Efficient Distributed Steiner Tree Construction in Wireless Sensor Networks with Unreliable Links

***Abstract*—In wireless sensor networks (WSNs), multicast tree is the most common structure for the multicast process from one source to m receivers out of totally n nodes. Constructing the minimum multicast tree can be converted to the classical Steiner Tree Problem, which is proved to be NP-complete. In this paper, we for the first time explore how to efficiently construct the minimum multicast tree in WSNs with unreliable links, meaning that the communications may fail between node pairs in the transmission range. To this end, we design the Unreliable Links based Tree Construction Algorithm (ULTCA) in a distributed manner. Specifically, ULTCA involves two novel components to tackle the energy inefficiency and tree performance degradation caused by unreliable links: (1) The Self-adapted Search Protocol that enables each node to appropriately determine the transmission range and then to search neighbors at affordable energy cost. (2) The Connecting Backwards Strategy that can reduce the number of isolated receivers by broadening the selection choice of target receivers. While returning a tree length fairly close to the optimum (with the ratio being only 1.061), the ULTCA also exhibits desirable time and message complexity, which are $O(\sqrt{n})$ and $O\big(\frac{(\alpha+1)n}{\sqrt{\log n}} + \frac{m\log n}{(\alpha+1)^2}\big)$ respectively (α denotes the link unreliability), the best so far even compared with existing algorithms under reliable links. Remarkably, we also find two interesting and counterintuitive facts: (i) The time cost by ULTCA does not increase with the growth of unreliability. (ii) When the number of receivers is large, the message complexity will decrease as the links become more unreliable.**

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have wide application such as environmental evaluation, health monitoring and military surveillance [1]–[3], etc. Multicasting is one of the most common data transmission patterns in WSNs since it can efficiently support users to task and query from one source to multiple sensors.

Multicast trees are widely adopted in WSNs to achieve efficient multicasting due to its acyclic property, and the minimal multicast tree construction can be converted to the classic Steiner Minimum Tree (SMT) problem, which is proved to be NP-complete [4]. Many prior arts have investigated how to efficiently construct approximate Steiner tree under reliable links [5], [6], meaning that each node can successfully communicate with all others in the transmission range. However, in many realistic WSNs the links between node pairs usually turn out to be unreliable due to factors such as energy depletion and natural hazards [7]. Moreover, the farther apart two sensors are, the higher the possibility of communication failure. Consequently, the unreliability raises new challenges on multicast tree construction, primarily from two aspects: *energy efficiency* and *tree performance*. (i). The energy efficiency comprimises since the expected distance between a sender and notified nodes may be far smaller than an inappropriately determined transmission range. (ii). The tree performance might be degraded since the link failure will disable the paths along which multi-hop communications could have happened under stable links, and even worse, hinder receivers from being involved into multicasting.

To solve the aforementioned issues, we are motivated to construct the multicast tree successfully with high efficiency under unreliable links. Specifically, due to NP-completeness of such construction, we aim to construct the approximate Steiner tree that preserves comparable tree length to the optimal one with desirable time efficiency and low energy cost when facing the effect of unreliable links. To this end, we design Unreliable Links based Tree Construction Algorithm (ULTCA), which is based on a distributed framework where each multicast member progressively enlarges the **search** coverage area to find a target member to **connect** to.

Specifically, ULTCA consists of two components that correspondingly address the limitations of energy inefficiency and tree performance degradation as mentioned above. The first is the *Self-adapted Search Protocol* (Section IV-A), which can be adopted in the **search** step to enhance the energy efficiency. The core ides is to firstly design a self-adapted algorithm to determine the proper transmission range based on the local geographic information, and then let only a small part of notified nodes forward search messages to achieve the neighbor searching. The second is the *Connecting Backward Strategy* (Section IV-C), which can be embeded in the **connection** step to reduce the number of isolated members. This strategy tries to broaden the choice of communication paths by adaptively allowing members to connect to those who are located farther from the source without increasing the extra tree length.

We further quantitively analyze the tree length under general node distribution (Section V) and the time and energy performance of ULTCA (Section VI). We show that the tree formed by ULTCA has order-optimal tree length, and the approximation ratio is only 1.061 in empirical results. While enabling the tree length close to the optimum, the time and message complexity of ULTCA are $O(\sqrt{n})$ and $O(\frac{(\alpha+1)n}{\sqrt{\log n}} + \frac{m\log n}{(\alpha+1)^2})$ respectively, where α denotes the unreliability, and both are the lowest even compared with other algorithms under reliable links! Those analytical results are further evaluated by our extensive experiments (Section VII).

We also find and explain two counterintuitive facts (Section VI): (i) The running time of ULTCA, which is $O(\sqrt{n})$, does not increase with the growth of unreliability. (ii) When the number of receivers is large, the message complexity, which is $O(\frac{m\log n}{(\alpha+1)^2})$, decreases as the unreliability increases.

II. RELATED WORK

Multicast routing in wireless sensor networks is widely studied, and after geographic routing was introduced by Finn [8], a series of works have adopted this method into multicast

TABLE I
Comparison of Approximate Steiner Tree Construction Algorithms

Algorithm	Tree Length	Time	Messages
SPH [14]	$O(\sqrt{m})$	$O(m\frac{n}{\log n})$	$O(mn)$
DA [15]		$O(n^2)$	$O(mn^2)$
TST [6]		$O(\sqrt{n}\log n)$	$O(n + m\log n)$
$ULTCA$		$O(\sqrt{n})$	$O(\frac{(\alpha+1)n}{\sqrt{\log n}} + \frac{m\log n}{(\alpha+1)^2})$

tree construction in WSNs. Sanchez et al. introduced a fully-localized algorithm, Geographic Multicast Routing (GMR), to select neighbors heuristically [9]. Localized Energy-Efficient Multicast Algorithm (LEMA) was proposed based on MST algorithm [10]. Dijkstra-based Localized Energy-Efficient Multicast Algorithm (DLEMA) can discover the energy shortest path for each node [11]. Park et al. designed a distributed multicast protocol based on distributed geographic multicasting (DGM) and beaconless routing [12]. Koutsonikolas et al. incorporated geographic multicast routing (GMR) and hierarchical rendezvous point multicast (HRPM) to achieve better performance on transmission time and encoding overhead [13].

Besides, some works stated the multicast tree construction as Euclidean Steiner Tree problem. For example, GEographic Multicast (GEM) adopts the Steiner tree theory into constructing multicast tree for the first time [5]. Shortest Path Heuristic (SPH) constructs an approximate Steiner tree by adding nodes to the subtree [14]. Distributed Dual Ascent (DA) builds trees with high performance in point-to-point networks [15]. A work that shares the closest correlation with us belongs to [6], where a distributed Toward Source Tree (TST) algorithm is proposed with order-optimal multicast tree length as well as the lowest time and message complexity among all previous Steiner tree construction algorithms. However, all these algorithms can only be applied to reliable link scenarios, and will suffer the decrease of energy efficiency and the degradation of tree performance under unreliable links. Table I summarizes the performance comparison of these algorithms with our ULTCA, where α denotes the unreliability and satisfies $\alpha = o(\sqrt{\log n})$.

To sum up, although there are extensive works on multicast routing in WSNs, we have not found any minimum-length multicast tree construction algorithms under unreliable links.

III. Models and Assumptions

We assume that the network contains n nodes, which are distributed independently and identically in a unit square S. S can be of any dimension, and for ease of presentation, we let it be a two-dimensional square, i.e., $S = [0, 1]^2$. The distribution of nodes can be represented by the density function $f(\xi)$ where $\xi \in S$, and we assume $0 < \zeta_1 \le f(\xi) \le \zeta_2$. Each node can only obtain its own location unless notified by others.

In the multicast process, the source sends messages to m receivers, also called as *multicast members*, and we assume receivers are randomly distributed in the network. Since there are a huge number of sensors and receivers in sensor networks, we focus on the case where n and m are large in our analysis.

Two nodes share a link if their Euclidean distance is no larger than the largest transmission range r_t, and the links are unreliable, i.e., they suffer the possibility of failure. The farther apart two nodes are, the more likely the link fails. We model the unreliability as each link being on or off with a certain probability. The link with length r is on with probability $g(r)$, where $g(\cdot)$ is a monotone decreasing function. We suppose[1]

$$g(r) = \frac{1}{(cr+1)^\alpha}, \tag{1}$$

where c, α are parameters related to link unreliability[2]. Since link quality changes slowly [16], we assume it is unchanged when we construct the tree. To ensure the network connectivity when all links are on, we suppose the largest transmission range $r_t = \Theta(\sqrt{\frac{\log n}{n}})$ [17]. We set $c = \Theta(\sqrt{\frac{n}{\log n}})$ and vary α to represent different unreliability for ease of analysis.

IV. Algorithm Design of ULTCA

A. Self-adapted Search Protocol

1) ***Search Range Determination****:* Since each node only has limited topology knowledge, the first significant part in multicast tree construction is to enable each multicast member to search others within a coverage area through multi-hop relays. Suppose that both the member and relays transmit messages with a same one-hop search range r_s to notify their neighbors. An appropriate r_s is critical to the energy efficiency of tree construction in the sense that if r_s is too large, the link unreliablity will significantly reduce the proportion of notified nodes on the periphery of transmission area, whereas a too small r_s might cause additionally involved relays.

To find an appropriate r_s, we propose Self-adapted Search Range Determination Algorithm (SSRDA) in Algorithm 1. For any node N, we denote $\Xi_{[r_1,r_2]}$ as the annulus region centered by N with radii r_1, r_2, and $n_{[r_1,r_2]}$ as the number of notified nodes in $\Xi_{[r_1,r_2]}$. Intuitively, SSRDA searches the region $\Xi_{[\frac{r}{2},r]}$ with the most notified nodes among those in $\Xi_{[\frac{r}{4},r_t]}$ by varying the value of r, where r_t is the largest transmission range specified in Section III. As SSRDA proceeds in this prescribed way, it will return the most appropriate r_s after termination.

Algorithm 1: Self-adapted Search Range Determination Algorithm (SSRDA)

```
input : The locations of all nodes that can be notified by node N.
output: The search range r_s of N.
calculate distances {d} with notified nodes.;
r ← max_i{d_i};
while n_[r/2, r] < n_[r/4, r/2] do
    r ← r/2;
r_s ← r;
```

To rigorously evaluate the energy efficiency of data transmission with the search range determined by SSRDA, we first define the *transmission efficiency index* in Definition 1.

Definition 1 (Transmission Efficiency Index)**.** *For a node N, $\rho_1(k)$ (resp. $\rho_2(k)$) is the ratio of the number of existing nodes (resp. nodes that can be notified) in N's peripheral search*

[1] If g is not a power function, it can be converted to the sum of power functions by Taylor series, and thus can be analyzed by our theoretical tools.

[2] To represent a certain degree of unreliability, c and α should vary with the distances between a node and its neighbors, which will reduce as the network size grows in our model. Thus, the values of c and α may vary with n.

area to that in the whole search area. The peripheral search area is the annular region with radius kr_s and r_s, and k is a constant smaller than 1. If N is non-isolated, the transmission efficiency index $\mathcal{A}$ is $\mathcal{A} = \frac{\rho_2(k)}{\rho_1(k)}$, otherwise, $\mathcal{A} = 0$.

Based on the Law of Large Numbers, we have

$$\rho_2(k) = \mathcal{N}_{con}(k)/\mathcal{N}_{con} \to P'_{con}\mathcal{N}(k)/P_{con}\mathcal{N} \leq \rho_1(k), \quad (2)$$

where $\mathcal{N}, \mathcal{N}_{con}, P_{con}$ are the number of existing nodes, number of notified nodes and probability of a node being notified in search area. $\mathcal{N}(k), \mathcal{N}_{con}(k)$, P'_{con} are corresponding parameters in peripheral search area. Since $P'_{con} \leq P_{con}$, $\mathcal{A} \leq 1$ with probability 1, and we can regard $\rho_1(k)$ as the normalization factor. Theorem 1 reveals the value of r_s determined by SSRDA and its transmission efficiency index $\mathcal{A}$.

Theorem 1. *The search range determined by SSRDA and its energy efficiency are show as follows.*

The node N is isolated with probability p, where

$$p = \begin{cases} 0 & \alpha = o\left(\sqrt{\log n}\right) \\ \Theta(1) & \alpha = \Theta\left(\sqrt{\log n}\right) \\ 1 & \alpha = \omega\left(\sqrt{\log n}\right) \end{cases}. \quad (3)$$

When N is not isolated, the search range r_s determined by Self-adapted Search Range Determination Algorithm satisfies

$$r_s = \begin{cases} \Theta(\sqrt{\frac{\log n}{n}}) & \alpha = O(1) \\ \Theta(\frac{1}{\alpha}\sqrt{\frac{\log n}{n}}) & \alpha = \omega(1) \ \& \ \alpha = O\left(\sqrt{\log n}\right) \end{cases}, \quad (4)$$

and its transmission efficiency index satisfies $\mathcal{A} = \Theta(1)$.

The proof can be referred to Appendix A. From the theorem, we know that SSRDA ensures high energy efficiency of the transmission step, i.e., $\mathcal{A} = \Theta(1)$. Besides, since some nodes are isolated and can never be involved in multicasting when $\alpha = \Theta(\sqrt{\log n})$, the multicast tree only contains part of multicast members and cannot be constructed completely. When $\alpha = \omega(\sqrt{\log n})$, since almost all nodes are isolated, we do not consider this case in the following analysis.

*2) **Search Protocol Design**:* After determining the search range by SSRDA, we design a search protocol to further improve the energy performance of the search process. To reduce the number of relays, we only let those notified nodes located at the edge of the current searched area forward search messages. Based on this pattern, we design the *Self-adapted Search Protocol* (SASP), through which a sender N_s can search its target node(s) within the coverage range R_c. SASP can be summarized into the following four steps.

Step 1 N_s transmits a message with the largest transmission range r_t and all notified nodes send N_s their locations. Then N_s runs SSRDA to determine the search range r_s. Besides, N_s calculates r_{min}, the smallest distance to the notified nodes.

Step 2 N_s transmits the search message with r_s. Each notified node whose distance to N_s is larger than $r_s - r_{min}$ forwards the message with the same search range r_s. Messages should be transmited in sequence due to the interference.

Step 3 When receiving the search message(s), the node N within the coverage range will forward the search message if distances to nodes that send it messages are all larger than $r_s - r_{min}$ and N is farther from the sender N_s. When a target node is notified, it responds to N_s by sending a response message to its *immediate sender*, the node sending the search message that it first receives. Each relay in the response path responds to its *immediate sender* after receiving response messages from target nodes and relays that regard it as the *immediate sender*.

Step 4 If the first three steps do not find all the expected neighbors, N_s will transmit the search message again, and all nodes will forward the message when they first receive it.

The pseudo-code of Steps 2 & 3 is shown in Algorithm 2.

Algorithm 2: Message Forwarding in the Search Protocol

Input: Coverage range R_c; Target nodes set $\mathbb{T}$.
for *notified node N_u in the coverage area* **do**
  if *N_u is farther from N_s and the distances to the nodes sending messages to it are all larger than $r_s - r_{min}$* **then**
    Forward the *search message* with radius r_s;
  if $N_u \in \mathbb{T}$ **then**
    $N_v \leftarrow$ *immediate sender* of N_u;
    Send *response message* back to N_v;
    while $N_v \neq N_s$ **do**
      $N_t \leftarrow$ *immediate sender* of N_v;
      N_v responds to N_t after receiving responses from target nodes and relays regarding it as the *immediate sender*;
      $N_v \leftarrow N_t$;

Hereinafter, we first show the distances between a sender and its k-hop relays, the nodes forwarding the search messages that have been transferred $k-1$ times. Then we figure out whether SASP can search all target nodes.

Lemma 1. *In the* Self-adapted Search Protocol*, the distance between a sender and any of its k-hop relay approaches to kr_s, where r_s is the search range.*

The proof is put into Appendix B. According to the lemma, k-hop relays are all located in a circle with radius kr_s. Thus, the searched area in SASP presents a discretized version of a wave-like expansion, which is illustrated in Fig. 1.

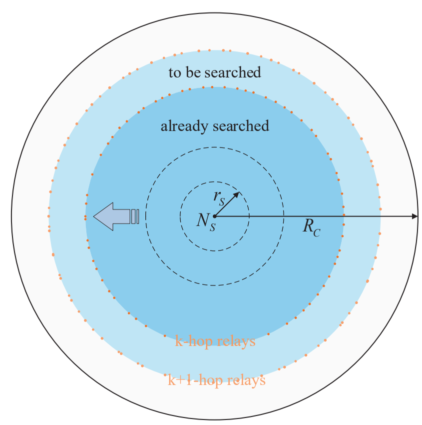


Fig. 1. Illustration of *self-adapted search protocol*

Theorem 2. *Through SASP, whether a sender can search its target nodes can be summarized as follows.*

(i). When $\alpha = o(\sqrt{\log n})$, only through the first three steps of SASP, the sender can successfully search any of its target node directly or through $\max\{\lfloor\frac{d_t}{r_s}\rfloor, 1\}$ relays with probability 1, where d_t is the distance between them.

(ii). When $\alpha = \Theta(\sqrt{\log n})$, the probability of a non-isolated target node being searched is $P = \Theta(1)$, and $P < 1$.

Proof. We prove the theorem based on the above two cases.

(1). When $\alpha = o(\sqrt{\log n})$, suppose one of N_s's target nodes N_t is located between k-hop and $(k+1)$-hop relays.

If $k = 0$, N_t can be notified by N_s directly or by 1-hop relays in Step 2 with probability P_0, where P_0 satisfies

$$P_0 = 1 - \int_0^{r_s} (1 - g(\gamma))^{n_k} \frac{\mathrm{d}P(d_t \leq \gamma)}{\mathrm{d}\gamma} \mathrm{d}\gamma > 1 - (1 - g(r_s))^{n_k},$$

where n_k is the number of 1-hop relays whose search areas contain N_t. Based on the Law of Cosines and the analysis of r_s and r_{min} in Appendices A and B, we can bound n_k by

$$n_k \geq 2r_{min} n \zeta_1 r_s \arccos \frac{d_t^2 + r_s^2 - r_s^2}{2d_t r_s} = \Theta\Big(\frac{\sqrt{\log n}}{\alpha}\Big) = \omega(1).$$

Since $g(r_s) = \Theta\big(\frac{1}{(1+\frac{1}{1+\alpha})^{\alpha}}\big) = \Theta(1)$, we can then get that $P_0 > 1 - o(1) \to 1$, and thus N_s can notify N_t directly or through 1 relays with probability 1.

If $k > 0$, with similar methods, we can get that N_t can be notified by k-hop relays in Step 3 with probability 1. Since $kr_s \leq d_t < (k+1)r_s$, N_s notifies N_t through $\lfloor \frac{d_t}{r_s} \rfloor = k$ relays.

(2). When $\alpha = \Theta(\sqrt{\log n})$, let S_j be the annulus area with radii $(j-1)\frac{r_s}{2}$ and $\frac{jr_s}{2}$, and its center is N_s. Hereinafter, we prove that any non-isolated node N_j in S_j can be searched by N_s with probability $P_n = \Theta(1)$ via mathematical induction.

If $j = 1$, $P_n \geq \int_0^{\frac{r_s}{2}} g(\gamma) \frac{\mathrm{d}P(r \leq \gamma)}{\mathrm{d}\gamma} \mathrm{d}\gamma = \Theta(1)$.

Suppose if $j_0 \leq j - 1$, any non-isolated node in S_{j_0} can be notified with probability $P_n' = \Theta(1)$. Let n_{j_0}' be the number of nodes in S_{j-1} whose distances with N_j are smaller than r_s. We can get that $\mathbb{E}[n_{j_0}'] = \Theta(\zeta_1 n r_s^2) = \Theta(1)$ and

$$P_n \geq 1 - (1 - g(r_s))^{P_n'(1 - P_{ISO}) n_{j_0}'} = \Theta(1),$$

where P_{ISO} is the probability of a node being isolated, and $P_{ISO} = \Theta(1)$ according to Theorem 1.

Above all, N_j can be searched with probability $P_n = \Theta(1)$.

Besides, N_j can be notified by N_s directly with probability $P < 1$. The probability that there exists any 1-hop relay is $P < 1 - P_{ISO}^{\pi \zeta_2 n r_s^2 g(r_s)} < 1$. Thus, we can get that $P_n < 1$. ☐

According to the theorem, only a small part of nodes forward messages when $\alpha = o(\sqrt{\log n})$, and thus the search process is energy-efficient. However, when $\alpha = \Theta(\sqrt{\log n})$, even all notified nodes forwarding messages will not find all non-isolated target nodes, which consolidates the necessity of the final step of SASP. Moreover, Theorem 2 also shows that adopting SSRDA to determine the transmission range will not generate isolated multicast members when $\alpha = o(\sqrt{\log n})$.

B. Connecting Backwards Strategy

After the search process, each multicast member should connect to its target member to form a tree. Since unreliable links limit the selection of communication paths as mentioned earlier, we attempt to enhance the tree performance by proposing the *Connecting Backwards Strategy* (CBS).

In CBS, each member can choose to connect forwards to its closest front member, i.e., the closest member closer to the source, or connect backwards to its closest back member, i.e., the closest member farther from the source. However, all members connecting to their target members simultaneously may form loops. For example, if a member N_1 connects forwards to N_2 and N_2 connects backwards to N_1, a loop will appear and they cannot communicate with the source. Thus we first let members tending to connect forwards take the connection process. Several connected components are formed after this step, and we denote *main connected component* as the one containing the source. Then other members reselect and connect to target members in the *main connected component* to form a tree. This step ensures messages to be transferred backwards for at most one time in communications between members and the source, and thus our strategy ensures desirable time performance of the multicasting. We summarize the *Connecting Backwards Strategy* as follows.

(i) Connection Step Each member calculates d_1, d_2, the distances to the closest front and back member respectively. If $\frac{d_1}{d_2} \leq \sqrt{m} \log m$, it connects forwards to the closest front member. Otherwise, it will not take any action in this step.

(ii). Reselection Step Each member not taking the connection process in the first step calculates d_1', d_2', the distances to the closest front and back member in the *main connected component* respectively. If $d_1' < d_2'$, the member connects forwards, and otherwise, it connects backwards.

Theorem 3 figures out the number of multicast members not in the *main connected component* after the Connection Step.

Theorem 3. *After the first step of CBS, the number of members not in the* main connected component *satisfies* $m_c = o(m)$.

We present the proof in Appendix C. From the theorem, we can get that the closest front and back members of the members taking the reselection process are in the *main connected component* with probability 1, which also ensures the rationality of the Reselection Step.

The advantage of the our strategy is twofold: (i). CBS reduces the number of isolated multicast members by increasing the probability of members successfully connecting to their neighbors since unreliable links or sensor failure may hinder members from connecting forwards. (ii). CBS can also help reduce the tree length, since it broadens the choice of communication paths by allowing nodes connecting both forward and backward depending on whether the link exists. In this way, more shorter paths are likely to be selected.

C. Whole Algorithm Design for ULTCA

Adopting *Self-adapted Search Process* and *Connecing Backwards Strategy* proposed in Sections IV-A and IV-B, we are now ready to design the whole Unreliable Links based Tree Construction Algorithm.

Phase 1: Notifying Multicast Members

The source sends the notification message via *Self-adapted Search Protocol*, in which the target node set contains all multicast members and the coverage area is the whole network.

Phase 2: Connecting Forwards

Each member searches its neighboring members via *Self-adapted Search Protocol* and then take the connection process by the first step of *Connecting Backwards Strategy*.

In the search process, the member N_s adopts SASP with the initial coverage range $R_c = r_s$, where r_s is the search range. If N_s searches any front member in the first three steps of SASP or $R_c > d_2\sqrt{m}\log m$, where d_2 is the distance to the closest back member, this process will be terminated. Otherwise, if coverage area contains the source, N_s will take the final step of SASP. If not, R_c will be doubled and N_s will take another search process. These steps are shown in Algorithm 3.

In the connection process, N_s takes the first step of CBS. If N_s decides to connect forwards, it will send the connection message to the closest front member N through the path that forwards N's response message in the search process. However, in order to avoid loops, if any relay in the path has transferred other connection message(s) before N_s's, it will transfer this message through the same path that transferred others even if the target member is not the original one.

Phase 3: Connecting Backwards

The source sends the identification message to members in the *main connected component* to mark them. Members not taking the connection process in Phase 2 take another search and connection process via SASP and the second step of CBS.

Algorithm 3: Search Process in Phase 2

for *multicast member* **do**
  Coverage range: $R_c \leftarrow r_s$;
  Target nodes set: $\mathbb{T} \leftarrow$ members in coverage area;
  Take the first three steps of SASP with input R_c, $\mathbb{T}$;
  Calculate d_2, the distance to the closest back member, and $d_2 \leftarrow \infty$ if no back members are found;
  if *no front members are found* & $R_c \leq d_2 m \log m$ **then**
    if *the distance to the source is smaller than* R_c **then**
      Take the final step of SASP;
    else
      $R_c \leftarrow 2R_c$;
      Take another search process;

Lemma 2 shows that ULTCA can construct a complete multicast tree when $\alpha = o(\sqrt{\log n})$.

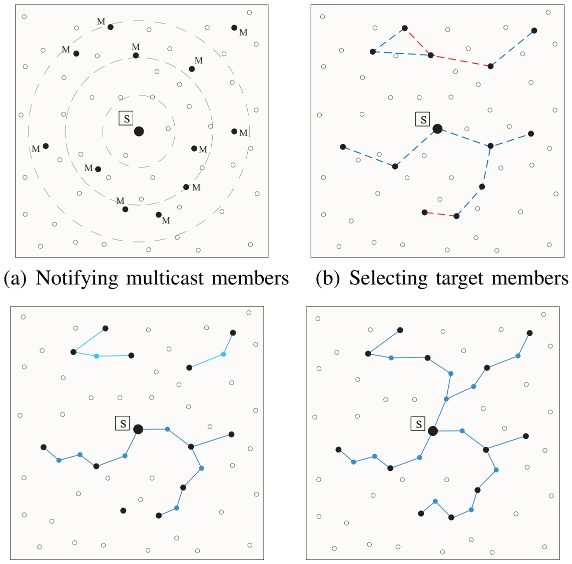
Lemma 2. *When $\alpha = o(\sqrt{\log n})$, the topology generated by ULTCA is a tree spanning all multicast members.*

Proof. When $\alpha = o(\sqrt{\log n})$, since all connected components connect to the *main connected component* in Phase 3, every member can find a path to the source and thus the topology ULTCA constructs is connected. Besides, assume the number of members and relays in the topology is n_t. Containing the source, the total number of nodes in the topology is $n_t + 1$. Since each member sends messages to one relay or directly to its target member and each relay transfers messages to only one node, the number of edges in the topology is n_t.

Above all, a connected topology with $n_t + 1$ nodes and n_t edges is a tree, and the tree spans all multicast members. □

When $\alpha = \Omega(\sqrt{\log n})$, although some members are isolated, the connected topology that contains the source is a tree. The proof is similar to that of Lemma 2.

Besides, we use an example to illustrate ULTCA in Figure 2. The bigger solid node labeled by "S" represents the source and others labeled by "M" in Figure 2(a) are multicast members.



(a) Notifying multicast members (b) Selecting target members
(c) Connecting forwards (d) Connecting backwards

Fig. 2. Illustration of ULTCA

Those hollow nodes are sensors that can be chosen as relays. After being notified by the source in Figure 2(a), multicast members take the search process and select target members in Figure 2(b). The blue or red dashed lines denote the virtual connections between members tending to connects forwards or backwards and their target members. Then in Figure 2(c), members tending to connect forwards take the connection process. Four connected components are formed and the one in dark blue is the *main connected component*. Finally, other members reselect the target members in the *main connected component* and take the connection process in Figure 2(d).

V. Multicast Tree Length Analysis

The previous section described our ULTCA algorithm, which, as Lemma 2 demonstrates, generates a multicast tree topology. In this section, we continue to analyze the corresponding tree length, which is presented in Theorem 4.

Theorem 4. *When $\alpha = o(\sqrt{\log n})$, the expected length of the multicast tree constructed by ULTCA is upper bounded by $c_t\sqrt{m}\int_{\xi\in S}\sqrt{f(\xi)}\mathrm{d}\xi$, where $c_t = 3.386$.*

Proof. We first figure out the expected path length $\mathbb{E}[l_{s,t}]$ between an arbitrary member N_s and the member N_t it connects to, and then calculate the expected tree length under general node distribution by summing all path lengths. According to Theorem 2, $l_{s,t}$ converges to the distance between N_s and N_t.

Denote L_1 as the expected path length generated in Phase 2 of ULTCA. It represents the expected distance between N_s and its closest front member N_t, and $L_1 < bL_1'$, where $b = \sqrt{m}\log m$ and L_1' is the distance between N_s and its closest back member. Denote L_2, L_3 as the expected path lengths for connecting forwards and backwards in Phase 3 respectively. In other words, L2 (resp. L3) represents the expected distance between N_s and its closet front (resp. back) member N_t in the *main connected component*. Let P_c be the probability of N_s joining Phase 3 in ULTCA. We can derive that $P_c = \Theta(\frac{1}{b})$

by the proof in Appendix C. Then the expected tree length of ULTCA satisfies $\mathbb{E}[l_{s,t}] = L_1 + P_c L_2 + P_c L_3$, where

$$L_1 = \int_0^{D_{N_s}} x(1 - \int_{S'_{\frac{x}{b}}} f(\xi)\mathrm{d}\xi)^{m-1} \frac{-\mathrm{d}(1-\int_{S_x} f(\xi)\mathrm{d}\xi)^{m-1}}{\mathrm{d}x}\mathrm{d}x,$$
$$L_2 = \int_0^{D_{N_s}} x(1 - \int_{S'_x} f'(\xi)\mathrm{d}\xi)^{m'} \frac{-\mathrm{d}(1-\int_{S_x} f'(\xi)\mathrm{d}\xi)^{m'}}{\mathrm{d}x}\mathrm{d}x,$$
$$L_3 = \int_0^{D_{N_s}} x(1 - \int_{S_x} f'(\xi)\mathrm{d}\xi)^{m'} \frac{-\mathrm{d}(1-\int_{S'_x} f'(\xi)\mathrm{d}\xi)^{m'}}{\mathrm{d}x}\mathrm{d}x.$$

D_{N_s} is the distance between N_s and the source. S_x (resp. S'_x) is the region where N_s's closest front (resp. back) member lies if its distance to N_s is smaller than x. $f'(\xi)$ is the distribution of totally m' members in the *main connected component*.

We equally divide the whole region S into squares with side length $\Theta(\sqrt{\frac{\log m}{m}})$ and suppose N_s is located in square S_a. When $x = O(\frac{1}{\sqrt{m}})$, S_x and S'_x will cover no more than four squares. Since $4\int_{S_a} \mathrm{d}\xi = o(\int_S \mathrm{d}\xi)$, the distribution of nodes in S_x and S'_x can be approximated by a uniform distribution with the density function being $\overline{f_a(\xi)} = \frac{\int_{S_a} f(\xi)\mathrm{d}\xi}{\int_{S_a}\mathrm{d}\xi}$. Then we calculate L_1, L_2, L_3 one by one as follows.

As for L_1, it is $o(\frac{1}{\sqrt{m}})$ when $x = \omega(\frac{1}{\sqrt{m}})$, and thus

$$\begin{aligned} L_1 &\to (m-1)\overline{f_i(\xi)}\pi \int_0^{\Theta(\frac{1}{\sqrt{m}})} x^2(1-\frac{1}{2}\pi x^2 \overline{f_i(\xi)})^{m-2}\mathrm{d}x \\ &< m\overline{f_i(\xi)}\pi \sum_{k=1}^{+\infty} \int_{\sqrt{\frac{k-1}{m\overline{f_i(\xi)}}}}^{\sqrt{\frac{k}{m\overline{f_i(\xi)}}}} x^2 e^{-\frac{(k-1)\pi}{2}}\mathrm{d}x \\ &< \frac{\pi}{2\sqrt{m\overline{f_i(\xi)}}} \sum_{k=1}^{+\infty} \lceil\sqrt{k}\rceil e^{-\frac{(k-1)\pi}{2}} \\ &< \frac{(1+\frac{\sqrt{2}}{2})\pi}{2(1-e^{-\frac{\pi}{2}})\sqrt{m\overline{f_i(\xi)}}} < 3.385\sqrt{\frac{1}{m\overline{f_i(\xi)}}}. \end{aligned}$$

As for L_2, we have

$$P_c L_2 \le P_c \int_0^{D_{N_s}} m'\pi\zeta_2 x^2 (1-\frac{1}{2}\pi\zeta_2 x^2)^{2m'-1}\mathrm{d}x = o\Big(\sqrt{\frac{1}{m}}\Big).$$

Similarly, we can get $P_c L_3 = o(\sqrt{\frac{1}{m}})$.

Thus, we can get that $\mathbb{E}[l_{s,t}] = L_1 + P_c L_2 + P_c L_3 \le 3.385\sqrt{\frac{1}{m\overline{f_i(\xi)}}} + o\big(\sqrt{\frac{1}{m}}\big) < c_t\sqrt{\frac{1}{m\overline{f_i(\xi)}}}$, where $c_t = 3.386$.

Then for the expected length of the tree T, we can get

$$\begin{aligned} \mathbb{E}[L_T] &= \sum_{e_{s,t}\in T} \mathbb{E}[l_{s,t}] < c_t \sum_{S_i\subset S}\sum_{N_s\in S_i} \sqrt{\frac{1}{m\overline{f_i(\xi)}}} \\ &= c_t \sum_{S_i\subset S} m\int_{S_i} f(\xi)\mathrm{d}\xi \sqrt{\frac{\int_{S_i}\mathrm{d}\xi}{m\int_{S_i} f(\xi)\mathrm{d}\xi}} \\ &= c_t\sqrt{m}\sum_{S_i\subset S}\sqrt{\int_{S_i}\mathrm{d}\xi \int_{S_i} f(\xi)\mathrm{d}\xi} \\ &\le c_t\sqrt{m}\sum_{S_i\subset S}\int_{S_i}\sqrt{f(\xi)}\mathrm{d}\xi = c_t\sqrt{m}\int_S \sqrt{f(\xi)}\mathrm{d}\xi. \end{aligned}$$

□

Remark: It has been derived in [18], [19] that the length of the minimum spanning tree that spans m nodes is $L_m = 0.656\sqrt{m}\int_S \sqrt{f(\xi)}\mathrm{d}\xi$ and is around $\frac{2}{\sqrt{3}}$ times of the Stenier tree length L_s [20]. Accordingly, we can deduce that our multicast tree length L_T satisfies $L_T < \frac{3.386}{0.656\frac{\sqrt{3}}{2}} L_s < 6L_s$. Thus ULTCA returns an order optimal tree length. Actually, as we will shortly demonstrate in Section VII, we can obtain an even better tree length that is fairly close to the optimum.

VI. Algorithm Performance Analysis

We further analyze running time and energy consumption of ULTCA. Again, we only consider the case where the tree can be fully constructed, i.e., $\alpha = o(\sqrt{\log n})$ shown in Lemma 2.

A. Time Complexity

Theorem 5. *When $\alpha = o(\sqrt{\log n})$, the time complexity of ULTCA is $O(\sqrt{n})$.*

Proof. We analyze the time complexity phase by phase.

In Phase 1, the source transmits a message in SSRDA. The number of notified nodes is $O(\pi\zeta_2 n r_t^2) = O(\log n)$ and thus the waiting time for a node to make response is at most $t_r = O(\log n)$. In the notification process, a k-hop relay ($k < \frac{1}{r_s}$) transfers the search or response message after at most all relays whose distances to it are smaller than $2r_s$ transferring messages, and thus the time of notification process is

$$t_n = O(4\pi\zeta_2 n r_s r_{min}) = O\left(\sqrt{n} r_s\right).$$

Then the time cost in Phase 1 is $t_1 = t_r + t_n = O(\sqrt{n})$.

In Phase 2, we can get that the time cost by SSRDA and sending response or conection messages is $t_d = O(\sqrt{n})$. In the i^{th} search process, the coverage range is $2^{i-1}r_s$. The number of search processes is $i_{max} = O(\log_2 \frac{1}{r_s})$, and thus the total time of the search process is

$$t_s = O(\sum_{i=1}^{i_{max}} 2^{i-1}\sqrt{n} r_s) = O(\sqrt{n}).$$

Then the time cost in Phase 2 is $t_2 = O(t_d + t_s) = O\left(\sqrt{n}\right)$.

In Phase 3, the time of identification process satisfies $t_m \le t_1$ and the time of the search and connection process is $t_c \le t_2$ since only a small part of nodes take the connection process. Then the time cost in Phase 3 is $t_3 = t_m + t_c = O(\sqrt{n})$.

Hence, the total time cost is $t = t_1 + t_2 + t_3 = O(\sqrt{n})$. □

From Theorem 5, we know that the time complexity will not vary with the unreliability. Although more relays are needed to transfer messages when unreliability grows, the search range will decrease and thus the time nodes should wait to avoid the interference before transmitting messages will be reduced. Combining these two factors, the running time will not change.

B. Message Complexity

Energy performance is the primary part in WSNs, and it is measured by the number of transmitted messages, known as *message complexity*.

Theorem 6. *When $\alpha = o(\sqrt{\log n})$, the message complexity of ULTCA is $O(\frac{(\alpha+1)n}{\sqrt{\log n}} + \frac{m\log n}{(\alpha+1)^2})$.*

The proof is deferred to Appendix D. From Theorem 6, an interesting phenomenon occurs at the boundary value $m_0 = \frac{\alpha^3 n}{\log^{\frac{3}{2}} n}$. When $m = O(m_0)$, the message complexity is $O(\frac{(\alpha+1)n}{\sqrt{\log n}})$, which increases as α grows. Once $m = \omega(m_0)$, the message complexity yields to $O(\frac{m\log n}{(\alpha+1)^2})$, which, surprisingly, decreases as α grows. The reason behind is that when m is small, the number of search messages dominates, and larger unreliability causes smaller search range, which induces more relays and thus more transferred messages. However, as m grows, the messages transmission part in SSRDA dominates. Under such case, with larger unreliability, fewer nodes are notified, thus reducing the response message sending in SSRDA and leading to decreased message complexity of ULTCA.

VII. Experimental Validations

In this section, we conduct the experimental validations on the performance of ULTCA and the constucted multicast tree.

A. Experiment Setup

1) Experimental Dataset: In our simulation, we fix the network size as 30000 and consider the normal distribution pattern with density function $f(\xi) = \frac{6}{\sqrt{2\pi}} e^{-18\|\xi-\xi_c\|^2}$, where ξ_c is the center of the unit square and $\|\xi - \xi_c\|$ is the Euclidean distance between the node and the center point. We list the parameters involved in our simulation in Table II. Note that for ease of presentation, we also define $\kappa = \log_{\ln n} \alpha$ to represent the degree of unreliability.

TABLE II
Main Experimental Parameters

Notation	Definition	Value
n	Number of Nodes	30000
m	Number of Multicast Members	300-30000
r_t	Maximum Transmission Range	$2\sqrt{\frac{\ln n}{n}}$
c	Parameter Related to the Degree of Unreliability	$\sqrt{\frac{n}{\ln n}}$
α	Parameter Related to the Degree of Unreliability	0-3 $\ln n$
κ	Parameter Related to α	$\log_{\ln n} \alpha$

Besides, we validate the algorithm performance under the topology of GreenOrbs, a long-term kilo-scale wireless sensor network for ecological surveillance in Tianmu Mountain with 1296 nodes, and the topology varies with time due to the changes of link states.

2) Performance Metrics and Baseline Algorithms: We evaluate tree length and validate message complexity by recording the total exchanged messages. We choose the TST algorithm for comparison, since it can construct the low-cost multicast tree with the lowest message complexity among all previous Steiner tree construction algorithms. We also illustrate the real tree length in GreenOrbs and the Steiner tree length, which can be approximately obtained by the heuristic algorithm in [21].

3) Supplementary Experiments: To further evaluate the effects of *Self-adapted Search Range Determination Algorithm* (SSRDA) and *Connecting Backwards Strategy* (CBS) on constructing the tree, we: (i) study the energy efficiency of the search range determined by SSRDA and compare it with that of TST; (ii) compare the number of isolated members caused by ULTCA and ULTCA-Forward, the ULTCA without CBS.

B. Experiment Results

1) Message Complexity: Figure 3 illustrates the message complexity under different unreliability. In Figures 3(a)-3(c), we set $\alpha = 0, (\ln n)^{\frac{1}{6}}, (\ln n)^{\frac{5}{12}}$ respectively. As can be seen, the message complexity of ULTCA is far less than that of TST in all cases, and when $\alpha = (\ln n)^{\frac{5}{12}}$, it is about 10 times lower than TST's, which shows the superiority of the *Self-adapted Search Protocol* on enhancing energy efficiency. Besides, in Figure 3(d), we set $\kappa = 0, \frac{1}{3}, \frac{5}{12}$, where $\kappa = \log_{\ln n} \alpha$. We can observe that as the unreliability grows, the message complexity of TST increases while that of ULTCA decreases, which is in line with our analysis. Thus, the energy efficiency of ULTCA will be much higher than TST when links are more unreliable.

2) Tree Performance: Figure 4 shows the tree length under different unreliability. From Figures 4(a)-4(c), we can observe that the tree length increases with m and the curves indicate that it agrees with $O(\sqrt{m})$, which echoes our theoretical result. Besides, the tree length of ULTCA is lower than that of TST in most scenarios and the gap between them increases with α. Moreover, our tree length is closer to the approximate Steiner tree as α grows and when $\alpha = (\ln n)^{\frac{5}{12}}$, with the ratio between them being only 1.061. Figure 4(d) illustrates the tree length as κ varies. We can observe that the unreliability causes the increase of tree length, and the gap between two curves grows with κ, demonstrating the effect of *Connecting Backwards Strategy* on reducing tree length especially when α is large.

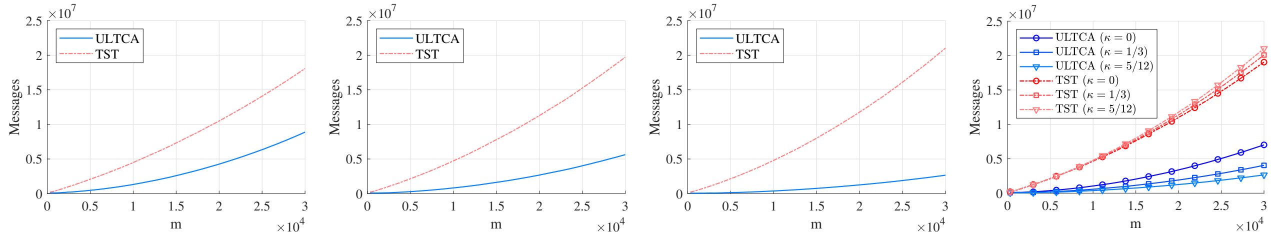
3) Algorithm and Tree Performance in GreenOrbs: With $m = 280$, Figure 5 shows the message complexity and tree length under the topology of GreenOrbs at different time. As can be seen, ULTCA incurs much fewer transmitted messages than TST, and ULTCA is less subject to the unreliability changes. Besides, the tree length of ULTCA is smaller than that of TST and the real length in GreenOrbs. These results indicate the superiority of ULTCA in practical scenarios.

4) Energy Efficiency of SSRDA: With $m = 27000$, Figure 6(a) illustrates the transmission efficiency index $\mathcal{A}$ of non-isolated members, and the parameter k that determines the pheripheral area is set as 0.75. We can observe that when $\kappa \leq 0$, $\mathcal{A}$ is close to 1 and the values in ULTCA and TST are roughly the same. When $\kappa > 0$, $\mathcal{A}$ drops sharply in TST while it maintains to be larger than 0.6 until all membes become isolated in ULTCA. These phenomena show the prominent effect of SSRDA on enhancing the energy efficiency.

5) Effect of CBS on Reducing Isolated Members: Figure 6 shows the number of isolated members in ULTCA and ULTCA-Forward when $m = 27000$. We can observe that almost no isolated member is generated when $\alpha < \sqrt{\log n}$, indicating that the tree can be completely constructed. As α further grows, isolated members increases sharply. When $\alpha = 4\sqrt{\log n}$, more than half of members cannot be involved in the tree and when $\alpha = \omega(\sqrt{\log n})$ ($\alpha > 10\sqrt{\log n}$), almost all members become isolated. All these phenomena are exactly in line with our analysis. Moreover, the number of ioslated members in ULTCA is smaller than that in ULTCA-Forward and the gap is large when $\alpha = \Theta(\sqrt{\log n})$, which demonstrates the prominent effect of CBS on reducing isolated members.

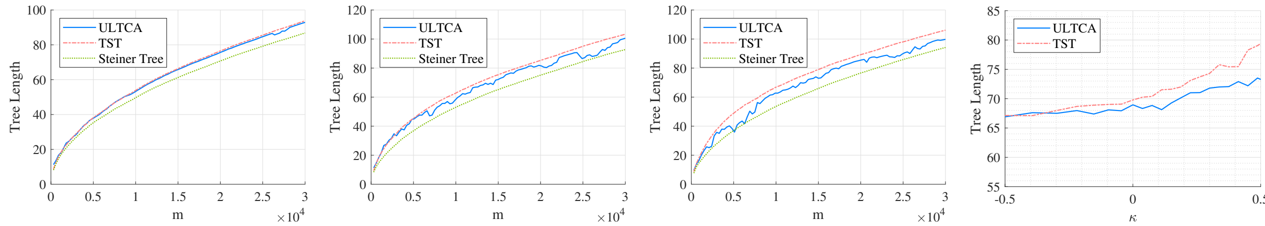
VIII. Conclusion

In this paper, we propose the Unreliable Links based Tree Construction Algorithm (ULTCA) to construct approximate Stenier trees under unreliable links. Adopting the *Self-adapted Search Process* and *Connecting Backwards Strategy*, ULTCA can energy-efficiently construct distributed multicast trees with desirable performance in wireless sensor networks. We prove



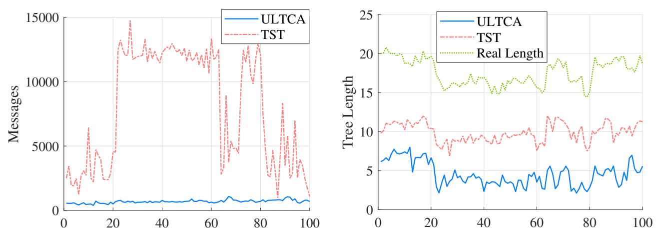
(a) $\alpha = 0$ (b) $\alpha = (\ln n)^{\frac{1}{6}}$ (c) $\alpha = (\ln n)^{\frac{5}{12}}$ (d) $\alpha = 1, (\ln n)^{\frac{1}{3}}, (\ln n)^{\frac{5}{12}}$

Fig. 3. Message Complexity vs. Multicast Group Size under Different Degrees of Unreliability (Parameterized by α)



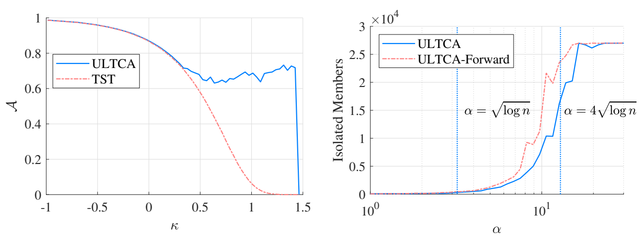
(a) $\alpha = 0$ (b) $\alpha = (\ln n)^{\frac{1}{6}}$ (c) $\alpha = (\ln n)^{\frac{5}{12}}$ (d) $m = 15000$

Fig. 4. Tree Length vs. Multicast Group Size under Different Degrees of Unreliability (Parameterized by α or κ)



(a) Message Complexity (b) Tree Length

Fig. 5. Message Complexity & Tree Length in GreenOrbs



(a) Energy efficiency of search range (b) Isolated members comparation

Fig. 6. Supplementary Experiments

that the time and message complexity of ULTCA are the lowest compared with all existing algorithms even under reliable links. We show that the tree length is in the same order as Steiner tree, and the gap between them is small in practice. We find and explain two counterintuitive facts about the time and message complexity.

Appendix A
Proof of Theorem 1

Let S be the whole region, Ξ_t be the largest transmission area. Since $\frac{\int_{\Xi_t} \mathrm{d}\xi}{\int_S \mathrm{d}\xi} = \pi r_t^2 \to 0$, we can get $\int_{\Xi_t} f(\xi)\mathrm{d}\xi \to f(\xi_0)\int_{\Xi_t}\mathrm{d}\xi = \zeta_0 \int_{\Xi_t}\mathrm{d}\xi$, where $\xi_0 \in \Xi_t$, and nodes in Ξ_t are approximately uniformly distributed. Let B be the event that N is isolated and P_{con} be the probability that a node within Ξ_t is connected with N. We can get $P_{con} = \int_0^{r_t} g(\gamma)\frac{\mathrm{d}P(r\le\gamma)}{\mathrm{d}\gamma}\mathrm{d}\gamma$ where $P(r \le \gamma) = \frac{\gamma^2}{r_t^2}$. Thus, we can derive $P(B)$ as

$$P(B) = e^{-\pi\zeta_0 r_t^2 n P_{con}} = e^{-2\pi\zeta_0 n \int_0^{r_t} g(\gamma)\gamma \mathrm{d}\gamma}. \quad (5)$$

Denote $G(r_t) = \int_0^{r_t} g(\gamma)\gamma\mathrm{d}\gamma$, and in order to figure out $P(B)$, we first clarify $G(r_t)$. When $\alpha \neq 1, 2$, we can get

$$G(r_t) = \frac{(cr_t+1)^{2-\alpha}(1-\alpha) - (cr_t+1)^{1-\alpha}(2-\alpha) + 1}{c^2(1-\alpha)(2-\alpha)}. \quad (6)$$

When $\alpha = o(\sqrt{\log n})$ and $\alpha \neq 1, 2$, we can derive that

$$2\pi\zeta_0 n G(r_t) = \Theta\Big(\frac{n}{c^2\alpha^2}\big(1 + \frac{cr_t(1-\alpha)-1}{(cr_t+1)^{\alpha-1}}\big)\Big) = \Theta\big(\frac{\log n}{\alpha^2}\big) = \omega(1).$$

Therefore, $P(B) = e^{-2\pi\zeta_0 n G(r_t)} \to 0$. Since $P(B|\alpha = 1, 2) < P(B|\alpha = 3)$, $P(B) \to 0$ when $\alpha = o(\sqrt{\log n})$.

Similarly, $2\pi\zeta_0 n G(r_t) = \Theta\big(\frac{\log n}{\alpha^2}\big) = \Theta(1)$ when $\alpha = \Theta(\sqrt{\log n})$ and thus $P(B) = \Theta(1)$.

When $\alpha = \omega(\sqrt{\log n})$, we can get that $P(B) \to 1$.

After figuring out $P(B)$, we calculate the search range r_s determined by SSRDA when node N is non-isolated. Suppose when $\alpha = O(1)$, $r_s = o(r_t)$. Since

$$\frac{(1-\frac{1}{4})\pi\zeta_0 r_s^2 n}{(\frac{1}{2}cr_s+1)^\alpha} > n_{[\frac{r_s}{2}, r_s]} > n_{[\frac{r_t}{2}, r_t]} > \frac{(1-\frac{1}{4})\pi\zeta_0 r_t^2 n}{(cr_t+1)^\alpha},$$

we can get that $\left(\frac{r_s}{r_t}\right)^2 > \left(\frac{\frac{1}{2}cr_s+1}{cr_t+1}\right)^\alpha$. However, since $\left(\frac{\frac{1}{2}cr_s+1}{cr_t+1}\right)^\alpha = \Theta(1)$, we have $\left(\frac{r_s}{r_t}\right)^2 < \left(\frac{\frac{1}{2}cr_s+1}{cr_t+1}\right)^\alpha$, which causes contradiction. Therefore, we can get that when $\alpha = O(1)$, $r_s = \Theta(r_t) = \Theta\Big(\sqrt{\frac{\log n}{n}}\Big)$.

When $\alpha = \omega(1)$, suppose $r_s = \frac{\lambda}{\alpha}\sqrt{\frac{\log n}{n}}$ and $c = \varphi\sqrt{\frac{n}{\log n}}$ where $\varphi = \Theta(1)$. Let $r_0 = \frac{1}{\alpha}\sqrt{\frac{\log n}{n}}$, then $r_s = \lambda r_0$ and

$$n_{[\frac{\lambda}{2}r_0, \lambda r_0]} = \frac{3\pi\zeta_0}{4}\lambda^2 r_0^2 \int_{\frac{\lambda}{2}r_0}^{\lambda r_0} g(\gamma)\frac{P(r\le\gamma)}{\mathrm{d}\gamma}\mathrm{d}\gamma = 2\zeta_0\int_{\frac{\lambda}{2}r_0}^{\lambda r_0}\gamma g(\gamma)\mathrm{d}\gamma$$

$$= \int_{\frac{\lambda}{2}r_0}^{\lambda r_0}\frac{2\zeta_0\gamma}{(\frac{\varphi\gamma}{\alpha r_0}+1)^\alpha}\mathrm{d}\gamma \to 2\zeta_0\int_{\frac{\lambda}{2}r_0}^{\lambda r_0}\gamma e^{-\frac{\gamma}{r_0}\varphi}\mathrm{d}\gamma = 2\zeta_0 r_0^2\int_{\frac{\lambda}{2}}^{\lambda}\varrho e^{\varrho\varphi}\mathrm{d}\varrho,$$

where $\varrho = \frac{\gamma}{r_0}$. Let $t = \lambda\varphi$ and

$$\mathcal{G}(t) = \int_{\frac{\lambda}{2}}^{\lambda}\varrho e^{-\varrho\varphi}\mathrm{d}\varrho = \frac{1}{\varphi^2}\Big((1+\frac{t}{2})e^{-\frac{t}{2}} - (1+t)e^{-t}\Big).$$

Since $\mathcal{G}(2t) \ge \mathcal{G}(t) > \mathcal{G}(\frac{t}{2})$, we can get $2\ln 2 < t < 6\ln 2$ and therefore $\lambda = \Theta(1)$. Thus, $r_s = \Theta(\frac{1}{\alpha}\sqrt{\frac{\log n}{n}})$.

As for the transmission efficiency index $\mathcal{A}$, we have

$$\mathcal{A} = \frac{\frac{\mathcal{N}_{con}(k)}{\mathcal{N}_{con}}}{\frac{\mathcal{N}(k)}{\mathcal{N}}} \to \frac{1 - \frac{\pi(kr_s)^2\zeta_0 n\int_0^{kr_s} g(\gamma)\frac{\mathrm{d}P(r\le\gamma)}{\mathrm{d}\gamma}\mathrm{d}\gamma}{\pi r_s^2\zeta_0 n\int_0^{r_s} g(\gamma)\frac{\mathrm{d}P(r\le\gamma)}{\mathrm{d}\gamma}\mathrm{d}\gamma}}{1-\frac{\pi(kr_s)^2\zeta_0 n}{\pi r_s^2\zeta_0 n}}$$
$$= \frac{1-\frac{(ckr_s+1)^{2-\alpha}(1-\alpha)-(ckr_s+1)^{1-\alpha}(2-\alpha)+1}{(cr_s+1)^{2-\alpha}(1-\alpha)-(cr_s+1)^{1-\alpha}(2-\alpha)+1}}{1-k^2}$$
$$= \begin{cases}\Theta\left(\frac{1-\frac{(ckr_s)^2}{(cr_s)^2}}{1-k^2}\right) & \alpha = O(1)\\ \frac{1-\frac{1-(1+\alpha ckr_s)e^{-\alpha ckr_s}+o(1)}{1-(1+\alpha cr_s)e^{-\alpha cr_s}+o(1)}}{1-k^2} & otherwise\end{cases} = \Theta(1)$$

APPENDIX B
PROOF OF LEMMA 1

Let d_k be the distance between a sender N_s and one of its k-hop relays N_{r_k}, r_k be the distance between N_{r_k} and its *immediate sender* $N_{r_{k-1}}$, d_{k-1} be the distance between N_s and $N_{r_{k-1}}$, φ be the angle between $N_sN_{r_{k-1}}$ and $N_{r_{k-1}}N_{r_k}$.

As for r_{min}, similar to the proof of Theorem 1, we have

$$\begin{aligned}\mathbb{E}\left[r_{min}\right] &= \int_0^{r_s} r_0\frac{\mathrm{d}P(r_{min}<r_0)}{\mathrm{d}r_0}\mathrm{d}r_0\\ &= \int_0^{r_s} 2\pi\zeta_0 n g(r_0) r_0^2 e^{-2\pi\zeta_0 n\int_0^{r_0} g(\gamma)\gamma\mathrm{d}\gamma}\mathrm{d}r_0\\ &= \Theta\Big(\int_{\frac{1}{\sqrt{n}}}^{\frac{2}{\sqrt{n}}} nr_0^2 e^{-nr_0^2}\mathrm{d}r_0\Big) = \Theta(\frac{1}{\sqrt{n}}) = o(r_s)\end{aligned} \tag{7}$$

Let $N'_{r_{k-1}}$ be another $k-1$-hop relay closest to $N_{r_{k-1}}$, and d_m be the distance between them. We can get that

$$\mathbb{E}[d_m] = \int_0^{2r_s}\gamma\frac{\mathrm{d}P(d_m<\gamma)}{\mathrm{d}\gamma}\mathrm{d}\gamma = \Theta(\int_0^{r_s}\sqrt{n}\gamma e^{-\sqrt{n}\gamma}\mathrm{d}\gamma) = o(r_s).$$

Hereinafter, we prove $d_k \to kr_s$ by mathematical induction.

When $k=1$, $r_s - r_{min} < d_k \le r_s$ and thus $d_k \to kr_s$.

Suppose $d_{k_0} \to k_0 r_s$ when $k_0 \le k-1$. Since $d_m = o(r_s)$ and both r_k and the distance between N_{r_k} and $N'_{r_{k-1}}$ are larger than $r_s - r_{min}$, we have $\varphi = o(1)$. Thus $\cos\varphi \to 1-\frac{\varphi^2}{2}$ and

$$\begin{aligned}d_k &= \sqrt{(d_{k-1}+r_k\cos\varphi)^2+(r_k\sin\varphi)^2}\\ &\to (d_{k-1}+r_s)\sqrt{1-\frac{d_{k-1}r_s}{(d_{k-1}+r_s)^2}\varphi^2} \to d_{k-1}+r_s \to kr_s\end{aligned} \tag{8}$$

Above all, we can get that $d_k \to kr_s$.

APPENDIX C
PROOF OF THEOREM 3

Let D_{N_s} be the distance between a member N_s and the source S, d_1 (resp. d_2) be the distance between N_s and its closest front member (resp. closest back member), and C be the event that $\frac{d_1}{d_2} \ge b$. We can derive that

$$\begin{aligned}P(C) &\le \int_0^{D_{N_s}}\left(1-\frac{1}{3}\pi x^2\right)^{m-1}\frac{\mathrm{d}\big(1-(1-\frac{2}{3}\pi\frac{x^2}{b^2})^{m-1}\big)}{\mathrm{d}\frac{x}{b}}\mathrm{d}x\\ &\le \frac{4\pi(m-1)}{3b}\sum_{k=1}^{\infty}\int_{\sqrt{\frac{k-1}{m}}}^{\sqrt{\frac{k}{m}}} x\big(1-\frac{k-1}{3m}\pi\big)^{m-1}\mathrm{d}x\\ &\to \frac{2\pi(m-1)}{3b}\sum_{k=1}^{\infty}\frac{1}{m}e^{-\frac{k-1}{3}\pi} \le \frac{2\pi}{3b}\big(1+\int_0^{\infty}e^{-\frac{\pi}{3}y}\mathrm{d}y\big) = \Theta(\frac{1}{b}).\end{aligned}$$

Similarly, we have $P(C) \ge \Theta(\frac{1}{b})$, and thus $P(C) = \Theta(\frac{1}{b})$.

Let T' be a tree only containing the source and all members, and they only take connecting forwards strategy. As for the distance between any two connected nodes in T', it satisfies

$$\mathbb{E}[d_t] \le \int_0^{D_{N_s}}\gamma\big(1-\frac{1}{3}\pi\gamma^2\big)^{m-1}\frac{\mathrm{d}\big(1-(1-\frac{1}{2}\pi\gamma^2)^{m-1}\big)}{\mathrm{d}\gamma}\mathrm{d}\gamma = \Theta(\frac{1}{\sqrt{m}}).$$

We can also get $\mathbb{E}[d_t] \ge \Theta(\frac{1}{\sqrt{m}})$, and thus $\mathbb{E}[d_t] = \Theta(\frac{1}{\sqrt{m}})$.

We analyze the path length PL_{N_s} between N_s and S in T' as follows. Let N_c be a node in the path between N_s and S, N_t be another one receiving messages tranferred by N_c, and d be the distance between N_c and N_t. For simplicity, we use D to represent D_{N_c} and denote θ_0 as $\arccos\frac{d}{2D}$. Then we have

$$\mathbb{E}\left[D_{N_t}|d=r\right] = \int_{-\theta_0}^{\theta_0}\frac{\sqrt{(D-r\cos\theta)^2+(r\sin\theta)^2}}{2\theta_0}\mathrm{d}\theta < D-\frac{\sqrt{3}-1}{4}r$$

and $\mathbb{E}\left[D_{N_t}|d=r\right] > D-r$. Therefore, $\mathbb{E}\left[D_{N_t}|d=r\right] = D-\Theta\left(r\right)$. Then we can derive that

$$\mathbb{E}\left[D_{N_t}\right] = \int_0^{D}\left(D-\Theta\left(r\right)\right)P(d=r)\mathrm{d}r = D-\Theta(\mathbb{E}[d_t]) = D-\Theta(\frac{1}{\sqrt{m}}).$$

Let $\mathcal{N}_s$ be the number of nodes in the path from N_s to S, then $\mathbb{E}[PL_{N_s}] = \mathbb{E}[\mathcal{N}_s]\mathbb{E}[d_t] = \frac{D_{N_s}\mathbb{E}[d_t]}{\mathbb{E}[D-D_{N_t}]} = \Theta(D_{N_s})$.

Therefore, the height of T' is $H_{T'} = \Theta\big(\frac{\max_{N_s} PL_{N_s}}{\mathbb{E}[d_t]}\big) = \Theta(\sqrt{m})$. Denote that a node in T' is in j^{th} level if there are j nodes in its path to S. The number of nodes in j^{th} level is

$$m_j = \Theta\Big(\pi\Big(\big(k_1(j-1)\mathbb{E}[d_t]+k_2\mathbb{E}[d_t]\big)^2-\big(k_1\left(j-1\right)\mathbb{E}[d_t]\big)^2\Big)m\Big) = \Theta(j)$$

where k_1, k_2 are constants. The multicast members not in the *main connected component* in T are those tending to connect backwards and all their child members in T'. Let m_c be the number of these members. When $b = \sqrt{m}\log m$, we can get $\mathbb{E}[m_c] = \sum_{j=0}^{H_{T'}}\big(P(C)m_j\big(1-P(C)\big)^j\frac{\sum_{i=j}^{H_{T'}} m_i}{m_j}\big) = \sum_{j=0}^{\Theta(\sqrt{m})}\Theta\big(\frac{\sqrt{m}(\sqrt{m}-j)}{b}\big(1-\Theta(\frac{1}{b})\big)^j\big) = o(m)$.

APPENDIX D
PROOF OF THEOREM 6

In Phase 1, the number of messages transmitted in SSRDA is $M_{d1} \le \pi\zeta_2 n r_t^2\int_0^{r_t} g\left(\gamma\right)\frac{\mathrm{d}P(r\le\gamma)}{\mathrm{d}\gamma}\mathrm{d}\gamma = O(\frac{\log n}{(\alpha+1)^2})$. In the search process, the number of forwarded messages is $M_{s1} \le \sum_{k=1}^{k_{max}} 2\pi\zeta_2 nkr_s r_{min} = \Theta(\frac{\sqrt{n}}{r_s})$, where $k_{max} = \Theta(\frac{1}{r_s})$. In the response step, since all members and relays send messages, $M_{r1} \le M_{s1}+m = \Theta(\frac{\sqrt{n}}{r_s}+m)$. Then the messages in Phase 1 are $Msg_1 = O(\frac{(\alpha+1)n}{\sqrt{\log n}}+m)$.

In Phase 2, the number of messages in SSRDA is $M_{d2} = mM_{d1} = O(\frac{m\log n}{(\alpha+1)^2})$. In the i^{th} search process of member N, the number of messages is $M_{s2}^N(r_s) = O(1)$ when $i=1$ since no messages should be transferred, otherwise, $M_{s2}^N(2^{i-1}r_s) = O(\frac{4^i\sqrt{\log n}}{\alpha+1})$. Since $\mathbb{E}\left[d_t\right] = \Theta(\frac{1}{\sqrt{m}})$ according to Appendix C, the maximum coverage range is $R_m = \Theta(\max\{r_s, \frac{1}{\sqrt{m}}\})$.

(1). When $m = O(\frac{(\alpha+1)^2 n}{\log n})$, the messages in the search process are $M_{s2} = \sum_{i=1}^{i_{max}} mM_{s2}^N(2^{i-1}R_0) = O(\frac{(\alpha+1)n}{\sqrt{\log n}})$, where $i_{max} = \Theta(\log_2\frac{R_m}{R_0})$. The total response and connect messages are $M_{r2} = O(mR_m^2 M_{r1}+m\frac{R_m}{r_s}) = O(\frac{(\alpha+1)n}{\sqrt{\log n}})$.

(2). When $m = \omega(\frac{(\alpha+1)^2 n}{\log n})$, the probability of at least one front member being searched directly by N is $P_f \ge 1-e^{-\zeta_1\pi g(r_s)mr_s^2} \to 1$, and thus $R_m = r_s$, $M_{s2} = O(m)$ and $M_{r2} = O(m^2 r_s^2+m) = O(\frac{m^2\log n}{(\alpha+1)^2 n})$.

To sum up, the number of messages in Phase 2 is $Msg_2 = O(\frac{(\alpha+1)n}{\sqrt{\log n}}+\frac{m\log n}{(\alpha+1)^2})$.

In Phase 3, messages in identification step are no more than that in Phase 1, and messages in other steps are no more than that in Phase 2. Thus we have $Msg_3 \le Msg_1+Msg_2$.

Above all, the total messages transmitted in ULTCA is $Msg = Msg_1+Msg_2+Msg_3 = O(\frac{(\alpha+1)n}{\sqrt{\log n}}+\frac{m\log n}{(\alpha+1)^2})$.

References

[1] L. Selavo, A. Wood, Q. Cao, T. Sookoor, H. Liu, A. Srinivasan, Y. Wu, W. Kang, J. Stankovic, D. Young *et al.*, "Luster: wireless sensor network for environmental research," in *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 2007, pp. 103–116.

[2] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 13–24.

[3] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic *et al.*, "Vigilnet: An integrated sensor network system for energy-efficient surveillance," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 1, pp. 1–38, 2006.

[4] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[5] L. Galluccio, G. Morabito, and S. Palazzo, "Geographic multicast (gem) for dense wireless networks: protocol design and performance analysis," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 4, pp. 1332–1346, 2013.

[6] H. Gong, L. Fu, X. Fu, L. Zhao, K. Wang, and X. Wang, "Distributed multicast tree construction in wireless sensor networks," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 280–296, 2017.

[7] S. Zhao, L. Fu, X. Wang, and Q. Zhang, "Fundamental relationship between nodedensity and delay in wireless ad hoc networks with unreliable links," in *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 2011, pp. 337–348.

[8] G. G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks. isi research report." 1987.

[9] J. A. Sanchez, P. M. Ruiz, and I. Stojmnenovic, "Gmr: Geographic multicast routing for wireless sensor networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON'06. 2006 3rd Annual IEEE Communications Society on*, vol. 1. IEEE, 2006, pp. 20–29.

[10] J. A. Sanchez and P. M. Ruiz, "Lema: Localized energy-efficient multicast algorithm based on geographic routing," in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. IEEE, 2006, pp. 3–12.

[11] B. Musznicki, M. Tomczak, and P. Zwierzykowski, "Dijkstra-based localized multicast routing in wireless sensor networks," in *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*. IEEE, 2012, pp. 1–6.

[12] H. Park, J. Lee, S. Oh, Y. Yim, and S.-H. Kim, "Distributed multicast protocol based on beaconless routing for wireless sensor networks," in *Consumer Electronics (ICCE), 2013 IEEE International Conference on*. IEEE, 2013, pp. 522–523.

[13] D. Koutsonikolas, S. M. Das, Y. C. Hu, and I. Stojmenovic, "Hierarchical geographic multicast routing for wireless sensor networks," *Wireless networks*, vol. 16, no. 2, pp. 449–466, 2010.

[14] F. Bauer and A. Varma, "Distributed algorithms for multicast path setup in data networks," *IEEE/ACM Transactions on networking*, vol. 4, no. 2, pp. 181–191, 1996.

[15] L. M. Drummond, M. Santos, and E. Uchoa, "A distributed dual ascent algorithm for steiner problems in multicast routing," *Networks: An International Journal*, vol. 53, no. 2, pp. 170–183, 2009.

[16] S. Lin, F. Miao, J. Zhang, G. Zhou, L. Gu, T. He, J. A. Stankovic, S. Son, and G. J. Pappas, "Atpc: adaptive transmission power control for wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 12, no. 1, p. 6, 2016.

[17] M. D. Penrose *et al.*, "A strong law for the longest edge of the minimal spanning tree," *The Annals of Probability*, vol. 27, no. 1, pp. 246–260, 1999.

[18] J. M. Steele, "Growth rates of euclidean minimal spanning trees with power weighted edges," *The Annals of Probability*, pp. 1767–1787, 1988.

[19] E. Gilbert, "Random minimal trees," *Journal of the Society for Industrial and Applied Mathematics*, vol. 13, no. 2, pp. 376–387, 1965.

[20] D.-Z. Du and F. K. Hwang, "A proof of the gilbert-pollak conjecture on the steiner ratio," *Algorithmica*, vol. 7, no. 1-6, pp. 121–135, 1992.

[21] T. Pajor, E. Uchoa, and R. F. Werneck, "A robust and scalable algorithm for the steiner problem in graphs," *Mathematical Programming Computation*, vol. 10, no. 1, pp. 69–118, 2018.