The 10th IEEE International Conference on High Performance Computing and Communications

Lifetime Approximation Schemes Allow Multicast Algorithm with Linear Message Complexity in Wireless Sensor Networks

Mianxiong Dong, Song Guo and Minyi Guo
School of Electrical and Computer Engineering, The University of Aizu, Japan
{d8101104, sguo, minyi}@u-aizu.ac.jp

Abstract

We consider an optimization problem in wireless sensor networks (WSNs) that is to find a multicast tree with maximum lifetime. While a recently proposed distributed algorithm for this problem guarantees to obtain optimal solutions, its high message complexity may prevent such contribution from being practically used in resource-constrained WSNs. In this paper, we proposed a new distributed algorithm that achieves a good balance on the algorithm optimality and message complexity. We use a graph theoretical approach, by the first time, to derive the bounds of both approximation ratio and message complexity in an analytical expression for this algorithm. The theoretical analysis on the tradeoff between these two performance metrics is also validated by our simulation studies.

1. Introduction

Wireless sensor networks (WSNs) have received significant attention in recent years due to their wide range of potential civil and military applications, *e.g.* in health monitoring and coordination among doctors and nurses, aircraft flight control, weather forecasting, home appliance control, and protection against bioterrorism. Since each node in such networks is usually powered by a battery with limited amount of energy, the WSNs will become unusable after the batteries are drained. Consequently, energy efficiency is an important design consideration for WSNs. The multicast communication is also an important issue as many routing protocols need this mechanism to maintain the routes between nodes. Therefore, one would be interested in finding an algorithm that would provide the maximum lifetime to the multicast session. The optimization metric is typically defined as the duration of the network operation time until the battery depletion of the first node in the network.

Some work has considered maximizing the network lifetime in a wireless ad hoc network for a broadcast session, *e.g.* [1, 2, 3, 4], or a multicast session, *e.g.* [4, 5, 6, 7, 8]. In [9, 10], the authors extend the minimum-energy metric by incorporating residual battery energy based on the observation that long-lived multicast / broadcast trees should consume less energy and should avoid nodes with small residual energy as well. Recently, some work has proved this optimization problem belonging to P [6, 7, 8, 11, 12] by proposing various optimal solutions.

We note that most of the existing solutions are centralized, *e.g.* [4 - 7], meaning that at least one node needs global network information in order to construct an energy efficient multicast tree. Sometimes, this centralized approach is impractical for resource-constrained WSNs in which each node has limited energy, bandwidth, memory, and computation capabilities. The most desirable work has been presented in [12], in which two distributed maximum-lifetime algorithms DMMT-OA (Distributed Min-Max Tree algorithm for Omnidirectional Antennas) and DMMT-DA (Distributed Min-Max Tree algorithm for Directional Antennas) have been proposed for directional communications. In particular, it has been proved that the degenerate version of both distributed algorithms, the DMMT algorithm, for omni-directional antennas is globally optimal. However, its theoretical performance in terms of message complexity is still unknown so far.

We consider this optimization problem in a general network that is modeled as a weighted graph with n nodes and m arcs. In this paper, we first reinvestigate the message complexity of the DMMT algorithm and have found that it is at the order of $\Omega(n^2)$, which motivates us to design new distributed algorithm with low message complexity. We then propose a heuristic algorithm that runs in a distributed fashion with guaranteed performance. From a graph theoretical approach, we prove that the proposed algorithm is a constant-factor approximation algorithm, *i.e.* its

978-0-7695-3352-0/08 $25.00 © 2008 IEEE
DOI 10.1109/HPCC.2008.121

approximation ratio[1] is bounded by a finite number. In order to study the tradeoff between the algorithm optimality and message complexity, we derive their upper bounds in a closed form. By simulation studies, the performance of the proposed algorithm is evaluated and our theoretical analysis on the tradeoff between these two performance metrics is also validated.

The rest of this paper is organized as follows. Section 2 develops the system model. Section 3 analyzes the message complexity of the existing DMMT algorithm and proposes a new distributed algorithm DMMT-EQ. The theoretical analysis on algorithm optimality and message complexity of the DMMT-EQ algorithm is given in Sections 4 and 5, respectively. Section 6 evaluates the performance of the distributed algorithms and studies the tradeoff of these two performance metrics using simulations. Section 7 gives the conclusion on the results.

2. Network model

In wireless sensor networks, the sensor nodes are a set of homogeneous low-cost electrical devices. We assume that the omni-directional antenna equipped for each node has a fixed coverage range R. A static wireless sensor network can thus be modeled as a simple graph G with a finite node set N ($n = |N|$) and an arc set A ($m = |A|$). A wireless link (v, u) corresponding to the unidirectional wireless communication channel between nodes v and u, separated by a distance r_{vu}, belongs to the arc set A only if $r_{vu} \le R$. A constant RF power p_R is used for each transmitting node to cover all the neighboring nodes within the range R.

We consider a source-initiated multicast with multicast members $M = \{s\} \cup D$, where s is the source node and D is the set of destination nodes. The topology constructed by a tree-based multicast routing protocol for multicast traffic delivery is a multicast tree, which is rooted at the source node and includes all the destination nodes. A multicast tree T_s, with a tree node set $N(T_s)$ and a tree arc set $A(T_s)$, is formally defined as a rooted Steiner tree, which is a directed acyclic graph with a source node s with no incoming arcs, and each other node has exactly one incoming arc. A node with no out-going arcs is called a leaf node, and all other nodes are internal nodes (also called relay nodes). An important property of a rooted multicast tree is that there exists a unique directed path from the source node to any destination node in the tree.

Let the battery supply ε_v be the energy level associated with each node v. At the beginning of the sensor node deployment, the battery supply is at the full energy level $\varepsilon_{\max}$. With the evolution of the network, the energy level at each node will gradually decrease until depletion. The lifetime of a multicast session is typically defined as the duration of the multicast tree operation time until the battery depletion of the first node in the network [1 - 8]. In order to avoid the premature failure of new-constructed multicast tree, the sensor nodes with energy level lower than $\varepsilon_{\min}$ should not be in the multicast tree as a relay node.

Base on the above model, the lifetime t_{vu} of arc (v, u) in a multicast tree T_s is ε_v/p_R. Let Ω_M be the family of all multicast trees rooted at s and including all the nodes in D. The maximum-lifetime multicast problem can thus be expressed as

$$\max_{T_s \in \Omega_M} \min_{(v,u) \in A(T_s)} (t_{vu}) = \frac{1}{\min\limits_{T_s \in \Omega_M} \max\limits_{(v,u) \in A(T_s)} (\frac{1}{t_{vu}})} . \tag{1}$$

Note that if we assign the arc weight function $w(v, u)$, or denoted as w_{vu} equivalently, as the reciprocal of arc (v, u)'s lifetime, *i.e.*

$$w_{vu} \equiv \frac{1}{t_{vu}} = \frac{p_R}{\varepsilon_v}, \quad \varepsilon_{\min} \le \varepsilon_v < \varepsilon_{\max} , \tag{2}$$

the maximum-lifetime multicast problem is equivalent to the *directed min-max Steiner tree problem*. Given a multicast tree T_s in a network instance $G(N, A)$ with an arc weight function w, we use $\delta_w(T_s)$ to denote the maximum arc weight, *i.e.*

$$\delta_w(T_s) = \max_{(v,u) \in A(T_s)} w_{vu} . \tag{3}$$

The tree arcs with weight $\delta_w(T_s)$ are called the bottleneck arcs. The directed min-max Steiner tree problem is to determine a rooted Steiner tree T_s spanning all the multicast members (*i.e.* $M \in N(T_s)$) such that the bottleneck arc weight is minimized. The corresponding optimal solution δ_w^* satisfies

$$\delta_w^* = \min_{T_s \in \Omega_M} \delta_w(T_s) , \tag{4}$$

which is just the reciprocal of the lifetime of the maximum-lifetime multicast tree.

In the following, we briefly describe some notations used in the rest of the paper. Let C_X denote the cut straddling a node partition X and $N–X$, in which the first node set X must include the source node s and the second node set $N - X$ must include at least one destination node, *i.e.*

[1] An algorithm for a problem has an approximation ratio of $\rho(n)$ if, for any input of size n, the expected cost c of the solution produced by the algorithm is within a factor of $\rho(n)$ of the cost c^* of an optimal solution: $\max\{c/c^*, c^*/c\} \le \rho(n)$.

$$C_X \equiv \{(v, u) \mid v \in X \wedge u \in N - X \wedge s \in X \wedge D \not\subset X\}. \tag{5}$$

We use $\psi_w(C_X)$ to denote the minimum weight of the cut links under the arc weigh function w, *i.e.*

$$\psi_w(C_X) = \min_{(v,u)\in C_X} w_{vu}\,. \tag{6}$$

3. Distributed min-max tree algorithms

As mentioned earlier, the most desirable result for the directed min-max Steiner tree problem is the DMMT (Distributed Min-Max Tree) algorithm presented in [12]. It runs in a distributed fashion and provides globally optimal solutions for wireless sensor networks with omni-directional antennas. While it requires low requirements on memory and computational capacities, its message complexity may be still high. In this section, we first give a brief description of the algorithm DMMT [12], which is non-strict but sufficient for us to study its worst-case message complexity. This reinvestigation would motivate us later to design a new distributed and scalable algorithm.

3.1. The DMMT Algorithm

The DMMT algorithm has an initial neighbor discovery process which allows each node v to aware the existence of all its neighbors N_v. Whenever there is a multicast session request (s, D) in the network but no route information is known, the source will iteratively perform the Search-and-Grow procedure, as described in pseudo code in Fig. 1, by propagating the Search-Report and Grow-Request messages to construct an optimal min-max tree until the tree contains all the destination nodes.

Search-and-Grow procedure	
Search:	find the minimum weight $\psi_w(C_{N(T_s)})$.
Grow:	the tree T_s then grows by absorbing as many links as possible such that any included link (v, u) must satisfy $w_{vu} \le \psi_w(C_{N(T_s)})$ and the resulting sub-graph still keeps a tree structure until no more such links can be found.

Figure 1. **The *Search-and-Grow* procedure**

Initially, T_s^0 contains the source node s only. We now consider the i-th ($i \ge 1$) round of Search-and-Grow procedure and use T_s^i to denote the tree partially constructed after such procedure. In the Search phase, each tree node $v \in N(T_s^{i-1})$ first calculates an upper bound, denoted as $\breve{\psi}_w(v)$, of $\psi_w(C_{N(T_s^{i-1})})$ locally as

$$\breve{\psi}_w(v) \equiv \begin{cases} \dfrac{p_R}{\varepsilon_v} & \exists u \notin N(T_s^{i-1}), r_{vu} < R \\ +\infty & \text{otherwise} \end{cases}. \tag{7}$$

It then sends back a Search-Report message to its parent node (if $v \ne s$) with the parameter $\breve{\psi}_w(v)$ if v is a leaf node or, otherwise, the parameter $\min\{\breve{\psi}_w(v), \breve{\psi}_w(u) \mid u\text{: } (v, u) \in A(T_s^{i-1})\}$ after collecting all the Search-Report messages from its child nodes. These messages propagating back to the source, shown as the dotted arrowed lines in Fig. 2, shall eventually allow the source to obtain $\psi_w(C_{N(T_s^{i-1})})$. Consequently, the source initiates a Grow phase by flooding the Grow-Request messages with the parameter $\psi_w(C_{N(T_s^{i-1})})$ over the tree T_s^{i-1}.

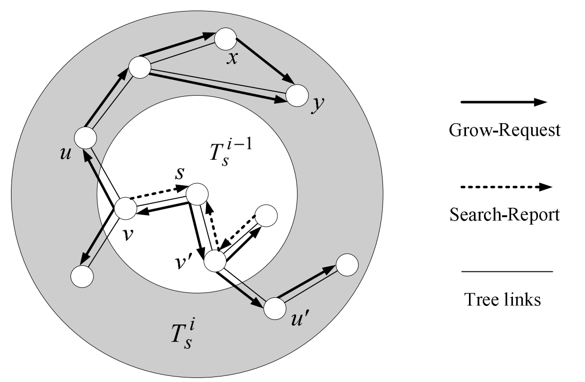


Figure 2. **Illustration of the *i*-th round of *Search-and-Grow* procedure**

After the tree-flooding of the Grow-Request messages, shown as the solid arrowed lines within the area T_s^{i-1} in Fig. 2, each tree node v would further forward Grow-Request to a non-tree node u if $w_{vu} \le \psi_w(C_{N(T_s^{i-1})})$. At each new included node, the node from which the first Grow-Request message is received will be set as the parent node and all subsequent duplicate Grow-Request messages are simply dropped (*e.g.* the message from x to y as shown in Fig. 2). Such message propagation, shown as the solid arrowed lines in the shaded area outside T_s^{i-1} in Fig. 2, will proceed until the incremental tree T_s^i ceases growing. If T_s^i does not span all destination nodes, a new round of Search-and-Grow procedure will start shortly. Eventually, a multicast forwarding tree is created by several Search-and-Grow cycles until all members join the tree. After that, the final min-max tree can be obtained straightforward by pruning all unnecessary links (the branches including non-member nodes only). Such post-procedure starts backwards from the non-member leaf nodes and performs in a distributed manner.

3.2. Message complexity of DMMT

Note that at each round of Search-and-Grow procedure, the tree grows by one additional node at least. Therefore, we can assume that there are exactly k ($1 \le k \le n-1$) rounds of Search-and-Grow procedure to achieve the final multicast tree.

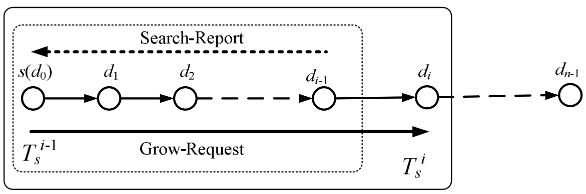


Figure 3. **Illustration of the message propagation in a special case for DMMT**

Now we consider a special case $k = n - 1$ for a message complexity analysis as shown in Figure 3, in which the final constructed tree is a directed chain staring from the source node s ($= d_0$) and connecting all the destination nodes $d_1, d_2, \ldots, d_{n-1}$ one by one. Figure 3 illustrates the i-th round of the Search-and-Grow iteration, in which the Search-Report messages propagate backwards from d_{i-1} to the source and the Grow-Request messages forwards from the source to d_{i-1} and further to d_i such that the tree is incremented from T_s^{i-1} to T_s^i by incorporating one more node d_i. The total number of messages to obtain the final tree is therefore

$$\sum_{i=1}^{n-1}((i-1)+i) = (n-1)^2 . \qquad (8)$$

3.3. The DMMT-EQ Algorithm

From the reinvestigation on the message complexity of the DMMT algorithm, we find that its message complexity could be at the order of $\Omega(n^2)$, which is obviously not scalable and motivates us to design new distributed algorithms with low message complexity, preferably in a linear complexity. From the description of the Search-and-Grow procedure in Fig. 1, we have the following observations that will be used for the new algorithm design.

Observation 1. Let T_s be the final tree obtained from the distributed DMMT algorithm. We observe that the sequence of the values $\psi_w(C_{N(T_s^{i-1})})$, $1 \le i \le k$, found in the multicast tree formation is in a strict increasing order and the final one in the sequence is equal to $\delta_w(T_s)$, *i.e.*

$$\psi_w(C_{N(T_s^0)}) < \ldots < \psi_w(C_{N(T_s^{k-1})}) = \delta_w(T_s) . \qquad (9)$$

From the above general observation, we can conclude that the weight sequence of arcs included into the tree by the DMMT algorithm satisfies the following condition in the case as shown in the Fig. 3,

$$w(d_0, d_1) < w(d_1, d_2) < \ldots < w(d_{n-2}, d_{n-1}),$$

which implies that the number of weights with unique value in the graph must be at least $n - 1$. In other words, if the number of unique weighs is less than $n - 1$, the case in Fig. 3 will not happen. ■

Observation 2. Suppose that the DMMT algorithm includes the links into the min-max spanning tree in the order of $(e_1, \ldots, e_{n-1})$, in which e_i is the i-th included link with the weight w_i. Note that the orders of the links that tie (*i.e.*, with the same weights) to be included into the tree at the same Search-and-Grow iteration of the algorithm may be arbitrary in the list $(e_1, \ldots, e_{n-1})$ [12]. Therefore, we can conclude that link e_i is the first one chosen to be included in a certain Search-and-Grow procedure, if and only if the variable x_i ($1 \le i \le n-1$) defined below is equal to one.

$$x_i = \begin{cases} 1 & w_j < w_i, \quad 1 \le j \le i-1 \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

Using this function, the number of the Search-and-Grow iterations, the determining factor of the message complexity, can be expressed as $\sum_{i=1}^{n-1} x_i$. ■

In order to reduce the message complexity, a good distributed algorithm with low message complexity would generate the list $(x_1, \ldots, x_{n-1})$ with many zero elements. From (10), a sufficient condition for satisfying $x_i = 0$ ($i > 1$) in a given list $(w_1, \ldots, w_{n-1})$ could be

$$\exists j,\ 1 \le j \le i-1 \text{ and } w_j = w_i. \qquad (11)$$

Both results in (9) and (11) from the above observations imply that an approach to reduce the number of unique arc weights in the given network would improve the distributed DMMT algorithm in terms of message complexity. A new algorithm, called DMMT-EQ (DMMT with Energy-supply Quantization) as the extension of the DMMT algorithm, redefines the weigh function as follows

$$w'_{vu} \equiv \frac{p_R}{\varepsilon_{\min} + \ell_v \cdot \Delta\varepsilon}, \text{ in which} \qquad (12)$$

$$\ell_v = \left\lfloor \frac{\varepsilon_v - \varepsilon_{\min}}{\Delta\varepsilon} \right\rfloor^2 \text{, and} \quad (13)$$

$$\Delta\varepsilon = \frac{\varepsilon_{\max} - \varepsilon_{\min}}{L}. \quad (14)$$

Note that there is only L number of values allowed for the new weight function w' by quantizing the energy supply at each node v to be a series of discrete values $\varepsilon_{\min} + \ell_v \cdot \Delta\varepsilon$ ($0 \le \ell_v < L$). The DMMT-EQ algorithm has the same iterative and distributed Search-and-Grow procedure as the DMMT algorithm, with the only variance that it runs under the new weight function w' defined in (12). Recalling that the DMMT algorithm generates the optimal solution [12], we have the following observation.

Observation 3. Given a network example, the final tree T_s obtained by DMMT-EQ is optimal under the weight function w', while not optimal in general under the original weight function w, *i.e.*

$$\delta_{w'}(T_s) = \delta^*_{w'}, \quad (15a)$$

$$\delta_w(T_s) \ge \delta^*_w. \ \blacksquare \quad (15b)$$

From the above observations, we can conclude that the DMMT-EQ algorithm must have a lower message complexity, but on the other hand it is a suboptimal algorithm in terms of proving longest-lived multicast trees. In the following sections, we shall quantitively study the tradeoff between the algorithm optimality and message complexity of the proposed algorithm.

4. Approximation ratio analysis

In this section, we shall first provide several fundamental lemmas that can be used to derive the upper bound of the approximation ratio for the heuristic algorithm DMMT-EQ. Let T_s be the tree obtained by the DMMT-EQ algorithm. Its approximation ratio ρ can be expressed as

$$\rho = \frac{\delta_w(T_s)}{\delta^*_w}. \quad (16)$$

Lemma 1. The following statements on the relationship between w and w' are satisfied.

$$\forall C_X, \quad w_{vu} = \psi_w(C_X) \Rightarrow w'_{vu} = \psi_{w'}(C_X) \quad (17)$$

$$\forall T_s, \quad \delta_w(T_s) \le \delta_{w'}(T_s) \quad (18)$$

[2] The symbol $\lfloor x \rfloor$ denotes the maximum integer that is not greater than x.

Proof: The results in Lemma 1 can be obtained straightforward from the definitions (2) and (12) and thus the detailed proof is omitted here. ■

Lemma 2. If $G(N, A)$ is connected, then for any weigh function w and any cut C_X,

$$\delta^*_w \ge \psi_w(C_X). \quad (19)$$

Proof: Note that there is at least one destination node z ($z \in D$) belonging to $N - X$ as shown in Fig. 4, *i.e.* $z \in N - X$, because $D \not\subset X$. Let T_s^* be a min-max tree of network G under the weigh function w. There must exist an arc $(v, u) \in A(T_s^*)$ connecting X and $N - X$ (*i.e.* $v \in X$ and $u \in N - X$) in order to satisfy that there must exist a directed path from s to the destination node z along the links in the tree T_s^*. Therefore, we can obtain $\delta_w^* = \delta_w(T_s^*) = \max\{w_{xy} \mid (x, y) \in A(T_s^*)\} \ge w_{vu} \ge \min\{w_{xy} \mid (x, y) \in C_X\} = \psi_w(C_X)$. ■

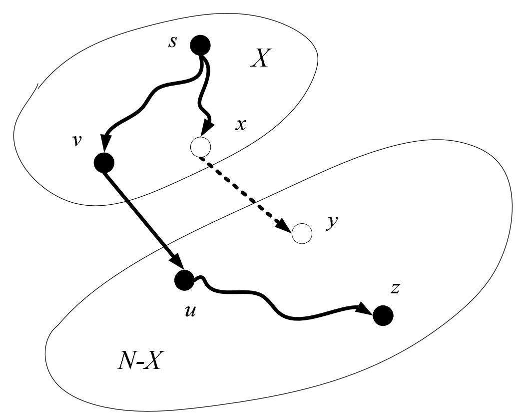


Figure 4. **Illustration of the proofs for Lemma 2 and Theorem 1**

Theorem 1. The DMMT-EQ algorithm is a constant-factor approximation algorithm.

Proof: Given a network $G(N, A)$, let T_s be the multicast tree obtained by the DMMT-EQ algorithm under the weight function w'. Suppose that there are exact k iterations of the Search-and-Grow procedure performed to achieve the final tree T_s as shown in Fig. 4, in which

$$X = N(T_s^{k-1}). \quad (20)$$

Let (x, y) be the link with minimum weight over the cut C_X under the function w, *i.e.*

$$w_{xy} = \psi_w(C_X). \quad (21)$$

The same link must be the one with minimum weight over C_X under the function w' as well, *i.e.*

$$w'_{xy} = \psi_{w'}(C_X) \quad (22)$$

that can be concluded from (17). Note that arc (x, y) is not necessary to be included in the tree T_s, shown as the dotted arrowed line in Fig. 4. Now we can derive by combining (18), (9) and (22) sequentially as follows.

$$\delta_w(T_s) \le \delta_{w'}(T_s) = \psi_{w'}(C_{N(T_s^{k-1})}) = w'_{xy} \tag{23}$$

On the other hand, we have

$$\frac{p_R}{\varepsilon_{\min} + (\ell_x + 1)\cdot\Delta\varepsilon} \le \frac{p_R}{\varepsilon_x} \le \frac{p_R}{\varepsilon_{\min} + \ell_x \cdot \Delta\varepsilon}, \tag{24}$$

from the definitions of w and w', or equivalently

$$\frac{\varepsilon_{\min} + \ell_x \cdot \Delta\varepsilon}{\varepsilon_{\min} + (\ell_x + 1)\cdot\Delta\varepsilon} \cdot w'_{xy} \le w_{xy} \le w'_{xy}. \tag{25}$$

We then can substitute (25) into (23) to achieve

$$\delta_w(T_s) \le w'_{xy} \le (1 + \frac{\Delta\varepsilon}{\varepsilon_{\min} + \ell_x \cdot \Delta\varepsilon}) \cdot w_{xy}. \tag{26}$$

Finally, combining (26), (21) and (19), we obtain

$$\begin{aligned}\delta_w(T_s) &\le (1 + \frac{\Delta\varepsilon}{\varepsilon_{\min} + \ell_x \cdot \Delta\varepsilon}) \cdot w_{xy} \\ &= (1 + \frac{\Delta\varepsilon}{\varepsilon_{\min} + \ell_x \cdot \Delta\varepsilon}) \cdot \psi_w(C_X). \\ &\le (1 + \frac{\Delta\varepsilon}{\varepsilon_{\min} + \ell_x \cdot \Delta\varepsilon}) \cdot \delta_w^*\end{aligned} \tag{27}$$

Therefore, the approximation ratio ρ of the DMMT-EQ algorithm can be express as

$$\begin{aligned}\rho = \frac{\delta_w(T_s)}{\delta_w^*} &\le 1 + \frac{\Delta\varepsilon}{\varepsilon_{\min} + \ell_x \cdot \Delta\varepsilon} \\ &= 1 + \frac{\varepsilon_{\max} - \varepsilon_{\min}}{L \cdot \varepsilon_{\min} + \ell_x \cdot (\varepsilon_{\max} - \varepsilon_{\min})}, \\ &= 1 + \frac{h-1}{L + \ell_x \cdot (h-1)}\end{aligned} \tag{28}$$

in which $h = \varepsilon_{\max}/\varepsilon_{\min}$ is a parameter that characterizes the dissimilitude of energy distribution over the network and the value of ℓ_x is determined by the tree generated from DMMT-EQ as shown in (24), (22) and (20). The theoretical upper bound of the approximation ratio ρ is therefore

$$\rho \le \mu_\rho \equiv 1 + \frac{h-1}{L + \ell_x \cdot (h-1)}. \tag{29}$$

Considering various cases, we further obtain

$$1 + \frac{h-1}{h \cdot L - h + 1} \le \mu_\rho \le 1 + \frac{h-1}{L + h - 1},\ 1 \le \ell_x < L; \tag{30a}$$

$$\mu_\rho \le 1 + \frac{h-1}{L},\ \ell_x = 0. \tag{30b}$$

When $\ell_x > 0$, we have $\mu_\rho < 2$ obtained from (30a), which is obviously bounded by a constant number. Recall that only nodes with relatively sufficient energy supply are allowed to be involved in the multicast tree, in other words, the h-value should be a bounded number, resulting in μ_ρ to be bounded by a small number given in (30b) as well when $\ell_x = 0$. Finally, we can conclude from both (30a) and (30b) that DMMT-EQ is a constant-factor approximation algorithm. ■

Corollary 1. The DMMT-EQ algorithm achieves the global optimal solution if $h = 1$ or $L = \infty$.

We can observe from (28) that to increase the value of L will improve the theoretical approximation ratio of the DMMT-EQ algorithm, its message complexity, however, would be exacerbated as analyzed in the following section.

5. Message complexity analysis

In order to study the upper bound of message complexity, we only need to consider the spanning tree case. The best message complexity is $O(n\log n + m)$ [13] for the min-max spanning tree problem in *undirected* graphs. In this section, we shall analyze the message complexity of DMMT-EQ for the same problem in more general *directed* graphs.

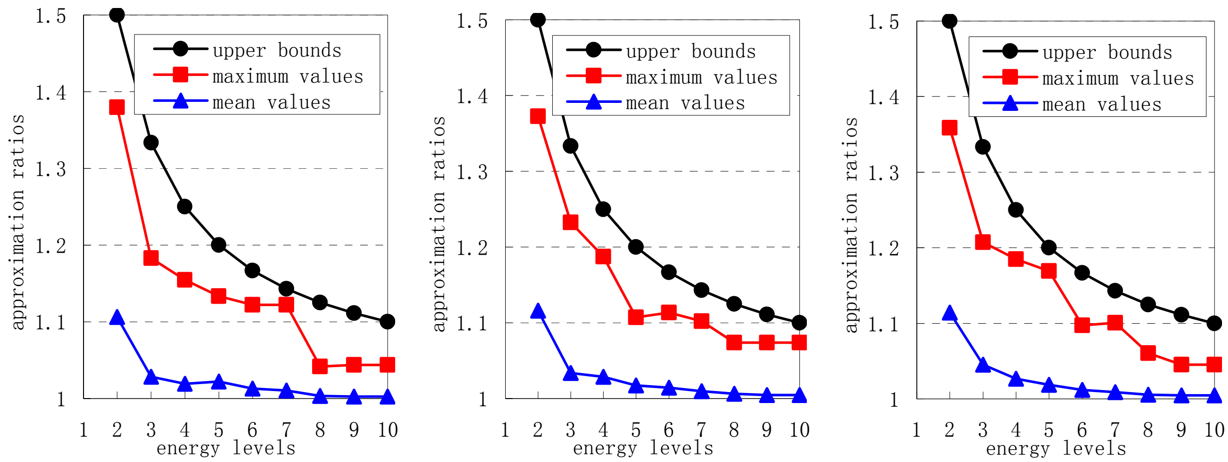
We consider the message interaction in the i-th round of Search-and-Grow procedure. It can be considered to consist of two components as shown in Fig. 2: (1) the tree-flooding of the messages (Search-Report and Grow-Request) in the area T_s^{i-1} and (2) the network-flooding of the messages (Grow-Request) in the area $T_s^i - T_s^{i-1}$. The number of messages of the corresponding components is denoted as c_t (tree-flooding) and c_n (network-flooding), respectively. Let n_i and m_i be the number of messages propagated within the areas T_s^{i-1} and $T_s^i - T_s^{i-1}$, respectively. Thus the message complexity c (*i.e.* the total number of Search-Report and Grow-Request messages) of the distributed algorithms using k rounds of Search-and-Grow procedure can be expressed as follows.

$$c = c_n + c_t \tag{31}$$

$$c_t = \sum_{i=1}^{k} n_i \le \sum_{i=1}^{k} 2\left|A(T_s^{i-1})\right| \tag{32}$$

$$c_n = \sum_{i=1}^{k} m_i \tag{33}$$

Recall that in the network-flooding, the Grow-Request messages are delivered only to non-tree nodes, which will join the tree and never receive such messages again in the later rounds.



(a) $n = 100$ (b) $n = 200$ (c) $n = 500$

Figure 5. **The approximation ratio as a function of *L* under various network sizes**

Therefore, throughout the whole distributed algorithm, the network-flooding Grow-Request message passes on each link at most once, resulting in

$$c_n = \sum_{i=1}^{k} m_i = \left|\{w'_{vu} \mid w'_{vu} \le \delta^*_{w'}\}\right| \le m \,. \tag{34}$$

The upper bound for the component c_n given in (34) is not related to the quantization level L. We now turn our attention to the more interesting task for a tradeoff analysis on the message complexity of component c_t in the worst case. Suppose that our distributed algorithm DMMT-EQ includes the links into the min-max spanning tree in the order of (e_1, …, e_{n-1}) with the corresponding list (x_1, …, x_{n-1}), in which variable x_i ($1 \le i \le n-1$) is defined in (10). The message complexity of component c_t in (32) can thus be rewritten as

$$c_t = \sum_{i=1}^{n-1} 2(i-1)x_i \,. \tag{35}$$

The upper bound of c_t can be obtained by letting the last L–1 elements in the list (x_1, …, x_{n-1}) be equal to one if $L < n$–1. In particular, we have

$$c_t \le \mu_c \equiv \begin{cases} 2(L-1)n-(L^2+L-2) & L < n-1; \\ n^2-3n+2 & \text{otherwise.} \end{cases} \tag{36}$$

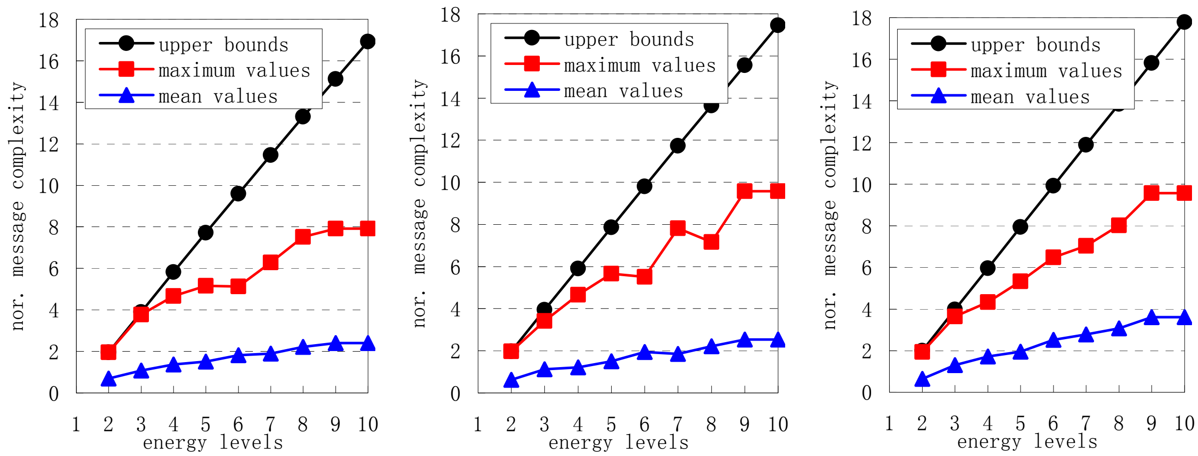
Considering the results in (34) and (36), we can conclude that the message complexity of the DMMT-EQ algorithm is at the order of $O(\min(L\cdot n, n^2) + m)$. Finally, the results in (30) and (36) show the tradeoff between the algorithm optimality of DMMT-EQ and its message complexity. In a typical scenario, a small value for L, *e.g.* L <10, would yield a linear message complexity $O(m)$ with an approximation ratio close to one.

6. Performance evaluation

In this section, we would like to evaluate the performances of the heuristic algorithm DMMT-EQ in terms of algorithm-optimality (its approximation ratio) and message complexity from an experimental perspective. The tradeoff analysis conducted in previous sections shall be verified by simulations as well.

In each network example, a number of sensor nodes (n = 100, 200 and 500) are randomly generated within a square region 10 × 10. The transmission range R is set as the value such that the number of neighbors of a sensor node is equal to 10 averagely over all sensor nodes. The initial energy supply at each node is evenly distributed between ε_{max} = 100 and ε_{min} = 50. Because we study the tradeoff between theoretical bounds (*i.e.* the extreme case), only the broadcast scenarios are considered. Note that all parameters above can be arbitrary units that are consistent with the units of distance.

The approximation ration defined in (16) is used to evaluate the real performance of the proposed DMMT-EQ algorithm. It shows how well it performs compared to the optimal solutions. A value closer to one means it performs near to the optimum. As explained in the previous section, only the message component c_t will be evaluated for a tradeoff analysis. In order to facilitate the comparison of message complexity under various network sizes, we use the metrics c_t/n, called the normalized message complexity, to evaluate the scalability of the DMMT-EQ algorithm.



(a) $n = 100$ (b) $n = 200$ (c) $n = 500$

Figure 6. **The normalized message complexity as a function of *L* under various network sizes**

The results given in Figs. 5 and 6 are based on 100 examples for each simulation setting. We observe that the correctness of our derived bounds holds for all cases. The tradeoff of these two performance metrics is also verified, *i.e.* using larger values of L will significantly improve the approximation ratio of the heuristic algorithm, while it would also incur higher message complexity. Another important observation is that the average performances, in terms of both metrics, are quite good even using small L (*e.g.* $L = 5$).

7. Conclusion

In this paper, we proposed a new distributed algorithm DMMT-EQ for the problem of longest-lived multicast communications in WSNs. We derive the theoretical bounds of its approximation ratio and message complexity. The tradeoff of these two performance metrics are investigated thoroughly from both theoretical and experimental perspectives.

8. References

[1] I. Kang and R. Poovendran, "On the lifetime extension of energy-efficient multihop broadcast networks", *World Congress on Computational Intelligence*, 2002.

[2] I. Kang and R. Poovendran, "Maximizing static network lifetime of wireless broadcast adhoc networks", *IEEE ICC*, Alaska, 2003, pp. 2256 -2261.

[3] A. K. Das, R. J. Marks II, et al, "MDLT: a polynomial time optimal algorithm for maximization of time-to-first-failure in energy-constrained broadcast wireless networks", *IEEE Globecom*, 2003, pp. 362 – 366.

[4] M. X. Cheng, J. Sun, and et al, "Energy-efficient broadcast and multicast routing in ad hoc wireless networks", *IEEE IPCCC*, 2003, pp.87 – 94.

[5] B. Wang and S. K. S. Gupta, "On maximizing lifetime of multicast trees in wireless ad hoc networks", *ICPP*, 2003, pp. 333 – 340.

[6] B. Floréen, P. Kaski, and et al, "Multicast time maximization in energy constrained wireless networks", *Workshop on Foundations of Mobile Computing*, 2003, pp. 50–58.

[7] L. Georgiadis, "Bottleneck multicast trees in linear time", *IEEE Communications Letters*, 7(11), 2003, pp. 564 – 566.

[8] S. Guo, V. Leung and O. Yang, "A scalable distributed multicast algorithm for lifetime maximization in large-scale resource-limited multihop wireless networks", *ACM IWCMC*, 2006, pp. 419 - 424.

[9] J. E. Wieselthier, G. D. Nguyen, et al, "Energy-limited wireless networking with directional antennas: the case of session-based multicasting", *IEEE INFOCOM*, 2002, pp. 190 - 199.

[10] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-aware wireless networking with directional antennas: the case of session-based broadcasting and multicasting", *IEEE Transactions on Mobile Computing*, 1(3), 2002, pp. 176-191.

[11] S. Guo and O. Yang, "Multicast lifetime maximization for energy-constrained wireless ad-hoc networks with directional antennas", *IEEE Globecom*, 2004, pp. 4120 – 4124.

[12] S. Guo, V. Leung and O. Yang, "Distributed multicast algorithms for lifetime maximization in wireless ad hoc networks with omni-directional and directional antennas", *IEEE Globecom*, 2006.

[13] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum weight spanning trees", *ACM Transactions on Programming Languages and Systems*, 5(1), 1983, pp. 66 – 77.