in the ad hoc network [72]. In on-demand power management, all nodes are treated equal, eliminating the need to know which nodes are sources and destinations. All nodes are initially in power-save mode. Upon reception of packets, a node starts a keep-alive timer and switches to active mode. Upon expiration of the keep-alive timer, a node switches from active mode to power-save mode. The goal is to have nodes that are actively

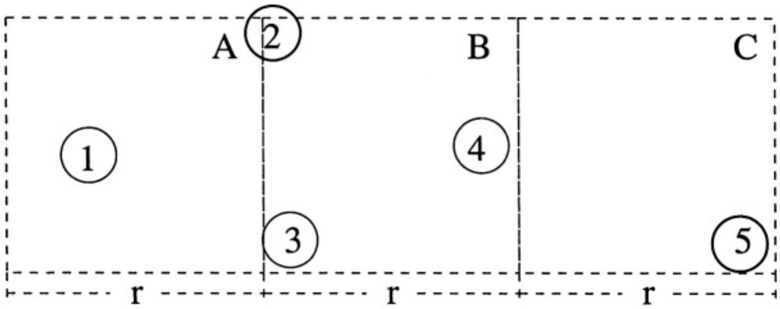

Figure 6.18. GAF's virtual grid. One node in each grid location remains awake to create a connected dominating set.

forwarding packets stay in active mode, while nodes that are not involved in packet forwarding may go into power-save mode. The key idea of on-demand power management is that transitions from power-save mode to active mode are triggered by communication events such as routing control packets or data packets and transitions from active mode to power save mode are determined by a soft-state timer.

In an ad hoc network, if a route is going to be used, the nodes along that route should be awake to not cause unnecessary delay for packet transmissions. If a route is not going to be used, the nodes should be allowed to use power-save mode. During the lifetime of the network, different packets indicate different levels of "commitment" to using a route. Knowledge of the semantics of such messages can help make better power management decisions. On one end, most control messages (e.g., link state in table-driven ad hoc routing protocols, location updates in geographical routing, route request messages in on-demand routing protocols, etc.) are flooded throughout the network and provide poor hints for the routing of data. Such control messages should not trigger a node to stay in active mode. On the other end, data packets are usually bound to a route on relatively large time scales. Therefore, data packets are a good hint for guiding power management decisions. For data packets, nodes should stay active on the order of packet inter-arrival times to ensure that no node along the route goes into power-save mode during active communication. There are also some control messages, such as route reply messages in on-demand routing protocols and query messages in sensor networks, that provide a strong indication that subsequent packets will follow this route. Therefore, such messages should trigger a node to switch to active mode. The time scale for such a transition should be on the order of the end-to-end delay from source to destination so the node does not transition back to power-save mode before the first data packet arrives.

The improvement in energy consumption comes at an increase in the initial delay of packets in a newly established route. Essentially, if all nodes along the route are asleep, they must all be woken up, incurring delay on the order of the length of the route times the time to wake up a node. However, in an active network, many nodes are expected to be awake. On-demand power management implicitly finds routes with more awake nodes, since those routes have shorter delays. Since on-demand power management favors awake nodes, it should be coupled with capacity-aware routing to support load balancing.

6.4 Conclusion

Energy conservation in ad hoc networks is a relatively new field of research. In this chapter, we have presented some of the recent proposals and specifications for achieving that goal. It is clear that there is still room for new approaches that tackle this extremely complex problem of balancing energy conservation with communication quality in dynamic ad hoc networks.

Acknowledgments

The authors wish to thank the many people who helped us bring this chapter together. We would like to specially thank Rong Zheng for her insights into energy conservation in ad hoc networks and Rob Kooper for making it all look great. Additional thanks go out to the members of the Mobius Group in the Computer Science Department at the University of Illinois, Urbana-Champaign.

References

[1] S. Agarwal, S. V. Krishnamurthy, R. H. Katz, and S. K. Dao. Distributed power control in ad-hoc wireless networks. In *IEEE Symposium on Personal Indoor and Mobile Radio Communication (PIMRC),* 2001.

[2] M. Anand, E. B. Nightingale, and J. Flinn. Self-tuning wireless network power management. In *9th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2003.

[3] M. Bahramgiri, M. T. Hajiaghayi, and V. S. Mirrokni. Fault-tolerant and 3-dimensional topology control in wireless multi-hop networks. In *International Conference on Computer Communications and Networks (ICCCN'02),* 2002.

[4] S. Banerjee and A. Misra. Minimum energy paths for reliable communication in multi-hop wireless networks. In *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc),* 2002.

[5] L. Bao and J. J. Garcia-Luna-Aceves. Topology management in ad hoc networks. In *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc),* 2003.

[6] A. Cerpa and D. Estrin. ASCENT: Adaptive self-configuring sensor networks topologies. In *IEEE INFOCOM,* 2002.

[7] J.-H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *IEEE INFOCOM,* 2000.

[8] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *7th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2001.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press/McGraw-Hill, second edition, 2001.

[10] F. Douglis, P. Krishnan, and B. N. Bershad. Adaptive disk spindown policies for mobile computers. In *Second USENIX Symposium on Mobile and Location Independent Computing,* 1995.

[11] F. Douglis, P. Krishnan, and B. Marsh. Thwarting the power-hungry disk. In *USENIX Symposium,* 1994.

[12] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi, Signal stability based adaptive routing (SSA)for ad hoc mobile networks. *IEEE Personal Communications,* 4(1):36–45, February 1997.

[13] J.-P. Ebert, B. Burns, and A. Wolisz. A trace-based aprroach for determining the energy consumption of a WLAN network interface. In *European Wireless 2002,* 2002.

[14] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad-hoc networks. In *IEEE INFOCOM,* 2002.

[15] T. A. ElBatt, S. V. Krishnamurthy, D. Connors, and S. Dao. Power management for throughput enhancement in wireless ad-hoc networks. In *IEEE International Conference on Communications (ICC),* 2000.

[16] L. M. Feeney. An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks. *Mobile Networks and Applications,* 6(3):239–249, June 2001.

[17] L. M. Feeney. A QoS aware power save protocol for wireless ad hoc networks. In *Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net),* 2002.

[18] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *IEEE INFOCOM,* 2001.

[19] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian. PARO: Supporting dynamic power controlled routing in wireless ad hoc networks. *Wireless Networks,* 9(5):443–460, September 2003.

[20] P. Gupta and P. R. Kumar. *Stochastic Analysis, Control, Optimization and Applications,* chapter Critical Power for Asymptotic Connectivity in Wireless Networks, pages 547–566. Birkhauser Boston, 1998.

[21] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory,* 46(2):388–404, March 2000.

[22] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Hawaii International Conference on System Sciences (HICSS),* 2000.

[23] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *7th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2001.

[24] T.-C. Hou and V. Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications,* 34(1):38–44, January 1986.

[25] C.-H. Hwang and A. C.-H. Wu. A predictive system shutdown for energy saving of event-driven computation. In *IEEE/ACM International Conference on Computer Aided Design,* 1997.

[26] IEEE 802 LAN/MAN Standards Committee. Wireless LAN medium access control MAC and physical layer (PHY) specifications. IEEE Standard 802.11, 1999.

[27] IEEE 802 LAN/MAN Standards Committee. Wireless LAN medium access control MAC and physical layer (PHY) specifications: Spectrum and transmit power management extensions in the 5ghz band in europe. Draft Supplement to IEEE Standard 802.11 1999 Edition, 2002.

[28] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai. Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks. In *International Conference on Parallel Processing (ICPP),* 2003.

[29] D. B. Johnson and D. A. Maltz. *Mobile Computing,* chapter Dynamic source routing in ad hoc wireless networks, pages 153–181. Kluwer Academic Publishers, February 1996.

[30] E.-S. Jung and N. H. Vaidya. A power control MAC protocol for ad hoc networks. In *8th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2002.

[31] V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *IEEE INFOCOM,* 2003.

[32] V. Kawadia, S. Narayanaswamy, R. Rozovsky, R. S. Sreenivas, and P. R. Kumar. Protocols for media access control and power control in wireless networks. In *IEEE Conference on Decision and Control,* 2001.

[33] D. Kim, J.J. Garcia-Luna-Aceves, K. Obraczka, J.-C. Cano, and P. Manzoni. Routing mechanisms for mobile ad hoc networks based on the energy drain rate. *IEEE Transactions on Mobile Computing,* 2(2): 161–173, April-June 2003.

[34] R. Krashinsky and H. Balakrishnan. Minimizing energy for wireless web access with bounded slowdown. In *8th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2002.

[35] R. Kravets and P. Krishnan. Application-driven power management for mobile communication, *Wireless Networks,* 6(4):263–277, 2000.

[36] S.-J. Lee and M. Gerla. Dynamic load-aware routing in ad hoc networks. In *IEEE International Conference on Communications (ICC),* 2001.

[37] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *7th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2001.

[38] K. Li, R. Kumpf, P. Horton, and T. Anderson. A quantitative analysis of disk drive power management in portable computers. In *USENIX Symposium,* 1994.

[39] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *Principles of Distributed Computing (PODC),* 2001.

[40] N. Li, J. C. Hou, and L. Sha. Design and analysis of an mst-based topology control algorithm. In *IEEE INFOCOM,* 2003.

[41] Q. Li, J. Aslam, and D. Rus. Online power-aware routing in wireless ad-hoc networks. In *7th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2001.

[42] J. R. Lorch and A. J. Smith. Reducing processor power consumption by improving processor time management in a single-user operating system. In *2nd Annual International Conference on Mobile Computing and Networking (MobiCom),* 1996.

[43] J. P. Monks, V. Bharghavan, and W.-M. Hwu. A power controlled multiple access protocol for wireless packet networks. In *IEEE INFOCOM,* 2001.

[44] A. Muqattash and M. Krunz. Power controlled dual channel (PCDC) medium access protocol for wireless ad hoc networks. In *IEEE INFOCOM,* 2003.

[45] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol. In *European Wireless 2002,* 2002.

[46] M. Pearlman, Z. J. Haas, P. Sholander, and S. S. Tabrizi. On the impact of alternate path routing for load balancing in mobile ad-hoc networks.

In *1st ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc),* 2000.

[47] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications,* 1999.

[48] N. Poojary, S. V. Krishnamurthy, and S. Dao. Medium access control in a network of ad hoc mobile nodes with heterogeneous power capabilities. In *IEEE International Conference on Communications (ICC),* 2001.

[49] J. Rabaey, J. Ammer, J. L. da Silva Jr., and D. Patel. PicoRadio: ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes. In *IEEE Computer Society Annual Workshop on VLSI (WVLSI'OO),* 2000.

[50] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *IEEE INFOCOM,* 2000.

[51] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, 1996.

[52] V. Rodoplu and T. H.-Y. Meng. Minimum energy wireless networks. *IEEE Journal on Selected Areas in Communications,* 17(8): 1333–1344, August 1999.

[53] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava. Topology management for sensor networks: Exploiting latency and density. In *3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc),* 2002.

[54] K. Scott and N. Bambos. Routing and channel assignment for low power transmission in pcs. In *5th IEEE International Conference on Universal Personal Communications,* 1996.

[55] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *IEEE Wireless Communications of Networking Conference (WCNC),* 2002.

[56] E. Shih, P. Bahl, and M. J. Sinclair. Wake on wireless: An event driven energy saving strategy of battery operated devices. In *8th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2002.

[57] S. Singh and C. S. Raghavendra. Pamas: Power aware multi-access protocol with signalling for ad hoc networks. *ACM SIGCOMM Computer Communication Review,* 28:5–26, July 1998.

[58] S. Singh, M. Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *4th Annual International Conference on Mobile Computing and Networking (MobiCom),* 1998.

[59] P. Sinha, R. Sivakumar, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. In *IEEE INFOCOM,* 1999.

[60] I. Stojmenovic and X. Lin. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems,* 12(11):1122–1133, November 2001.

[61] M. W. Subbarao. Dynamic power-conscious routing for MANETs: An initial approach. *Journal of Research of the National Institute of Standards and Technology,* 104:587–593, November-December 1999.

[62] H. Tagaki and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications,* 32(3):246–257, March 1984.

[63] C.-K. Toh. Associativity-based routing for ad hoc mobile networks. *Wireless Personal Communication,* 4(2): 103–139, March 1997.

[64] C. K. Toh. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *IEEE Communications Magazine,* 39(6): 138–147, June 2001.

[65] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh. Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks. In *IEEE INFOCOM,* 2002.

[66] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for wireless multihop ad-hoc networks. In *IEEE INFOCOM,* 2001.

[67] Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin. Topology control protocols to conserve energy in wireless ad hoc networks. Technical Report CENS Technical Report 0006, Center for Embedded Networked Sensing, January 2003. submitted to IEEE Transactions on Mobile Computing.

[68] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *7th Annual International Conference on Mobile Computing and Networking (MobiCom),* 2001.

[69] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *IEEE INFOCOM,* 2002.

[70] W. H. Yoen and C. W. Sung. On energy efficiency and network connectivity of mobile ad hoc networks. In *International Conference on Distributed Computing Systems (ICDCS),* 2003.

[71] R. Zheng, J. C. Hou, and L. Sha. Asynchronous wakeup for ad hoc networks. In *4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc),* 2003.

[72] R. Zheng and R. Kravets. On-demand power management for ad hoc networks. In *IEEE INFOCOM,* 2003.