1 Maximizing Multicast Lifetime in Wireless Ad Hoc Networks

G. DENG and S.K.S. GUPTA

Department of Computer Science and Engineering
Ira A. Fulton School of Engineering at Arizona State University, Tempe, AZ 85287
E-mail: {Guofeng Deng, Sandeep.Gupta}@asu.edu

1.1 INTRODUCTION

Wireless ad hoc networks (WANET), which can consist of any handheld device with a transceiver (e.g., PDAs and laptops), are designed for applications, such as disaster rescue and military applications, where traditional infrastructure-based cell networks do not exist or can not exist. Moreover, recent advances in micro-electro-mechanical systems (MEMS) technology, wireless communications and digital electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate untethered in short distances [16]. However, most of these devices, including both sensors and regular size devices, are driven by batteries with finite capacity. Not only it is inconvenient to recharge batteries while in operation, but also in sensor networks, which are based on collaborative effort of a large number of sensors, it is impractical to collect all sensors to recharge the batteries.

Due to the limitation of available energy in the battery driven devices in WANETs, the longevity of the network is of prime concern. This chapter mainly considers the problem of maximizing lifetime of WANET when multicast traffic dominates. Multicast is an efficient method to disseminate data to a group of destinations. It is the basis of most group communications, which are important applications in WANETs. Most implementations of multicast are tree-based, where data flow along multicast tree links from the root to all destinations. A tree established for each source is a source-based multicast tree, which is rooted at the source and spanning the group of destinations. Problems discussed in this chapter have one source and therefore, multicast information is delivered to the group of destinations through a single or combination of multiple source-based multicast trees. In the situation of multiple sources, one option is that sources share a tree structure, which is referred to as group-shared multicast tree, or core-based tree. To multicast data, each source is first required to transmit data to the core, which is the root of the shared tree. Then, the core forwards them to all destinations through the tree, as if it were the source of the data. The application of group-shared tree inevitably leads to traffic concentration on the shared tree, and in turn fast energy depletion at the nodes in the tree. From this perspective, group-shared tree is not preferred when network lifetime is of concern. Alternatively, multiple source-based multicast trees could be constructed, each of which is rooted at a single source. Thus, it is possible that some nodes are common nodes for several trees. The problem of maximizing lifetime in this scenario is complicated and beyond the discussion in this chapter.

Besides multicast lifetime, some other objectives motivated by limited energy in WANETs have been addressed. The one closely relevant is energy efficient multicast routing, whose target is to maintain a multicast connection from the source to a group of destinations with minimum total energy consumption by all nodes. This problem has been studied intensively in the past few years and many topology control and power-aware techniques have been developed to address the issue of limited energy [5][6][7]. A wireless node is able to choose dynamically the transmit power level at which its transceiver transmits packets. The value of transmit power, in reality, is proportional to the transmission range, a scope within which the signal is legible to the receiver. Further, due to the broadcast nature of the wireless medium, every node located within a node's transmission range can receive packets from that node. This phenomenon is so-called wireless multicast advantage (WMA). Therefore, each node can determine the set of possible one hop away neighbors by adjusting its transmit power. Adjustment of the transmit power in wireless nodes to create a desirable optimized topology is called topology control [1].

The problem of constructing a multicast tree with minimum total energy consumption has been proven to be NP-complete [15]. An energy efficient multicast tree, however, can not guarantee optimum lifetime. Approaches tackling energy constraints in unicast routing in WANETs have one of two different targets. Similar to energy efficient multicast algorithms, the performance objective of energy efficient unicast routing is to minimize the total consumed energy per unit flow, that is to find a path from the source to the destination on which the sum of energy consumption is minimized [2]. The maximum lifetime unicast algorithms consider the problem of maximizing the time to network partition [2][3][4][8]. Results of unicast routing are not directly applicable to multicast problems due to different traffic pattern between unicast and multicast routing, which is one-to-one versus one-to-many. But unicast is a special case of multicast, where only one destination exists.

This chapter is organized as follows. We first introduce energy consumption model in WANETs as background. This is followed by definitions of maximum multicast lifetime routing. In the fourth section, we present solutions to the problem of maximizing multicast lifetime using single tree (MMLS). Then, approach is discussed when multiple multicast trees are allowed, that is, maximizing multicast lifetime using multiple trees (MMLM). We briefly summarize this chapter in the last section and discuss some open research issues.

1.2 ENERGY CONSUMPTION MODEL IN WANETs

As in traditional networks, nodes in WANETs consume energy for processing internal data and simply for being "on" in an idle mode. But the main factor, which constitutes the energy consumption of a wireless node, is the power required to transmit and receive data. Node i can communicate directly to node j if the transmit power from i exceeds some threshold value p_{ij}. In reality, $p_{ij} \propto d_{ij}^{\alpha}$, where d_{ij} is the Euclidean distance between nodes i and j, and exponent α, $2 \leq \alpha \leq 4$, models the decay of the radio signal in the intervening medium [18]. p_{ij} captures cost of link from node i to node j. For simplicity, we assume symmetric links, that is $p_{ij} = p_{ji}$ for any nodes i and j. Therefore, the power required at node i in order to communicate to node j can be expressed as

$$p_i = \begin{cases} p_{ij} & \text{if } i \text{ is the source node,} \\ p_{ij} + p_R & \text{otherwise,} \end{cases} \tag{1.1}$$

where p_R is power used for reception. The source does not receive data from other nodes and therefore no reception cost is involved in a source node. Actually, as can be seen later, the approaches are valid, even if the reception cost is not considered except for some minor differences in results. For simplicity, we choose to only consider power used to transmit data in our energy consumption model.

Moreover, the unit of power mentioned above is Joule per unit time, say second. The energy consumed in each node is directly related to the volume of data transmitted and/or received, especially when power consumed for being simply "on" is negligible compared with that for data transmission and reception. Two units, which are Joule per second and Joule per bit of data (or Joule per packet), can be related to each other by a constant transmission rate C in bit per second (or packet per second). Therefore, they are equivalent. However, to avoid confusion, e_i is used as the transmit power of node i, e_{ij} as the cost of the link from node i to node j, and e_R as the power used for reception, all of which are in Joule per bit. Clearly, $e_i = p_i/C$, $e_{ij} = p_{ij}/C$ and $e_R = p_R/C$. From this point of view, the two units could be used interchangeably. Further, lifetime or duration of multicast tree(s) and wireless node(s) can be described in either time, say seconds, or volume of data, say bits, for the same reason.

Any node, say i, is assumed to be able to control its transmission range by choosing an appropriate transmit power level p_i. Basically, p_i could be any value satisfying $p_{min} \leq p_i \leq p_{max}$, where p_{min} is the minimum transmit power required to send a packet to an arbitrarily near node, and p_{max} is the maximum transmit power in a wireless node. $p_i = 0$ means that node i decides not to transmit packets to any other nodes. If we define node j is a neighbor of node i when node j can successfully receive packets from node i directly, the neighborhood relationship is irreversible. It is possible that node i can hear node j but node j can not receive packets from node i because of different transmit power at nodes i and j. So, the connectivity of the wireless network depends on the transmit power levels at all nodes, which is referred to as a power assignment.

Wireless multicast advantage (WMA) is due to broadcast nature of wireless medium. For example, in Fig.1.1, node s is transmitting data to nodes a and b. The minimum power required at node s is $\max\{p_{sa}, p_{sb}\}$ instead of $p_{sa} + p_{sb}$. From another point of view, node s transmits data with a power level enough for the farthest neighbor to receive it, but to the rest of its neighbors that are closer at free of charge of energy. It is safe to say that WMA has enabled the design toward achieving energy efficiency as well as extension of network lifetime.

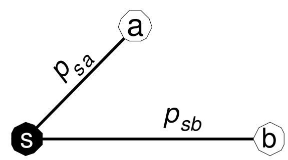

Fig. 1.1 Wireless multicast advantage

For a node in a multicast tree, specifically, the power required at this node depends on the farthest child it has, i.e.,

$$e_i(T_k) = \max_{j \in V_i(T_k)} e_{ij}, \tag{1.2}$$

where $e_i(T_k)$ stands for transmit power required at node i in multicast tree T_k, and $V_i(T_k)$ is the set of children of node i in tree T_k. Unless stated otherwise, algorithms presented in this chapter are off-line algorithms based on global knowledge of the network deployment. We assume once a multicast tree is constructed, it will be used for comparatively long time before any structure modification. Also because nodes in the network are assumed to be reliable and static, overhead for multicast tree maintenance is negligible.

So far, we have introduced power consumption by a node in WANETs. Due to WMA, both energy and bandwidth in wireless communications are conserved. On the other hand, however, it leads to increased level of interference and collision. To focus on the discussion on network lifetime, we assume the work is based on some reliable protocols at the Medium Access Control(MAC) layer and energy consumed for reliable transmission at the MAC layer is not considered in this chapter.

1.3 DEFINITIONS OF MAXIMUM MULTICAST LIFETIME

Lifetime of an individual node can be defined as a continuous amount of time during which the node is operational till the first failure due to battery depletion, or equivalently, the number of bits of data a node can transmit at certain transmit power. Lifetime of node i, denoted by L_i, is determined by both its battery capacity, R_i (Joule or milli-Joule), and the transmit power required at that node, i.e.,

$$L_i = \frac{R_i}{e_i}. \tag{1.3}$$

Further, link longevity is defined as the maximum number of bits that the source of a link can transmit to the node on the other end of the link. If we denote longevity of link i to j by L_{ij},

$$L_{ij} = \frac{R_i}{e_{ij}}. \tag{1.4}$$

Notice that $L_{ij} \neq L_{ji}$ if $R_i \neq R_j$.

Then, what is the relationship between longevity of a node and that of each link starting from it? Since a node is always transmitting data to reach its farthest neighbor, the lifetime of this node is determined by the smallest link longevity originating from it, i.e.,

$$\begin{aligned} L_i &= \tfrac{R_i}{e_i} && (according\ to\ Equation\ 1.3), \\ &= \tfrac{R_i}{\max_j e_{ij}} && (according\ to\ Equation\ 1.2), \\ &= \min_j \tfrac{R_i}{e_{ij}}, \\ &= \min_j L_{ij} && (according\ to\ Equation\ 1.4), \end{aligned}$$

where j is any neighbor of node i. By viewing Equation (1.3), the lifetime of node i in multicast tree T_k can be expressed as

$$L_i(T_k) = \frac{R_i}{e_i(T_k)}. \tag{1.5}$$

It means that in order to maintain a tree connection, the transmit power at each node in the tree is at least the maximum power required to communicate to its farthest child, i.e.,

$$L_i(T_k) = \frac{R_i}{\max_{j \in V_i(T_k)} e_{ij}}. \tag{1.6}$$

A leaf node has an infinite lifetime because of its zero transmit power.

A tree is an acyclic structure. Any node failure will result in at least one tree link failure and hence, disconnect the tree. So, a (multicast) tree gets disconnected even if a single node dies. Consequently, the lifetime of a multicast tree is same as the lifetime of the bottleneck node, the node in the tree with shortest lifetime among all the tree nodes.

Definition 1: Lifetime of a single multicast tree In the scenario where one multicast tree is used throughout the multicast session, lifetime of this tree is the duration till the first node dies due to power depletion. We denote lifetime of multicast tree T_k by $L(T_k)$ such that

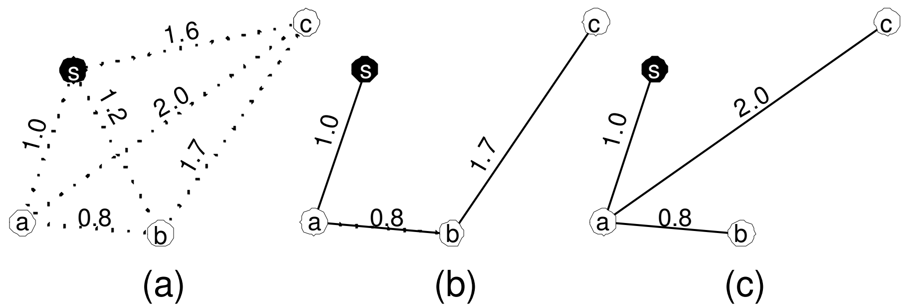

Fig. 1.2 Network deployment and sample multicast trees

$$L(T_k) = \min_{i \in V(T_k)} L_i(T_k), \tag{1.7}$$

where $V(T_k)$ is set of nodes in tree T_k. Then by substituting $L_i(T_k)$ in Equation (1.5), we get

$$L(T_k) = \min_{i \in V(T_k)} \frac{R_i}{e_i(T_k)}. \tag{1.8}$$

For example, Fig.1.2a shows the network deployment with the cost of each link specified. Node s is the source and nodes a, b, and c are destinations, all of which have an identical initial energy of 200 units. In multicast tree in Fig.1.2b, transmit power at node s, a, b, and c is 1.0, 0.8, 1.7 and 0 unit, respectively. In multicast tree in Fig.1.2c, the values are 1.0, 2.0, 0 and 0 unit, respectively. Node a in Fig.1.2c transmits packets at a power of 2.0 unit in order to reach its farthest child c, but can multicast data to nodes b and c at the same time. According to definition 1, multicast tree in Fig.1.2b has a lifetime of 118 bits and multicast tree in Fig.1.2c 100 bits with nodes b and c being bottleneck nodes, respectively.

For a given network deployment, the problem of maximizing multicast lifetime of the network using single tree (MMLS) is to find a multicast tree that has the highest lifetime among all viable multicast trees connecting the given source and destinations, i.e.,

$$MMLS = \max_{k} \min_{i \in V(T_k)} \frac{R_i}{e_i(T_k)}, \tag{1.9}$$

where T_k is any viable multicast tree.

When a tree is used throughout the multicast session, the intermediate nodes in the multicast tree consume their energy fast, especially the bottleneck. The tree dies when the bottleneck exhausts its battery. In many situations, multiple multicast trees can be used alternately to increase the multicast lifetime of the network. For instance, in the earlier example drawn in Fig.1.2, multicast tree in Fig.1.2b has a lifetime of 118 bits and multicast tree in Fig.1.2c 100 bits. If both trees in Fig.1.2b and Fig.1.2c are used one after another, for example, tree in Fig.1.2b is used to transmit 80 bits of data and then multicast session is continued through tree in Fig.1.2c till it dies. After 80 bits transmitted on multicast tree in Fig.1.2b, the residual energy at nodes s, a, b and c is 120, 136, 64, and 200 units, respectively. This means that 68 bits of data can be delivered with the residual energy till the failure of first node, i.e., node a. As a result, a total of 148 bits of data are sent from node s to nodes a, b and c. Instead, if to choose multicast tree in Fig.1.2b for 100 bits and then switch to tree in Fig.1.2c to multicast as many data as possible, a multicast lifetime of 160 bits can be achieved.

Multicast lifetime can be increased by using multiple trees alternately because it is able to balance energy consumption at wireless nodes. The idea is that in each multicast tree, there are one or more nodes whose transmit power is comparatively higher than others. The bottleneck is one of these nodes. If multiple trees exist such that nodes with higher energy cost in each tree are different, or in another word, each node in the network is not costly all the time, it is possible to find a combination of these trees, which is capable of extending the time till the first node dies.

We know lifetime of a single multicast tree is determined by distribution of initial energy and power consumed at each node. For multiple multicast trees, however, the lifetime is affected by the set of multicast trees and duration of each tree used as well, which is shown in the preceding example.

Definition 2: Switching schedule In a multicast session, several multicast trees are used alternately to deliver data from the given source to the group of destinations. A switching schedule describes the duration of each multicast tree that will be used during the multicast session such that energy available at each node in the network can be enough for all transmissions described in the switching schedule and after the execution of the entire switching schedule, the residual energy at each node can afford no more multicast tasks[1].

Formally, a switching schedule SP can be expressed as a set of bi-tuple (t, d), where t is a multicast tree, and $d > 0$, is the duration that multicast tree t will be used. $t_i \neq t_j$ for any elements $(t_i, d_i), (t_j, d_j) \in SP, i \neq j$.

Definition 3: Multicast lifetime of multiple trees Given several multicast trees and a switching schedule, lifetime of these trees is the sum of duration of each multicast tree in the switching schedule. If L stands for the multicast lifetime using switching schedule SP, then

$$L = \sum_{(t_i, d_i) \in SP} d_i. \tag{1.10}$$

For instance, two switching schedule appeared in the previous example are {(tree (b), 80 bits),(tree (c), 68 bits)} and {(tree (b), 100 bits), (tree (c), 60 bits)}, respectively. Therefore, lifetime is 148 bits and 160 bits, respectively.

[1]According to our assumption, only duration of each tree has effects on the multicast lifetime, but the sequence does not.

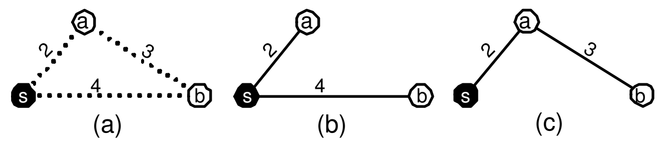

Fig. 1.3 Network deployment and sample multicast trees

The task of maximizing multicast lifetime of the network using multiple trees (MMLM) is actually to find a switching schedule that maximizes the network lifetime. In another word, besides choosing appropriate set of multicast trees, duration of each tree has to be optimized.

Broadcast is a special case of multicast, where all nodes, except the source, are destinations. So, the above definitions are directly applicable to broadcast applications. So far, the definitions of multicast lifetime have been provided. We will look at the differences between two related designs, energy efficient and maximum lifetime multicast routing, based on these definitions. The idea of energy efficient algorithms is to minimize the total energy consumption by all nodes in the multicast tree. Due to wireless multicast advantage, a wireless node can communicate to all nodes within its transmission range with transmit power required to reach its farthest one hop away neighbor. To reach an even farther node, the sender can transmit packets at higher power or one of its neighbors can forward packets for the sender. Among two options, the first is preferred if the increment of the transmit power at that node is smaller than the transmit power required at the forwarder. Generally, objective of energy efficiency leads to higher energy consumption at a subset of nodes and lower consumption at others. We know that for a node with fixed battery capacity, however, the higher the transmit power, the shorter the lifetime. Therefore, minimum total energy consumption can not guarantee maximum multicast lifetime. Fig.1.3a illustrates an example deployment of wireless ad hoc networks consisting of 3 nodes and with link cost specified. Among these nodes, node s is the source, and both nodes a and b are receivers. We assume that all three nodes have same initial energy, say 200 units. Fig.1.3b and Fig.1.3c are two multicast trees connecting the source and destinations. Actually, tree in Fig.1.3b is a multicast tree with the minimum total energy consumption by all nodes and tree in Fig.1.3c is the one with maximum lifetime. Specifically, transmit power at nodes s, a and b is 4, 0, and 0 unit respectively in Fig.1.3b and 2, 3, and 0 unit in Fig.1.3c. Total power consumption is 4 units $(4 + 0 + 0)$ in Fig.1.3b and 5 units $(2 + 3 + 0)$ in Fig.1.3c, while lifetime is 40 bits with node s being the bottleneck and 200/3, a little more than 66 bits, with bottleneck node a, respectively. The result in this example is that tree in Fig.1.3c has better lifetime than tree Fig.1.3b although it consumes more energy as a whole.

1.4 MAXIMUM MULTICAST LIFETIME OF THE NETWORK USING SINGLE TREE (MMLS)

MMLS problem naturally leads to a max-min optimization problem because the minimum time at which the first node failure occurs need to be maximized [10]. In this section, we first prove that MMLS is solvable within polynomial time under both identical and non-identical battery capacity situations. Then, solution in a special case, which is maximum broadcast lifetime of the network using single tree (MBLS), is presented. Approaches discussed in first two subsections are centralized solutions. A distributed algorithm, L-REMiT [9], will be introduced briefly at the end of this section.

1.4.1 MMLS

We know battery capacity together with transmit power assignment determines node lifetime, and further affects lifetime of the multicast tree, as shown in Equation (1.5) and (1.8). In the case of same type of battery at each node, the problem of MMLS is reduced to find a multicast tree, whose highest transmit power is minimized. That is, if $R_1 = R_2 = \cdots = R^*$,

$$\begin{aligned} MMLS &= \max_k \min_{i \in V(T_k)} \frac{R^*}{e_i(T_k)}, \\ &= \frac{R^*}{\min_k \max_{i \in V(T_k)} e_i(T_k)}. \end{aligned} \quad (1.11)$$

A straightforward method is to enumerate all possible multicast trees connecting the given source and destinations and then choose the tree whose highest transmit power is smallest among all trees. The lifetime of this tree is an answer to MMLS. However, this simple method is not desirable because of poor efficiency. According to Cayley Formula [14], the number of spanning trees of a complete graph K_n is n^{n-2} for any $n \geq 2$. So, the above method has an exponential time complexity.

Is there a solution with better complexity? The answer is yes. Referring to Equation (1.8), if $R_1 = R_2 = \cdots = R^*$, we have:

$$\begin{aligned} L(T_k) &= \min_{i \in V(T_k)} \frac{R^*}{e_i(T_k)}, \\ &= \frac{R^*}{\max_{i \in V(T_k)} e_i(T_k)}. \end{aligned}$$

This means that under the situation of identical battery capacity, the lifetime of multicast tree is determined by the largest transmit power in the tree. Suppose that in multicast tree T_k, the highest transmit power is $e^*(T_k)$. For any node i in tree T_k, its transmit power satisfies $e_i(T_k) \leq e^*(T_k)$. By increasing $e_i(T_k)$ to $e^*(T_k)$, lifetime of the multicast tree is not changed although the lifetime of node i is decreased. So, we can increase power consumed at each node to $e^*(T_k)$ if possible without affecting lifetime of the tree. The idea is to assign same transmit power to each node in the network and search for the smallest power such that a multicast tree can be constructed spanning the given source and destinations [15].

The network is modeled as a unidirectional graph $G = (V, E)$, where V is the set of nodes and E is the set of available links, that is $E = \{(i, j)|e_{ij} \leq e_i\}$. When each node in the network is assigned the same transmit power p, E is denoted by E_p and $E = E_p = \{(i, j)|e_{ij} \leq p\}$. It is straightforward that $E_p \subseteq E_q$ if $p \leq q$. To construct corresponding E_p for given network deployment, the only thing needs to be done is to compare link cost between each node-pair to the assigned power. Further, we know in a connected graph, basic tree-growing scheme [14] is able to generate a multicast tree spanning all the nodes. So, there exists such a tree as long as the source and all the receivers are in a single connected graph. The following algorithm is used to incrementally construct a connected subgraph starting from the given source, say node s, when E is available.

Connected-Subgraph-Construction*(E, s)*:

```
C = {s};
N = V\C;
for each node i in C
   for each node j in N
   {
      if (i, j) ∈ E
      {
         C = C + {j};
         N = N − {j};
      }
   }
return C;
```

Return value C from the algorithm, Connected-Subgraph-Construction, is a connected subgraph including the source s. All destinations are connected to node s directly or indirectly as long as they are included in C. The procedure for solving MMLS in the scenario of identical battery is summarized in the following algorithm, which is based on the ideas by Lloyd et al. [15]:

Minimum-Transmit-Power-Search:

```
Sort all possible transmit power levels;
While(true)
{
   p = smallest possible transmit power that has not been tried so far;
   Calculate E_p;
   C = Connected-Subgraph-Construction(E_p , s);
   If (the set of destinations ⊆ C) break;
}
return p;
```

After minimum transmit power p is returned successfully from the preceding algorithm, MMLS can be computed easily by dividing R over p. If C is a connected subgraph corresponding to the minimum transmit power p, any multicast tree built through tree-grow scheme based on C has a lifetime of MMLS. The multicast tree may include some unrelated nodes, which are not multicast nodes (either the source or one of the destinations) and none of their descendants is multicast node. These nodes, will be removed by pruning. The structure of the multicast tree with maximum lifetime may not be unique.

Now, we consider the complexity of this searching algorithm [2]. Since e_{ij} is known for any nodes i and j, to calculate E_p, n^2 comparisons are needed where n is the total number of nodes in the network. The number of steps to construct a connected subgraph will be less than $O(n^2)$ as per Connected-Subgraph-Construction algorithm. So, for each possible transmit power value, $O(n^2)$ steps are required to check the feasibility. For each node, there are not more than $n-1$ possible transmit power levels because there is at most one new power value for each neighbor. This does not include the possibility that a node could choose not to forward any data, that is zero transmit power. These add up to at most n possibilities. Then, for all n nodes in the network, there are n^2 different transmit power levels at most. As a result, the total running time to find the optimized maximum lifetime is $O(n^4)$, i.e., polynomial. The above minimum power search algorithm can be optimized by sorting the $O(n^2)$ candidate solution values and using binary search to determine the smallest value such that the given source and destinations are connected [15]. In this situation, the running time is reduced to $O(n^2 \log n)$.

The above approach is able to solve MMLS in the scenario of identical battery capacity, where the lifetime of a multicast tree is determined by the highest transmit power in the tree. In the more general situation, where batteries are not identical, the node with highest transmit power may not be the bottleneck. To search for the maximum lifetime, longevity of each node is used instead of transmit power. Each node is assigned the same lifetime and the transmit power is adjusted according to its initial energy. For example, if l^* is a candidate lifetime, which is intuitively the lifetime of the possible multicast tree, for node i with initial energy R_i, it must choose its transmit power e_i to be $\min\{\frac{R_i}{l^*}, p_{max}\}$. Based on the transmit power at each node, link set E is calculated. Similar to the situation of identical battery, the number of possible node lifetime is at most n^2. The following Maximum-Lifetime-Search algorithm is a variation of the algorithm of Minimum-Transmit-Power-Search and therefore has same running time. As opposed to Minimum-Transmit-Power-Search algorithm, this algorithm returns MMLS directly.

Maximum-Lifetime-Search:

Sort all possible node lifetime:
While(true)
{
  $l =$ the highest lifetime value that has not been tried so far;
  Calculate $e_i = \frac{R_i}{l}$ for each node i;
  Calculate $E = \{(i,j) | e_{ij} \leq e_i\}$
  $C =$ Connected-Subgraph-Construction(E, s);
  If (the set of destinations $\subseteq C$) break;
}
return l;

[2] Similar complexity analysis appeared in [15]

1.4.2 MBLS

So far, we show that MMLS is solvable by using enumeration within polynomial time. Now we are going to introduce approaches to a special case problem, which is maximum broadcast lifetime of the network using single tree (MBLS). Similarly, the solutions are divided into two cases: identical and non-identical battery capacity.

Minimum spanning tree (MST) is a tree containing each vertex in the graph such that the sum of the edges' weights is minimum. It has been proven to be the tree minimizing the maximum power consumption as well [10]. In the case of identical battery capacity, according to Equation (1.11), MST is a globally optimal solution to MBLS. Further, a hybrid algorithm, which applies both Borvka algorithm and Prim algorithm, is developed to solve the minimum spanning tree problem in $O(m \log \log n)$ time, where m is the number of edges and n is the number of nodes [19]. This is faster than $O(n^2 \log n)$, the time complexity of the enumerating method, since $m \leq n(n-1)$. One example is depicted in Fig.1.4a and Fig.1.4d. Fig.1.4a is network deployment with link cost specified and the tree in Fig.1.4d is a MST generated through Prim algorithm [14]. If an identical battery capacity of 200 units is assumed, MBLS of network in Fig.1.4a will be 125 bits with node s being the bottleneck.

MST could also be applied to solve MBLS problem when batteries are non-identical, where the network is modeled as a directed graph instead and the weight of each link is no longer transmit power, but the inverse of link longevity [12]. Then, MST is built based on the directed graph and the result MST is referred to as DMST (directed MST). DMST minimizes the maximum inverse of link longevity, and in turn, maximizes the minimum link longevity. Therefore, DMST is a globally optimal broadcast tree with maximum broadcast lifetime.

Here is an example. The deployment and link costs are shown in Fig.1.4a. Fig.1.4b shows initial energy available at each nodes, specifically, 100, 200, 300 and 400 units at nodes s, a, b and c, respectively. Inverse of each link longevity is also specified in Fig.1.4b. The tree in Fig.1.4e is a DMST generated from network in Fig.1.4b through Prim algorithm. Then, MBLS in this scenario is 100 bits with node s dying first. Another example with the same deployment but different initial energy at each node is provided in Fig.1.4c to illustrate the effects of different distribution of battery capacity. In this network, node s, a, b and c has 200, 100, 300, and 400 units energy, respectively. The tree in Fig.1.4f is the DMST constructed from Fig.1.4c and it is different from the broadcast tree in Fig.1.4e. MBLS of the network in Fig.1.4c is 176 bits till node b's depletion of energy.

1.4.3 A distributed MMLS algorithm: L-REMiT

Above discussion presents centralized approaches to MMLS problem in WANETs. L-REMiT [9], a refinement-based distributed algorithm is introduced as follows.

Basically, L-REMiT formulates the task of extending the lifetime of a multicast tree as extending the lifetime of bottlenecks in the tree. By reassigning the farthest children to other nodes, the bottleneck is able to reduce its transmit power and in

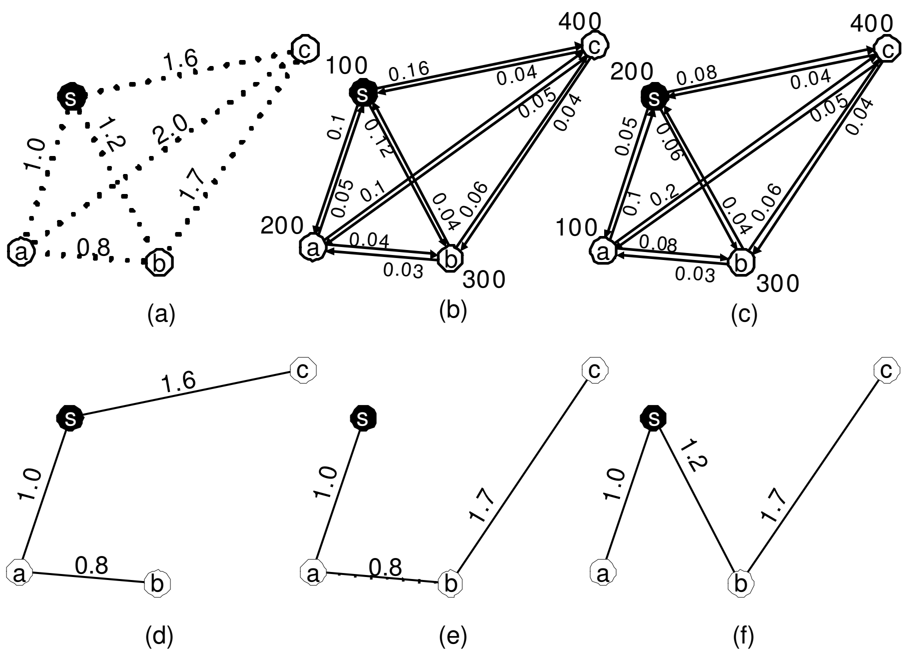

Fig. 1.4 Example of maximum multicast lifetime using one tree

turn increase its lifetime. Rational behind L-REMiT is that a multicast tree remains connected after a node switches its parent to a non-descendant node if the node itself, its original parent and possible new parent are all in the multicast tree. Further, the two nodes, the parent before and after the switch, are the only nodes whose lifetimes are affected. Therefore, the lifetime of the multicast tree will be increased as long as the lifetime of the former parent, which is the bottleneck, is increased, and at the same time the updated lifetime of the new parent is greater than the lifetime of the original multicast tree.

L-REMiT works as follows. All nodes run a distributed algorithm, say algorithm proposed by Gallager et al. [20], to build a MST at the beginning. Then, refinement steps are conducted on the initial spanning tree in rounds coordinated by the source. In each round, a bottleneck is chosen in a bottom up manner from the leaf nodes to the source. After the bottleneck gets L-REMiT token from the source, it switches its farthest child to another node in the tree if lifetime of the tree will be increased. Otherwise or after the refinement, the bottleneck passes the token back to the source using the reverse tree path from itself to the source. The procedure stops till no further improvement can be found. As the last step, pruning is conducted to remove all the nodes which are not needed to cover all the multicast group nodes.

1.5 MAXIMUM MULTICAST LIFETIME OF THE NETWORK USING MULTIPLE TREES (MMLM)

In the previous section, the approaches to optimal maximum lifetime are studied when one multicast tree is used throughout multicast session. This is also referred to as static power assignment [11]. In a static power assignment, one result is inevitable, where only a subset of nodes fail due to battery exhaustion while there are still some amount of energy remaining at other nodes. Especially, residual energy at the leaf nodes is barely touched. This part of energy is actually wasted. In this section, approaches to extend multicast lifetime by making use of all available energy are discussed.

1.5.1 MMLM

The problem of maximum multicast lifetime of the network using multiple trees (MMLM) is also called dynamic assignment problem [11] because the transmit power at each node is allowed to change dynamically in order to switch from one multicast tree to another. We have seen an example of multiple multicast trees being used alternately earlier in this chapter. Recall the conclusion we reached earlier: to solve MMLM problem is equivalent to finding a collection of multicast trees connecting the source and destinations, along with the corresponding duration of each tree, which we refer to as an appropriate switching schedule.

We first introduce a solution based on linear programming (LP), which is an efficient mathematical tool to solve maximum or minimum linear functions subject to linear constraints [13]. Authors in [11] have come with a similar approach. If

MMLM problem can be transformed to a linear function and corresponding linear constraints, LP is capable of solving the problem. The following expression shows one possible transform.

$$\begin{array}{l} \max CX \\ s.t.\, AX \leq R\, and\, X \geq 0 \\ where\, C = [1, 1, ..., 1], \\ X = [x_1, x_2, ..., x_m]^T, \\ A = [a_{i,j}]_{n \times m}, \\ and\, R = [R_1, R_2, ..., R_n]. \end{array} \tag{1.12}$$

In this linear program, the objective is to maximize total execution time, i.e. the lifetime. Suppose there are n nodes in the network and m viable multicast trees, x_i stands for execution time of tree T_i, A is power assignment matrix, and element $a_{i,j}$ is transmit power of node i in tree T_j. Since network deployment is given, all viable multicast trees could be enumerated. Actually, matrix A is a description of these multicast trees. R_i is battery capacity in node i. The constraints $AX \leq R$ mean that the summary of energy consumed in each tree will not exceed its initial energy. Apparently, execution time in each tree should not be less than 0. The objective function and constraints are then fed to any mathematical tool, which is capable of solving linear programs, for example, Matlab [21]. Finally, MMLM is computed by summing up x_i.

A numerical example is provided to illustrate how LP finds optimal solutions. Network deployment is given in Fig.1.5a. Node s is the source and nodes a, b and c are receivers. Each node has an identical battery capacity of 200 units. To make the example more realistic, we assume reception cost to be 0.1 units. Matlab [21] is used as linear program optimization tool.

As a result, MMLM of the network in Fig.1.5a is 188 packets and detailed switching schedule can be seen in Table1.1. Notice that in Fig.1.5, only a subset of all possible multicast trees are shown, among which multicast trees in Fig.1.5b, Fig.1.5c, and Fig.1.5d appear in the LP result. Multicast trees in Fig.1.5e, Fig.1.5f, and Fig.1.5g are drawn here for comparison. The tree in Fig.1.5g is actually a MST generated through Prim algorithm. From the results reached in the previous section, we know in single tree situation, MST is a tree that has the maximum lifetime, which is 125 bits. Compared with MMLS, MMLM is about 50% better in this scenario.

Table 1.1 Optimal switching schedule in the example

Multicast tree	Duration
Fig.1.5 (b)	55 bits
Fig.1.5 (c)	75 bits
Fig.1.5 (d)	58 bits

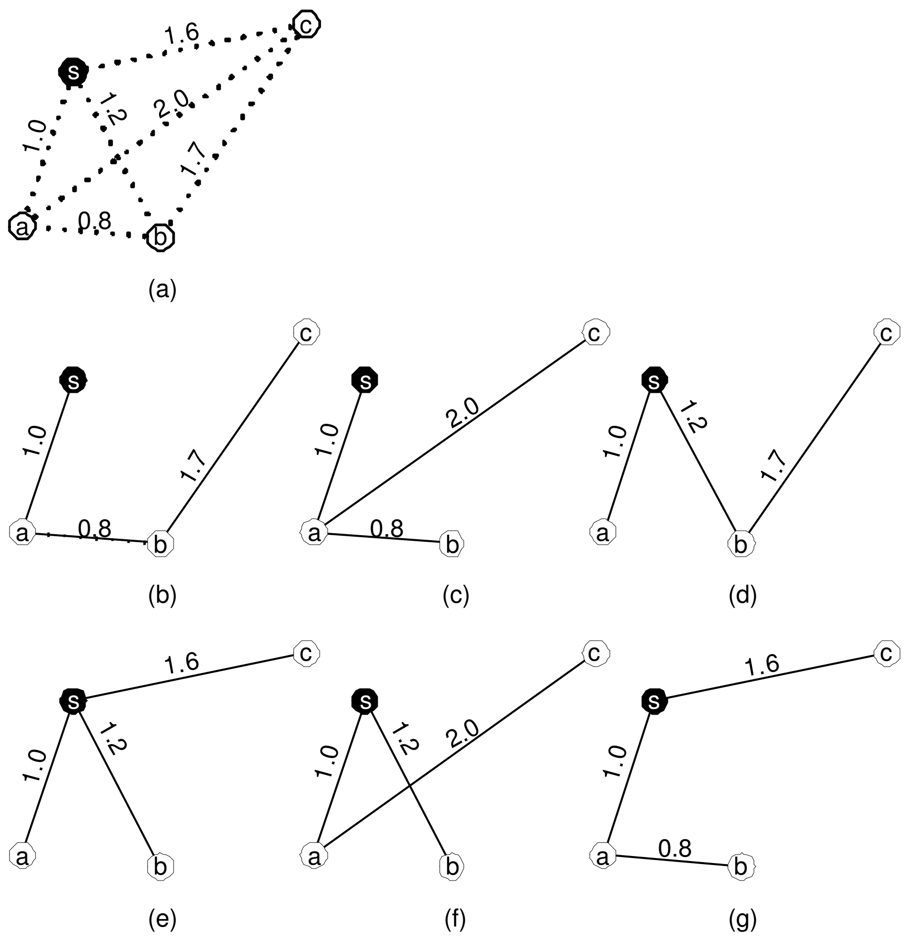

Fig. 1.5 Example of maximum multicast lifetime using multiple trees

We have described that LP is capable of dealing with MMLM problem when all the multicast trees are given. Although LP can solve max/min problem efficiently even for some comparatively large number of constraints, it is not desirable to apply LP in the above manner. This is because the time complexity to enumerate all possible multicast trees is exponential. It has been proven that quantized version of MMLM problem is NP-hard [11].

1.5.2 Some approximations

The problem of MMLM is hard because of NP-hardness. Up to now, only very few results have appeared in the literature. Now, we are going to introduce two algorithms aiming to extend lifetime by applying the idea of multiple multicast trees.

One algorithm developed by Floréen et al. [11] utilizes LP in a way similar to the method presented earlier in this section. To avoid exponential complexity, the algorithm generates fixed number of multicast trees. All these trees are built randomly. To conserve energy, these trees are optimal in the sense that transmit power at each node is minimized such that any reduction will lead to tree partition. Based on these trees, LP is used to achieve optimal solutions. The problem with this algorithm is that it is hard to choose appropriate number of multicast trees. Besides, randomly generated multicast trees can not guarantee optimal solution.

Another algorithm by Sheu et al. [17] is a probabilistic solution. It is different from the idea of dynamic assignment presented earlier in this section and the above algorithm because this algorithm does not work toward a specific switching schedule. It is actually a probabilistic flooding algorithm. A node determines its transmit power or whether or not to forward the packet randomly only after a packet is received. The decision is made based on local information, such as its residual energy level, number of neighbors, and average residual energy of neighbors. The nodes with less energy will have lower probability to broadcast than those with more energy. The low energy nodes are inhibited from forwarding in order to balance the residual energy at each node and in turn extend the lifetime of the networks. The problems with this algorithm include data reception by destinations is not guaranteed, and the result is not optimal.

1.6 SUMMARY

In this chapter, we have discussed approaches to both MMLS and MMLM problems. Solutions to MMLS and its special case problem, which is MBLS, have been explained and proven to have polynomial time complexity. Although multicast lifetime could be extended by using a switching schedule with multiple trees, the problem of MMLM is hard due to its NP-hardness. We will end our discussion by introducing some observed impacts of various parameters on network lifetime performance with multiple multicast trees through simulation [12]: the achievable gain in network lifetime is about twice the optimal network lifetime using single tree; when the switching interval is below a certain threshold, no further gain in lifetime is observed; the network lifetime using multiple trees increases linearly as a function of network density, which is mainly due to increase in available energy pool of the network, i.e., total energy at all nodes.

Acknowledgments

Special thanks to Bin Wang and Tridib Mukherjee of Arizona State University for valuable suggestions and discussions. This work is supported in part by NSF grants ANI-0123980 and ANI-0196156.

REFERENCES

1. R. Ramanathan and R. Rosales-hain, "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment," *IEEE Proceedings of INFOCOM,* Tel-Aviv, Israel (2000)

2. S. Singh, M. Woo, and C.S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proceedings of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking,* Dallas, TX , Oct.1998, pp.181-190

3. J.-H. Chang and L. Tassiulas, "Routing for Maximum system Lifetime in Wireless Ad-hoc Networks," *Proceedings of 37th Annual Allerton Conference on Communication, Control, and Computing,* September 1999

4. J.-H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," *IEEE Proceedings of INFOCOM,* 2000

5. J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Tree in Wireless Networks," *IEEE Proceedings of INFOCOM,* 2000

6. M. Cagalj, J.P. Hubaux, and C. Enz, "Minimum-Energy Broadcast in All Wireless Networks: NP-Completeness and Distribution Issues," *ACM Proceedings of MOBICOM,* 2002

7. B. Wang and S.K.S. Gupta, "S-REMiT: A Distributed Algorithm for Source-Based Energy Efficient Multicasting in Wireless Ad Hoc Networks", *Proceedings of IEEE 2003 Global Communications Conference (GLOBECOM),* San Francisco, CA, December 2003

8. A. Sankar and Z. Liu, "Maximum Lifetime Routing in Wireless Ad-hoc Networks," *IEEE Proceedings of INFOCOM,* 2004

9. B.Wang, S.K.S. Gupta, "On Maximizing Lifetime of Multicast Trees in Wireless Ad Hoc Networks," *Proceedings of the 2003 International Conference on Parallel Processing (ICPP'03)*

10. I.Kang, R.Poovendran, "Maximizing Static Network Lifetime of Wireless Broadcast Adhoc Networks," *Communications, 2003. ICC '03. IEEE International Conference on,* Volume: 3 , 11-15 May 2003, Pages:2256 - 2261

11. P.Floréen, P.Kaski, J.Kohonen, P.Orponen, "Multicast Time Maximization in Energy Constrained Wireless Networks," *DIALM-POMC'03,* September 19, 2003, San Diego, California, USA

12. I.Kang, R.Poovendran, "Maximizing Network Lifetime of Broadcasting Over Wireless Stationary Adhoc Networks," *UWEETR-2003-0002, UWEE Technical Report Series*

13. R.B. Darst, *Introduction to Linear Programming - Applications and Extensions*, Marcel Dekker, Inc. 1990

14. J. Gross, J. Yellen, *Graph Theory and its Applications,* CRC Press 1998

15. E.L. Lloyd, R. Liu, M.V. Marathe, "Algorithm Aspects of Topology Control Problems for Ad Hoc Networks," *MOBIHOC'02,*June 9-11, 2002, EPFL lausanne, Switzerland

16. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless Sensor Networks: a Survey," *Computer Networks* 38 (2002) 393-422

17. J.-P. Sheu, Y.-C. Chang, H.-P. Tsai, "Power-Balance Broadcast in Wireless Mobile Ad Hoc Networks," *The Fifth European Wireless Conference,* February 24-27, 2004, Barcelona, Spain

18. T.S. Rappaport, *Wireless Communications: Principles & Practice,* Prentice Hall, Upper Saddle River NJ, 1996

19. B.Y. Wu and K.-M. Chao, *Spanning Trees and Optimization Problems,* Chapman & Hall/CRC, 2004

20. R. Gallager, P.A. Humblet, and P.M. Spira, "A Distributed Algorithm for Minimum Weight Spanning Trees," *ACM Trans. Programming Lang. & Systems,* 5(1):66-77, Jan. 1983

21. www.mathworks.com