Achieving Energy Efficiency and Reliability for Data Dissemination in Duty-Cycled WSNs

Kai Han, Jun Luo, Liu Xiang, Mingjun Xiao and Liusheng Huang

Abstract—Because data dissemination is crucial to Wireless Sensor Networks (WSNs), its energy-efficiency and reliability are of paramount importance. While achieving these two goals together is highly non-trivial, the situation is exacerbated if WSN nodes are duty-cycled (DC) and their transmission power is adjustable. In this paper, we study the problem of minimizing the expected total transmission power for reliable data dissemination (multicast/broadcast) in DC-WSNs. Due to the NP-hardness of the problem, we design efficient approximation algorithms with provable performance bounds for it. To facilitate our algorithm design, we propose the novel concept of Time-Reliability-Power (TRP) space as a general data structure for designing data dissemination algorithms in WSNs, and the performance ratios of our algorithms based on the TRP space are proven to be $O(\log \Delta \log k)$ for both multicast and broadcast, where $\Delta$ is the maximum degree in the network and $k$ is the number of source/destination nodes involved in a data dissemination session. We also conduct extensive simulations to firmly demonstrate the efficiency of our algorithms.

I. INTRODUCTION

Designing effective data dissemination mechanisms for Wireless Sensor Networks (WSNs) is of paramount importance, as WSNs rely on data dissemination to carry critical commands or code updates from a sink to a set of (or all) nodes in the networks [1], [2]. Reliability and energy-efficiency are perhaps the most crucial requirements for data dissemination due to the notorious packet-losses in wireless communications and the limited power supply of sensor nodes.

Achieving reliability and energy-efficiency simultaneously is by no means trivial, and the problem gets even more challenging when modern WSNs' features such as duty-cycling and power-adjustability are taken into account. On one hand, in a Duty-Cycled WSN (DC-WSN), nodes usually switch between active/dormant states periodically, which imposes extra difficulty on attaining energy-efficiency even without regarding reliability [3]–[5]. On the other hand, nodes' capability of adjusting the transmission power (tx-power) to several levels brings out a tradeoff between reliability and energy-efficiency, as a node can increase its tx-power to improve link reliability at the cost of higher energy consumption [6]. All these entangled features make it extremely challenging to design energy-efficient reliable data dissemination mechanisms for DC-WSNs.

Despite major efforts on energy-efficient data dissemination in wireless networks, a large body of them have taken the overly-optimistic assumption of error-free wireless transmissions [3], [4], [7]–[10]. Other proposals considering unreliable links roughly follow two lines. One line bears the common objective of minimizing the expected tx-power consumption under guaranteed reliability, but they are designed only for Always-Active Wireless Networks (AAWNs) [6], [11], [12]. This batch of work has unanimously concentrated on unicasting or employing unicasting to achieve multicasting [6] without leveraging the Wireless Broadcast Advantage (WBA). Another line aims at reducing the tx-power consumption for broadcasting in DC-WSNs without a stringent requirement on reliability. As a representative proposal in this line, [13] proposed a tree-based opportunistic flooding approach. However, the broadcast tree used in [13] is constructed for very low duty-cycle, hence transmissions are still done through unicasting (similar to [6]).

In this paper, we present the first study on energy-efficient reliable data dissemination in DC-WSNs with guaranteed performance. We aim to minimize the expected total tx-power consumption in multicasting or broadcasting under guaranteed reliability, and we take into account WBA, unreliable links, power-adjustability and duty-cycling holistically. Given that the resulting Energy Efficient Reliable Data Dissemination (ERDD) problem is NP-hard, we propose approximation algorithms for it with performance ratios of $O(\ln \Delta \ln k)$, where $\Delta$ is the maximum node degree in the network and $k$ is the total number of nodes involved in a multicast/broadcast session. We also show the efficiency of our approach by extensive simulation results.

The rest of our paper is organized as follows. Section II reviews related work. Section III provides our models and problem definitions. In Section IV, we present our approximation algorithms. The performance ratios of our algorithms are analyzed in Section V. Section VI reports our simulation results and Section VII concludes the paper. To maintain fluency, all the proof sketches of our lemmas/theorems are postponed to the Appendix.

II. RELATED WORK

Always-Active Wireless Networks with Reliable Links: Min-energy multicast/broadcast and related topology control prob-
problems in AAWNs with reliable links have been studied extensively in the literature. As these problems are generally NP-hard [14], many approximation algorithms have been proposed under different network models and constraints. The seminal work in [15] provides upper and lower bounds for the power assignment problem under different connectivity constraints, and proposes asymptotically optimal $O(\log n)$ approximation algorithms for min-power (strong or symmetric) connectivity and min-power broadcasting problems. The work in [16] proposes an approximation algorithm for the min-power broadcast routing problem in wireless ad hoc networks that achieves an exponentially better approximation ratio compared to the Minimum Spanning Tree (MST) heuristic. Recently, the work in [17] provides both fast exact algorithms and approximation algorithms for the min-power strong connectivity problem. Other approximation algorithms in this area can be found in [7]–[10], [18]–[23], and a comprehensive survey is presented in [24]. Our work differs from all these proposals in that we are considering the min-energy data dissemination problem in WSNs with duty-cycling and unreliable links; this problem is non-trivial even when all nodes’ tx-power is predetermined and can’t be adjusted.

**DC-WSNs with Reliable Links:** Recently, the min-energy multicast problem in DC-WSNs has been tackled in [3], [4], where the nodes’ tx-power is assumed to be fixed and identical. The work in [4] has presented a dynamic programming approach to solve the problem optimally. However, the optimal algorithms proposed in [4] have exponential time complexities, hence are only used when the number of destination nodes is small (e.g., a few sinks). In contrast to [4], a later proposal [3] has provided polynomial-time approximation algorithms with provable performance ratios. However, both [4] and [3] assume that the wireless links are perfectly reliable, so the energy consumption due to retransmissions in realistic settings can be much higher than what claimed in these proposals.

**Always-Active Wireless Networks with Unreliable Links:** Assuming AAWNs, [6], [11], [12] have taken unreliable links into account when designing energy efficient data dissemination algorithms. Whereas they bare the same objective as ours, i.e., minimizing the expected total tx-power consumption for guaranteed reliability, all of them have stressed on unicasting cases and have extended traditional shortest-path algorithms to solve their problem optimally. As an exception, [6] did consider a multicast scenario (under the assumption of fixed tx-power). Although a multicast tree was built in [6], its solution is intrinsically unicasting, i.e., any non-leaf node in the multicast tree must transmit data individually to each of its children nodes until success. Neglecting the WBA feature of wireless communication, this method is too pessimistic and hence harms energy-efficiency. To understand this, we use a toy example proposed in [25]. Suppose that the nodes in Figure 1 all have a unit tx-power and $A$ is the source node, then [6] creates the tree in Figure 1(b) for broadcast, whose expected tx-power (based on unicasting) is 19. However, when WBA is considered, the expected tx-power of a better solution, shown in Figure 1(c), is at most 11.9 (computed by Lemma 1 in Section III).

**DC-WSNs with Unreliable Links:** While ADB [26] proposes MAC-layer protocols for efficient broadcasting in asynchronous DC-WSNs, [13] has adopted an opportunistic flooding approach for the same purpose, where a BFS tree is constructed as a broadcast backbone. Although opportunistic links outside the broadcast tree are exploited by [13], transmissions along the tree are still based on unicasting (the same as [6] shown by Figure 1), as WBA can be “crippled” under very low duty-cycle. Therefore, the broadcast tree used in [13] might not be energy-efficient given that both WBA and common waking-up time can be explored under a generic duty-cycling pattern. We note that neither [26] nor [13] aims at guaranteed reliability as opposed to [6], [11], [12], and it would be non-trivial for them to adapt to the multicast case. In addition, the power-adjustability factor has not been considered by these proposals.

**Summary:** To the best of our knowledge, we are the first to design algorithms with provable performance bounds for energy-efficient reliable data-dissemination in DC-WSNs, where the impact of WBA, unreliable links, duty-cycling and power-adjustability are all taken into account. Moreover, as AAWNs can be considered as special cases of DC-WSNs, our approach also upgrades the current technique for AAWNs that uses unicasting to fulfill energy-efficient reliable multicasting [6].

**III. MODEL AND PROBLEM DEFINITION**

We assume a set $V$ of DC-WSN nodes; they switch between active/sleeping states periodically. The set of active time slots in a working period of any node $u \in V$ is denoted by $A(u) \subseteq L = \{1, 2, ..., l\}$, where $l$ is the length of the working period. Following [3], [4], [13], we assume that all nodes in $V$ are time synchronized, and a node can wake up its transceiver to transmit data at any time slot, but can only receive data when it is active. We also assume that each node $u \in V$ can adjust its tx-power to several levels, which are $\epsilon_{\text{min}} = \epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_d = \epsilon_{\text{max}}$. As opposed to some other work on topology control (e.g., [15]), we assume that both the number of power levels (i.e., $d$) and the power values (i.e., $\epsilon_1, \epsilon_2, \cdots, \epsilon_d$) are predefined constants, because commercial sensor nodes can only adjust their power to a few predefined levels. Let $L = \{\epsilon_1, \epsilon_2, ..., \epsilon_d\}$. When $u$’s tx-power is adjusted to $l \in L$, let the link quality $p_{uv}(l) \in (0, 1]$ denote the success ratio of data transmission on link $(u, v)$. Following [6], we assume that $p_{uv}(l)$ increases with $l \in L$, and the link qualities considered in the network are independent and have a positive
lower bound (denoted by $\Lambda$). Let $N_u(l)$ denote the node set $\{v|v \in \mathcal{V}\{u\} \land p_{uv}(l) \geq \Lambda\}$. We also assume that the link quality $p_{uv}(l)$ is mainly affected by the euclidian distance between $u$ and $v$ given a fixed transmission power; this implies that $p_{uv}(l) = p_{vu}(l)$, as well as $v \in N_u(l) \Rightarrow u \in N_v(l)$.

In a data dissemination session, there exists a source node $s \in \mathcal{V}$ that needs to send data to a set of destination nodes $\mathcal{R} \subseteq \mathcal{V} - \{s\}$. Let $k = |\mathcal{R}| + 1$, hence the session is a broadcast if $k = |\mathcal{V}|$; otherwise a multicast. A valid power assignment for such a session is a function $L: \mathcal{V} \rightarrow \mathcal{L}$, such that when each node $u \in \mathcal{V}$ is adjusted to the tx-power level $L(u)$, there exists a data dissemination tree $T$ spanning the nodes in $\mathcal{R} \cup \{s\}$; and the set of parent/children nodes of any node $v \in T$ must be contained in $N_v(L(v))$ for $v$ to send data and to receive control messages.

To conduct a data dissemination session in a DC-WSN, we need not only to find a valid power assignment $L$ and a data dissemination tree $T$, but also to select the transmission time slots of the forwarding nodes in $T$, in order to avoid transmitting data to sleeping nodes. Based on this observation, we provide the definition of Viable Data-dissemination Solution (VDS) in Definition 1, where we define the set of non-leaf nodes in $T$ by $\lambda(T)$, the set of children nodes of any node $u \in T$ by $\mathcal{C}_T(u)$, and the set $\{v|v \in N_u(l) \land t \in \mathcal{A}(v)\}$ by $\partial_l(u, t)$:

**Definition 1 (VDS):** Suppose that $T$ is a data dissemination tree under a valid power assignment $L$, and $S$ is a function that satisfies:

1. For any $t \in \mathcal{I}$ and any $u \in \lambda(T)$, $S(u, t) \subseteq \partial_l(u, L(u)) \cap \mathcal{C}_T(u)$,
2. For any $u \in \lambda(T)$, $\mathcal{C}_T(u) \subseteq \bigcup_{t \in \mathcal{I}} S(u, t)$,

then $S$ is called a Viable Transmission Schedule for $T$ and $(L, T, S)$ is called a Viable Data-dissemination Solution (VDS).

Basically, VDS requires a node $u \in T$ to be responsible for sending data to nodes in the set $S(u, t)$ at time slot $t$. Due to the link unreliability, $u$ may have to retransmit several times (in different working periods) for all the nodes in $S(u, t)$ to receive the data (if $S(u, t) \neq \emptyset$). To understand how many retransmissions are needed for a forward node, we introduce Lemma 1:

**Lemma 1:** For any node $u \in \mathcal{V}$, any power level $l \in \mathcal{L}$ and any set $Q \subseteq N_u(l)$, let $\chi_u(l, Q)$ be the random variable that denotes the number of transmissions by $u$ for all the nodes in $Q$ to receive a data packet when $u$’s tx-power is $l$. If $Q \neq \emptyset$, then we have:

$$\mathbb{E}[\chi_u(l, Q)] = \sum_{i=0}^{\infty} \left[ 1 - \prod_{v \in Q} (1 - (1 - p_{uv}(l))^i) \right]$$

Our objective is to find a VDS such that the expected total transmission energy of the forward nodes is minimized. We introduce the formal definition of this problem by Definition 2 and Definition 3:

**Definition 2 (Energy-Consumption Function):** The energy-consumption function of a VDS $(L, T, S)$ is

$$\Psi(L, T, S) = \sum_{u \in \lambda(T)} \sum_{t \in \mathcal{I}} L(u) \cdot \chi_u(l, S(u, t))$$

**Definition 3 (ERDD Problem):** Given a set $\mathcal{V}$ of DC-WSN nodes, a source node $s \in \mathcal{V}$, and a set of receiver nodes $\mathcal{R} \subseteq \mathcal{V} - \{s\}$, the Energy-efficient Reliable Data Dissemination (ERDD) problem is to find a VDS $(L^*, T^*, S^*)$ such that $\mathbb{E}[\Psi(L^*, T^*, S^*)]$ is minimized.

As a special case of the ERDD problem, if all the network links are reliable and all nodes are active at all time slots, then the ERDD problem degenerates to the min-energy broadcast/multicast problem in traditional AAWNs with perfect links, which is known to be NP-hard [9], [10]. Since the ERDD problem contains a special case which is NP-hard, we have:

**Theorem 1:** The ERDD problem is NP-hard.

In the following section, we will propose approximation algorithms for the ERDD problem. To facilitate reading, we summarize the notations in Table I.
IV. APPROXIMATION ALGORITHMS

A basic idea of our approximation algorithm design is to build a novel data structure called the Time-Reliability-Power (TRP) Space, where data dimensions on time, reliability and power levels are involved to facilitate our algorithm design and analysis. We believe that the TRP space can be used as a general data structure to facilitate the algorithm design for many optimization problems related to data routing/dissemination in WSNs. In the sequel, we will first introduce the concepts about the TRP space, and then present our algorithms in details.

A. Time-Reliability-Power Space

To build a TRP space, we first define a positive number $\gamma_{uv}(l)$ for any $u \in \mathcal{V}, l \in \mathcal{L}$ and $v \in \mathcal{N}_u(l)$ as follows:

$$
\gamma_{uv}(l) = \begin{cases} 
1 - \frac{1}{\min(1-p_{uv}(l))}, & p_{uv}(l) \in (0, 1) \\
1, & p_{uv}(l) = 1
\end{cases}
$$

Note that $\gamma_{uv}(l)$ is not less than 1 and decreases when $p_{uv}(l)$ increases. Hence, a larger $\gamma_{uv}(l)$ indicates a poorer link. Besides, since link qualities have a constant lower bound in practice [6], $\gamma_{uv}(l)$ has a constant upper bound. Based on this definition, we introduce the concept of TRP space and a weight assignment method in Definition 4 and Definition 5, respectively.

**Definition 4 (TRP Space):** For any $u \in \mathcal{V}$, define the TRP set of $u$ as $\vartheta(u) = \{(u, t, r, l) | (t, r, l) \in \mathcal{I} \land l \in \mathcal{L} \land r \in \bigcup_{v \in \mathcal{N}_u(l)} \gamma_{uv}(l)\}$. Define $\mathcal{U} = \bigcup_{u \in \mathcal{V}} \vartheta(u) \cup \mathcal{V}$. An access-relationship $\mathcal{W}$ is any set of ordered 2-tuplets wherein each 2-tuplet consists of two elements from $\mathcal{U}$. The tuple $(\mathcal{U}, \mathcal{W})$ is called a TRP space.

**Definition 5 (Weight Assignment):** Any $x \in \mathcal{U}$ has a weight $\vartheta(x)$. If $x \in \mathcal{V}$ then $\vartheta(x) = 0$. If $x = (u, t, r, l) \in \mathcal{U}|(\mathcal{V)}$, then $\vartheta(x) = r \cdot l$. The weight of any $\mathcal{V} \subseteq \mathcal{U}$ is also denoted by $\vartheta(\mathcal{V}) = \sum_{x \in \vartheta(x)} \vartheta(x)$. Intuitively, the elements in $\mathcal{U}$ can be viewed as weighted 4-dimensional (vector) nodes whose adjacent relationships are determined by an access-relationship $\mathcal{W}$, while $\mathcal{W}$ is defined by a given problem, as we shall do in Section IV-B and IV-C. Based on above definitions, we introduce the concept of “TRP path” in Definition 6.

**Definition 6 (TRP Path):** Given a TRP space $(\mathcal{U}, \mathcal{W})$, a sequence $h = (x_1, x_2, ..., x_m)$ is called a TRP path from $x_1$ to $x_m$ iff $(x_i, x_{i+1}) \in \mathcal{W}, \forall l \leq t \leq m - 1$. The length of $h$ is defined as

$$
\ell(h) = \begin{cases} 
\sum_{i=2}^{m-1} \vartheta(x_i) & m > 2 \\
0 & \text{otherwise}
\end{cases}
$$

Based on the length function $\ell$, the shortest TRP path from $x \in \mathcal{U}$ to $y \in \mathcal{U}$ is the one with the minimum length, which is denoted by $x \rightarrow y$.

In our algorithm design, we will map a connected component in the TRP space $(\mathcal{U}, \mathcal{W})$ to a data-dissemination solution. Hence, we introduce some definitions for this mapping. The node-image of any $x \in \mathcal{U}$ is defined as the node $u \in \mathcal{V}$ such that $x \in \vartheta(u) \cup \{u\}$. The edge-image of any tuple $(x, y) \in \mathcal{W}$ is defined as the directed edge $(u, v) : u \in \mathcal{V}, v \in \mathcal{V}$ such that $x \in \vartheta(u) \cup \{u\}$ and $y \in \vartheta(v) \cup \{v\}$. Suppose that $(x_1, \gamma_1)$ is a subpace of $(\mathcal{U}, \mathcal{W})$ (i.e., $x_1 \subseteq \mathcal{U}$, $\gamma_1 \subseteq \mathcal{W}$), we define a mapping function $\mathfrak{f}$ such that $\mathfrak{f}(x_1, \gamma_1)$ is a directed graph constructed by all node-images of the elements in $x_1$ and all edge-images of the tuples in $\gamma_1$, i.e., all nodes in $\mathfrak{f}(x_1, \gamma_1)$ are in $\mathcal{V}$.

For the convenience of description, we introduce some other notations/definelines about the TRP space here. Given a TRP space $(\mathcal{U}, \mathcal{W})$ and any $x, y \in \mathcal{U}$, we say $x$ is accessible to $y$ (or $y$ is accessible from $x$) iff $(x, y) \in \mathcal{W}$. Given any $x \in \mathcal{U}$, we define the element closure and relationship closure of $x$ with respect to $\mathcal{W}$ as $\mathfrak{g}_1(x, \mathcal{W}) = \mathcal{U} \cup \{y | x \in \mathcal{U} \land (x, y) \in \mathcal{W}\}$ and $\mathfrak{g}_2(x, \mathcal{W}) = \{(x, y) | x \in \mathcal{U} \land (x, y) \in \mathcal{W}\}$, respectively. For any TRP path $h = (x_1, ..., x_m)$ in $(\mathcal{U}, \mathcal{W})$, we define $\min(h) = \{x_i | 1 \leq i \leq m - 1\}$. If $\mathfrak{g}_1$ contains all $x_i : 1 \leq i \leq m$, we say $\mathfrak{g}_1$ embraces $h$. For any $u \in \mathcal{V}$, define $\gamma_{\max} = \max\{\gamma_{uv}(l) | l \in \mathcal{L} \land v \in \mathcal{N}_u(l)\}$ and $\gamma_{\min} = \min\{\gamma_{uv}(l) | l \in \mathcal{L} \land v \in \mathcal{N}_u(l)\}$. Define $\lambda = \max\{\gamma_{\max} | u \in \mathcal{V}\}$; it has a constant upper bound according to the definition of $\gamma_{uv}(l)$.

B. Solving ERDD for the Multicast Case

To design an approximation algorithm for the ERDD problem under multicast, we construct a TRP space $(\mathcal{U}, \mathcal{W}_M)$ where $\mathcal{W}_M$ is defined by the following rules:

**M1:** Two elements $(u_1, t_1, r_1, l_1)$ and $(u_2, t_2, r_2, l_2)$ in $\mathcal{U} \setminus \mathcal{V}$ are accessible to each other iff either $u_1 = u_2$ or $[u_1 \in \partial_1(u_2, l_2)] \land [u_2 \in \partial_1(u_1, l_1)] \land [r_1 \geq \gamma_{u_1, u_2}(l_1)] \land [r_2 \geq \gamma_{u_2, u_1}(l_2)]$ is true.

**M2:** Two elements $x = (u_1, t_1, r_1, l_1)$ in $\mathcal{U} \setminus \mathcal{V}$ and $u_2 \in \mathcal{V}$ are accessible to each other iff either $u_1 = u_2$ or there exists $y \in \partial(u_2)$ such that $(x, y) \in \mathcal{W}_M$.

A simplified illustration of the above rules is shown in Fig. 2. Note that $\mathcal{W}_M$ is symmetric, i.e., $(x, y) \in \mathcal{W}_M \Rightarrow (y, x) \in \mathcal{W}_M$. Intuitively, M1 embodies the covering relationship between two nodes under the time, reliability and power constraints, and M2 is set up to avoid counting the weights of a multicast tree’s leaf nodes in our algorithm. Based on this TRP space, we introduce an approximate algorithm MC-ERDD (Algorithm 1).

In Algorithm 1, we first consider $(\mathcal{U}, \mathcal{W}_M)$ as an undirected graph with node set $\mathcal{U}$ and edge set $\mathcal{W}_M$ and find an approximate Node Weighted Steiner Tree (NWST) with node set $\mathcal{U}$ (line 1), then we map the NWST to an approximate solution $(L^m, T^m, S^m)$ for ERDD (lines 2-18). Roughly speaking, the idea for doing this is that we can find the tx-power and
Algorithm 1: MC-ERDD($\mathcal{V}, s, R, I, U, \mathcal{W}_M, \mathcal{L}, A$)

1 Use an approximate node weighted Steiner tree algorithm [27] to find a subspace $\langle u', \mathcal{W}' \rangle$ of $\langle U, \mathcal{W}_M \rangle$ which contains $R \cup \{s\}, \mathcal{I} \leftarrow u'$
2 Let $T^m$ be an arbitrary spanning tree of $\mathcal{S}(u', \mathcal{W}')$
3 foreach $u \in \lambda(T^m)$ do
4   foreach $v \in C_{T^m}(u)$ do
5     if $\{(u', v) \in \mathcal{W}_M | u' \in \mathcal{U}' \cap \partial(u) \} = \emptyset$ then
6         Find $\langle v, \hat{r}, \hat{l}, \bar{l} \rangle \in \mathcal{I}\setminus \mathcal{V}$ such that $u \in \partial(v, \hat{l})$
7         Select an arbitrary time slot $\hat{t} \in A(v)$ and add $\langle u, \hat{t}, \hat{r}, \hat{l} \rangle$ into $\mathcal{U}$
8         $B \leftarrow C_{T^m}(u)$
9     foreach $t \in I$ do
10        $S^m(u, t) \leftarrow \emptyset$
11        foreach $l \in C_{T^m}(u)$ do
12           $\hat{r} \leftarrow \max\{r' \mid \langle u, t, r', l \rangle \in \mathcal{I}\setminus \mathcal{V} \land r' = 0\}$
13           $S^m(u, t) \leftarrow S^m(u, t) \cup \{v' \mid r' \in \mathcal{I}\setminus \mathcal{V} \land r' = 0\}$
14        $B \leftarrow B - S^m(u, t)$
15     foreach node $u \in T^m$ do
16        $J \leftarrow$ the set of $u$'s neighboring nodes in $T^m$
17        $\bar{l} \leftarrow \min\{l' \mid J \subseteq N_{L}(l') \land l' \in C_{T^m}(u)\}$
18        $L^m(u) \leftarrow \max\{l' \mid \langle u, t', r', l' \rangle \in \mathcal{I}\setminus \mathcal{V} \land l' = \bar{l}\}$
19 return $\langle L^m, T^m, S^m \rangle$

transmission schedule of any $u \in \lambda(T^m)$ based on the NWST nodes in $\partial(u)$ (lines 8-18). To make this idea work correctly, we need to add some elements to $\mathcal{U}'$ (hence expand $\mathcal{U}'$ to $\mathcal{U}$) (lines 4-7), because there may exist $u \in \lambda(T^m)$ such that $\mathcal{U}' \cap \partial(u) = \emptyset$ according to rule M2. The correctness and performance ratio of Algorithm 1 will be proved in Section V. The dominating running time of Algorithm 1 is spent in line 1, which is determined by the time complexity of the NWST algorithm [27]. Hence, we get:

**Theorem 2:** The time complexity of the MC-ERDD algorithm is $O(k^2|\mathcal{V}|^2\Delta^2)$.

### C. Solving ERDD for the Broadcast Case

Although Algorithm 1 can also be used for broadcast, we introduce another algorithm for the broadcast case in this section which has a better approximation ratio. Again, the first step is to construct a TRP space $\langle U, \mathcal{W}_B \rangle$, but the access-relationship $\mathcal{W}_B$ is defined differently from $\mathcal{W}_M$ according to the following rules:

**B1:** An element $\langle u_1, t_1, r_1, l_1 \rangle \in U \setminus \mathcal{V}$ is accessible to another element $\langle u_2, t_2, r_2, l_2 \rangle \in U \setminus \mathcal{V}$ iff the Boolean expression $\langle u_1 \neq u_2 \land u_2 \in \partial(u_1, t_1, l_1) \land r_1 \geq r_2 \rangle \cup \langle u_1 \in N_{\mathcal{L}_w}(l_2) \rangle$ is true.

**B2:** An element $x = \langle u_1, t_1, r_1, l_1 \rangle \in U \setminus \mathcal{V}$ is accessible to $u_2 \in \mathcal{V}$ iff $u_1 \neq u_2$ and there exists $y \in \partial(u_2)$ such that $(x, y) \in \mathcal{W}_B$.

**B3:** For any $u \in \mathcal{V}$ and any $y \in \partial(u)$, $(u, y) \in \mathcal{W}_B$.

A simplified illustration of B1-B3 is shown in Fig. 3. Note that $\mathcal{W}_B$ is not symmetric and hence we cannot run Algorithm 1 on $\langle U, \mathcal{W}_B \rangle$. Actually, B1-B3 are deliberately designed with nice properties desired by our new approximation algorithm BC-ERDD (Algorithm 2), whose idea originates from the NWST algorithm in [27]. As we shall prove, by leveraging rules B1-B3. Algorithm 2 yields a better performance ratio than Algorithm 1 in the broadcast case.

Algorithm 2: BC-ERDD($\mathcal{V}, s, R, I, U, \mathcal{W}_B, \mathcal{L}, A$)

1 $X \leftarrow \emptyset$, $Z \leftarrow \mathcal{V} \setminus \{s\}$
2 while $|Z| > 0$ do
3   Find $a \in U$ and a non-empty set $D \subseteq Z$ such that $|D| \geq 2 \lor a \in s$ is true and $avl(a, D)$ is minimized. Let $b$ be the node-image of $a$
4   $A \leftarrow \bigcup_{a \in D} in(a \sim y)$
5   $X \leftarrow A \setminus X$
6   $Z \leftarrow Z \setminus \{A, \mathcal{W}_B\}$
7   if $b \neq s$ then $Z \leftarrow Z \cup \{b\}$
8 Let $T^b$ be an arbitrary directed spanning tree of $\mathcal{S}(q_1(X, \mathcal{W}_B), q_2(X, \mathcal{W}_B))$ rooted at $s$
9 Set $L^b, S^b$ for the nodes in $T^b$ using the same method as lines 8-18 of Algorithm 1
10 return $\langle L^b, T^b, S^b \rangle$

In Algorithm 2, we use $X$ to denote the subset of $\mathcal{U}$ which will be mapped to the forward nodes in broadcasting, and use $Z$ to denote the set of nodes in $\mathcal{V}$ which are not accessible from any element in $X$. At first, $X$ is initialized to $\emptyset$ and $Z$ is initialized to $\mathcal{V} \setminus \{s\}$. Then the algorithm uses a greedy strategy to expand $X$ and reduce $Z$ until $Z = \emptyset$ (lines 2-7). In each iteration, we greedily select an element $a \in U$ and a subset $D$ of $U$ such that

$$avl(a, D) = \frac{\omega(a) + \sum_{y \in D} l(a \sim y)}{|D|}$$

is minimized (line 3), and then update $X$ and $Z$ accordingly (lines 4-7). Intuitively, $avl(a, D)$ denotes the average cost for “covering” the nodes in $D$, which is analogous to the “cost effectiveness” measure in the greedy set-cover algorithm [28]. When $X$ is finally determined, we use it to find an approximation solution $\langle L^b, T^b, S^b \rangle$ in lines 8-9 based on a similar
mapping process as that in Algorithm 1 (regarding $X$ as $\tilde{U}$).
The time complexity of Algorithm 2 is given by:

**Theorem 3:** The time complexity of the BC-ERDD algorithm is $O(|\mathcal{V}|^3 \Delta^3)$.

V. PERFORMANCE ANALYSIS
In this section, we prove the correctness and performance ratios of the algorithms proposed in Section IV. We shall first introduce an analysis method which is used for analyzing both the MC-ERDD and the BC-ERDD algorithms, and then give the detailed performance analysis of each algorithm.

A. A Method for Performance Analysis
From Lemma 1 and Definition 2 we can see that, for any VDS $\langle L, T, S \rangle$, the expectation value $E[\Psi(L, T, S)]$ is a summation of infinite series, which is hard to calculate. This makes it hard to find the performance ratios of our algorithms. To bypass this problem, we introduce a surrogate function $\phi$, which is defined as follows:

$$\phi(L, T, S) = \sum_{u \in \lambda(T)} \sum_{t \in T} L(u) \cdot \max\{\gamma_{uv}(L(u)) | v \in S(u, t)\}$$

(3)

It serves as an approximation to the expectation of $\Psi$. Let $\langle \hat{L}, \hat{T}, \hat{S} \rangle$ denote the output of Algorithm 1 or Algorithm 2. We find the quantitative relationships between $\phi$ and $\Psi$ by Lemma 2 and Lemma 3:

**Lemma 2:** We have $E[\Psi(\hat{L}, \hat{T}, \hat{S})] \leq (\ln \Delta + 1) \cdot \phi(\hat{L}, \hat{T}, \hat{S})$ and $\sigma(\Psi(\hat{L}, \hat{T}, \hat{S})) \leq \sqrt{2(\ln \Delta + 1)} \cdot \phi(\hat{L}, \hat{T}, \hat{S})$, where $\sigma(\cdot)$ denotes the standard deviation.

**Lemma 3:** $\phi(L^*, T^*, S^*) \leq (1 + 1/\sqrt{2}) \cdot \lambda E[\Psi(L^*, T^*, S^*)]$. Let $\langle \hat{L}^*, \hat{T}^*, \hat{S}^* \rangle$ be a VDS such that $\phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)$ is minimized. Intuitively, $\langle \hat{L}^*, \hat{T}^*, \hat{S}^* \rangle$ is a solution in which each forwarding node selects the more reliable links for data transmission. Combining the above lemmas with the fact that $\phi(\hat{L}^*, \hat{T}^*, \hat{S}^*) \leq \phi(L^*, T^*, S^*)$, we immediately get:

**Theorem 4:** If $\phi(L, T, S) \leq \beta \cdot \phi(L^*, T^*, S^*)$, then we have $E[\Psi(L, T, S)] \leq \alpha \beta \cdot E[\Psi(L^*, T^*, S^*)]$ and $\sigma(\Psi(L, T, S)) \leq \sqrt{2 \alpha \beta} \cdot E[\Psi(L^*, T^*, S^*)]$ where $\alpha = (1 + 1/\sqrt{2}) \lambda (\ln \Delta + 1)$.

**Theorem 4** actually suggests a method for analyzing our algorithms, i.e., to find the performance ratios of Algorithm 1 and Algorithm 2, we only need to find their approximation ratios with respect to the surrogate function $\phi$. In the following sections, we will analyze our algorithms based on this method.

B. Analyzing the MC-ERDD Algorithm
We first prove the correctness of Algorithm 1 in Lemma 4, and then prove in Lemma 5 that $\phi(L^*, T^*, S^*)$ is within constant times of $\varpi(\mathcal{U}')$, which is the weight of the NWST we found in Algorithm 1. The proofs of these lemmas are based on the construction rules of $\langle \mathcal{U}, \mathcal{W}_M \rangle$ as well as the mapping process employed in Algorithm 1 that maps an NWST to a VDS.

**Lemma 4:** $\langle L^*, T^*, S^* \rangle$ is a VDS for multicast.

**Lemma 5:** $\phi(L^*, T^*, S^*) \leq \frac{\mu_{\max}}{\epsilon_{\min}} \cdot \varpi(\mathcal{U}')$

Next, we reveal the quantitative relationship between $\varpi(\mathcal{U}')$ and $\phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)$ by Lemma 6. The main idea behind Lemma 6 is that, we can find a tree spanning $\mathcal{R} \cup \{s\}$ in $\langle \mathcal{U}, \mathcal{W}_B \rangle$ whose weight is at most $(1 + \lambda) \phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)$, whereas the NWST algorithm we used in Algorithm 1 has an approximation ratio of $2 \ln k$.

**Lemma 6:** $\varpi(\mathcal{U}') \leq 2(1 + \lambda) \ln k \cdot \phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)$

Recall that $\epsilon_{\max}, \epsilon_{\min}$ are both pre-defined constants and $\lambda$ has a constant upper bound. Hence, combining Lemma 5, Lemma 6, and Theorem 4 yields:

**Theorem 5:** The expectation value and standard deviation of $\Psi(L^m, T^m, S^m)$ are within $9.8\eta j$ and $13.9\eta j$ times of $E[\varpi(L^*, T^*, S^*)]$, respectively, where $\eta = \frac{\mu_{\max}}{\epsilon_{\min}} \cdot (1 + \lambda)(2 \ln k + 1) \ln k = O(\ln \Delta \ln k)$.

C. Analyzing the BC-ERDD Algorithm
Based on similar reasoning to the proof of Lemma 4, we can also prove the correctness of Algorithm 2 based on the definition of $\mathcal{W}_B$, as shown by Lemma 7:

**Lemma 7:** $\langle L^b, T^b, S^b \rangle$ is a VDS for broadcast.

Suppose that the while loop in lines 2-7 of Algorithm 2 executes for $q$ times. Let $Z_j = Z$ and $X_j = X$ after the $j$th while loop is executed. Let $D_j$ be the set found in the $j$th while loop. Clearly we have $\varpi(X_j) = 0$ and $|Z_j| = 0$. Indeed, the value $(\varpi(X_{j+1}) - \varpi(X_j))/|D_{j+1}|$ represents the “cost effectiveness” of the elements added to $X$ in the $(j + 1)$th loop, which is bounded by Lemma 8:

**Lemma 8:** $(\varpi(X_{j+1}) - \varpi(X_j)) \cdot |Z_j| \leq |D_{j+1}| \cdot \phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)(0 \leq j < q - 1)$

Note that the elements in $X_q$ are finally mapped to the transmission schedules of the non-leaf nodes in $T^b$. Hence, based on Lemma 8 and the mapping process in Algorithm 2, we can get:

**Lemma 9:** $\phi(L^b, T^b, S^b) \leq \frac{\mu_{\max}}{\epsilon_{\min}} \cdot (2 \ln k + 1) \cdot \phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)$

Combining Lemma 9 with Theorem 4, we get:

**Theorem 6:** The expectation value and standard deviation of $\Psi(L^b, T^b, S^b)$ are within $2.5\mu j$ and $3.5\mu j$ times of $E[\varpi(L^*, T^*, S^*)]$, respectively, where $\mu = \frac{\mu_{\max}}{\epsilon_{\min}} \cdot \lambda (\ln \Delta + 1)(2 \ln k + 1) = O(\ln \Delta \ln k)$.

D. The Price of Power Adjustability and Link Reliability
Having shown that involving more parameters in a data dissemination problem does not affect the approximability of the problem, we do need to pay a price in terms of the coefficients hidden behind the $O$-notation. Fortunately, we shall show that this price is acceptable given the actual parameters suggested by the real-life WSNs. Taking the CC2520 RF transceiver [29] (the most up-to-date radio interface of a sensor node) as an example, it offers 9 tx-power levels between (-18)dBm and 5dBm, but the most usable levels lie between (-7)dBm and 5dBm (as a very low tx-power leads to unstable link performance easily affected by ambient interference from, for example, WiFi transmissions). This suggests that $\frac{\mu_{\max}}{\epsilon_{\min}} \approx 15.8$ in practice. Moreover, the link quality lower bound $\Delta$ is often decided by a WSN routing protocol, in a way that a link whose
quality is below $\Lambda$ is not used in carrying data traffic. Whereas the value of $\Lambda$ depends on a specific protocol, the fact that $\lambda$ is a logarithmic function of link quality implies that its value can be negligible. For example, the value of $\lambda$ is only increased from 1.62 to 5.48 if $\Lambda$ is reduced by 4 times from 0.8 to 0.2.

VI. PERFORMANCE EVALUATION

We conduct extensive simulations to evaluate the performance of our algorithms. In the simulations, we randomly deploy $|V|$ nodes in a square of $10^2 |V| \text{m}^2$ and set the maximum tx-power ($\varepsilon_{\text{max}}$) to 15.0dBm as that in [6]. The source/destination nodes are randomly selected in $V$. Following [13], the link qualities are set by considering the path loss channels and shadowing effects [30]. All the reported data in our figures are the average of 50 simulation results.

We compare our algorithms with three representative algorithms in the literature: (1) The multicast algorithm (MEERMRL) proposed by [6] for AAWNs; (2) The multicast algorithm (TCS) proposed in [3] for DC-WSNs with perfectly reliable links; (3) The broadcast algorithm (Opportunistic Flooding, or OF) proposed in [13] for DC-WSNs with unreliable links. Since all these algorithms require a fixed tx-power, we fix the tx-power of all nodes to 15.0dBm when comparing with them. For fairness sake, we apply the simple node-coloring based collision-free scheduling scheme proposed by [3] to all the algorithms except OF, which uses a backoff scheme to avoid collisions [13].

In Figure 4, we compare MC-ERDD (Algorithm 1) with the MEERMRL algorithm, where nodes are set always-active. The number of network nodes is set to 100 and 200 in Figure 4(a) and Figure 4(b), respectively, and the percentage of destination nodes scales from 10% to 90% with an increment of 20%. Though multicast is implemented by unicast in [6], we also implement an adapted version of MEERMRL (denoted by MEERMRL-W) that leverages WBA for multicasting but still applies the same multicast tree that MEERMRL constructs. It can be seen from Figure 4 that MC-ERDD outperforms both MEERMRL and MEERMRL-W significantly, which can be explained by the reason that WBA is neglected when the MEERMRL multicast tree is constructed in [6].

In Figure 5, we compare MC-ERDD with the TCS algorithm. Nodes are duty-cycled for both cases, with the length of working period set to 20. Each node may have at most 10 active time slots within a working period (50% duty-cycle ratio) and these slots are randomly selected. MC-ERDD (again) significantly outperforms TCS, simply because TCS is only duty-cycle-aware but not link-quality-aware, it hence may select poor links for data transmission, resulting in higher energy consumption for guaranteeing reliability.

In Figure 6, we compare BC-ERDD (Algorithm 2) with the OF algorithm for broadcasting, where the duty-cycle setting is the same as that in Figure 5. The parameters for implementing OF are based on the experimental results reported in [13], i.e., we set $l_{th} = 0.9$ and $p = 0.5$, in order to obtain the best performance of OF. Following the testing method in [13], we scale the packet delivery ratio PDR from 75% to 99%, and record the total tx-power consumption when certain PDR is attained. It can be seen that BC-ERDD outperforms
OF significantly regardless of the attainable PDR, although opportunistic links are exploited in OF to reduce the energy consumption. Both results in Figure 4 and 6 suggest that, for the sake of energy efficiency, WBA is better to be taken into account at the tree construction phase; any “patching” after a WBA-oblivious tree construction may not achieve the best performance. Also, OF constructs its broadcast tree regardless of nodes’ duty cycling, which further degrades its performance in general DC-WSNs.

As OF is designed for WSNs with very low duty-cycle, we further compare BC-ERDD with OF under different Duty-Cycle Ratios (DCRs) in Figure 7, where we set $|V| = 100$, $|I| = 20$ and fix the PDR to 99%. We scale nodes’ maximum number of active time slots within a working period from 2 to 10 with an increment of 2, hence the DCR varies from 10% to 50%. As expected, BC-ERDD outperforms OF under all these ratios, although the advantage of BC-ERDD is more conspicuous under higher ratios, as WBA becomes less significant if nodes sleep most of the time under low DCRs.

Finally, we study the impact of power-adjustability on energy consumption. Since we are not aware of any other multicast/broadcast algorithms for DC-WSNs with power-adjustability, we study only our own algorithms in Figure 8, where Figure 8(a) is for MC-ERDD and Figure 8(b) is for BC-ERDD. We use “FIXED” to denote the case where each node’s tx-power is fixed to 15.0dBm, and use “ADJUST” to denote the case where each node can adjust its tx-power to three levels, namely 11.0dBm, 13.0dBm and 15.0dBm. The number of network nodes in Figure 8 is set to 100, and nodes are duty-cycled as that in Figure 5 and Figure 6. It can be seen from Figure 8 that introducing power-adjustability can significantly improve energy-efficiency, because nodes can use lower tx-power to communicate with nearer nodes, hence reducing energy consumption.

VII. CONCLUSION

We have studied the energy-efficient reliable data dissemination problem in DC-WSNs with unreliable links. We seek to minimize the total expected tx-power consumption for reliable multicasting/broadcasting. Due to the NP-hardness of the problem, we have proposed approximation algorithms with provable performance ratios based on a novel data structure named as Time-Reliability-Power space. To the best of our knowledge, these algorithms are, on the one hand, the first to holistically take into account various aspects including duty-cycling, wireless broadcast advantage, unreliable links and power-adjustability, and on the other hand, to provide guaranteed performance bounds for energy-efficient reliable data dissemination in DC-WSNs. In addition to the theoretical contribution, our extensive simulation results also demonstrate the practical efficiency of our algorithms, by comparing with representative proposals in the literature.

REFERENCES

\[ E[X_u(l, Q)] = \sum_{i=0}^{\infty} \left[ 1 - \mathbb{P}\{X_u(l, Q) \leq i\} \right] \]
\[ = \sum_{i=0}^{\infty} \left[ 1 - \prod_{v \in S(u,t)} p_{uv}(l)(1 - p_{uv}(l))^{y_v - 1} \right] \]
\[ = \sum_{i=0}^{\infty} \left[ 1 - \prod_{v \in Q} (1 - (1 - p_{uv}(l))^{i}) \right]. \] (4)

For any \( u \in \mathcal{V}(T) \) and any \( t \in \mathcal{T} \) satisfying \( S(u,t) \neq \emptyset \), let \( \delta = \delta(u,t) = \min\{p_{uv}(l)\forall v \in S(u,t)\} \) and \( \delta = \delta(u,t) \). Let \( H(\cdot) \) be the harmonic number. Suppose \( c_1 \neq 0 \) (otherwise the proof is trivial), we have:

\[ \frac{E[X_u(l, S(u,t))]}{c_1} \leq H(\delta_1) \left( 1 - \frac{1}{\ln c_1} \right). \] (5)

Now let \( g(x) = x[1 - (1 - c_1^i)^{\delta}] \). We shall prove \( g(x) \leq \frac{H(\delta_1)}{\ln c_1}, \forall x \in [0, +\infty) \). Let \( y = 1 - c_1^i \in [0, 1] \), so

\[ g(x) = (1 - y^\delta) \ln(1 - y) - \sum_{j=1}^{\infty} y_j^i \frac{1 - y_j^\delta}{\ln c_1} \]
\[ = -\frac{1}{\ln c_1} \sum_{j=1}^{\infty} y_j^i \frac{1 - y_j^\delta}{\ln c_1} \]
\[ = -\frac{1}{\ln c_1} \sum_{j=1}^{\infty} \frac{y_j^i}{\delta_1} \frac{1 - y_j^\delta}{\ln c_1} \]
\[ \leq -\frac{1}{\ln c_1} \sum_{j=1}^{\infty} \frac{y_j^i}{\delta_1} \frac{1}{\ln c_1} \] (7)

**APPENDIX**

**Proof of Lemma 1 and 2:** Lemma 1 is proved by equation (4):

**6** 

**7**


Using (6) and (7), we get
\[
\sigma^2[\chi_u(l, S(u, t))] \\
\leq -\frac{2H(\delta_1)}{\ln c_1} + 2 \int_0^\infty g(x)dx \\
= -\frac{2H(\delta_1)}{\ln c_1} - 2 \int_0^1 \ln(1-y) \frac{\delta_1-1}{\ln^2 c_1} \sum_{j=0}^{\delta_1-1} y^j dy. \\
\tag{8}
\]

Meanwhile, we have
\[
\frac{1}{j+1} \int_0^1 \left( (j+1)y^j \ln(1-y) + \sum_{i=0}^j y^j \right) dy \\
\leq \frac{\delta_1-1}{j+1} \int_0^1 y^{j+1} \ln(1-y)dy - H(j+1) \\
= -\sum_{j=1}^{\delta_1-1} \frac{H(j)}{j}.
\]
Plugging (9) into (8) gives us
\[
\sigma^2[\chi_u(l, S(u, t))] \\
\leq -\frac{2H(\delta_1)}{\ln c_1} + 2 \int_0^\infty g(x)dx \\
\leq 2H^2(\delta_1) \cdot \left( 1 - \frac{1}{\ln c_1} \right)^2 \\
\leq 2(\ln \Delta + 1)^2 \cdot \max_x \{\gamma_{uv}(l) | v \in \tilde{S}(u, t)\}. \\
\tag{10}
\]

As the random variables in \{\chi_u(L(u), S(u, t)) | u \in \lambda(\tilde{T}), t \in \tilde{T}\} are mutually independent, using (10) we have
\[
\sigma^2[\chi(L, \tilde{T}, \tilde{S})] \\
\leq \left( \sum_{u \in \lambda(\tilde{T})} \sum_{t \in \tilde{T}} \sigma[\chi_u(L(u), S(u, t))] \right)^2 \\
\leq \left[ \sqrt{2} (\ln \Delta + 1) \phi(L, \tilde{T}, \tilde{S}) \right]^2,
\]
hence \(\sigma[\chi(L, \tilde{T}, \tilde{S})] \leq \sqrt{2} (\ln \Delta + 1) \phi(L, \tilde{T}, \tilde{S})\). \(\square\)

**Proof of Lemma 3:** For any \(u \in \lambda(T^m)\) and any \(t \in \tilde{T}\) satisfying \(S^*(u, t) \neq \emptyset\), let \(l^* \in \Lambda(u), c_2 = 1 - \max\{p_{uv}(l^*) | v \in S^*(u, t)\}\) and \(\delta_2 = |S^*(u, t)|\). Suppose \(c_2 \neq 0\) (otherwise the proof becomes trivial), we have:
\[
\min\{\gamma_{uv}(l^*) | v \in S^*(u, t)\} \geq \frac{\gamma_{uv}}{\lambda} \\
\geq \frac{\gamma_{uv}}{\lambda} = \frac{1}{\lambda} \cdot \max\{\gamma_{uv}(l^*) | v \in S^*(u, t)\}. \\
\tag{11}
\]
Since \(1 - \frac{1}{\ln c_2} = \min\{\gamma_{uv}(l^*) | v \in S^*(u, t)\}\), we get
\[
\mathbb{E}[\chi_u(l^*, S^*(u, t))] \\
= \sum_{i=0}^{\infty} \left[ 1 - \prod_{v \in S^*(u, t)} (1 - (1 - p_{uv}(l^*))^i) \right] \\
\geq \sum_{i=0}^{\infty} [1 - (1 - c_2^i)^{\delta_2}] \geq \int_0^\infty 1 - (1 - c_2^i)^{\delta_2} dx \\
= -\frac{H(\delta_2)}{\ln c_2} \geq -\frac{\ln(\delta_2 + 1)}{\ln c_2} \\
\geq \ln 2 \cdot \left( \min\{\gamma_{uv}(l^*) | v \in S^*(u, t)\} - 1 \right). \\
\tag{12}
\]
Using (11), (12) and \(\mathbb{E}[\chi_u(l^*, S^*(u, t))] \geq 1\), we have
\[
(1 + \ln 2) \mathbb{E}[\chi(L^*, T^*, S^*)] \geq \frac{\ln 2}{\lambda} \cdot \phi(L^*, T^*, S^*).
\]
Hence the lemma follows. \(\square\)

**Proof of Lemma 4:** Let us call the edge-images of the 2-tuples constructed by rules M1 and M2 the “M1-edges” and the “M2-edges”, respectively. For \(\forall u \in \lambda(T^m)\) and \(\forall v \in C_{T^m}(u)\), we have:
(1) If \(\{(u', v) \in \mathcal{W}_M | u' \in U' \cap \partial(u)\} = \emptyset\), then \((u, v)\) must be a M2-edge. According to M2, we can always find a satisfactory \((v, \bar{t}, \bar{r}, \bar{l})\) in line 6. Hence there exists \(\exists (v, \bar{t}, \bar{r}, \bar{l}) \in \mathcal{A}(v)\) such that \(v \in \partial(l, u, l)\) and \(v \in S^m(u, t)\) according to lines 6-7 and lines 12-13 of Algorithm 1.
(2) If \(\{(u', v) \in \mathcal{W}_M | u' \in U' \cap \partial(u)\} \neq \emptyset\), then \((u, v)\) can be a M1-edge or a M2-edge. In either case, there must exist an element \((u, t, r', l) \in U' \setminus U\) such that \(v \in \partial(l, u, l)\) and \(v \geq \gamma_{uv}(l)\). Hence, we can know \(v \in \bigcup_{u \in T} S^m(u, t)\) according to lines 12-13 of Algorithm 1.

The above reasoning actually proves \(C_{T^m}(u) \subseteq \bigcup_{u \in T} S^m(u, t)\) for any \(u \in \lambda(T^m)\). Other conditions for \((L^m, T^m, S^m)\) to be a VDS can be proved similarly based on rules M1 and M2. Hence the lemma follows. \(\square\)

**Proof of Lemma 5:** Let \(K(u, t) = \{u, t, r', l' \in \mathcal{U}\}\). According to lines 8-18 of Algorithm 1, for any \(u \in \lambda(T^m)\), \(t \in \tilde{T}\) and any \(v \in S^m(u, t) \neq \emptyset\), there must exist \((u, t, r, l) \in \mathcal{U} \setminus \mathcal{V}\) such that \(l \leq L^m(u)\) and \(\gamma_{uv}(l) \leq r'\). Hence we get \(\gamma_{uv}(L^m(u)) \leq \gamma_{uv}(l) \leq r' \leq \omega((u, t, r, l))/\varepsilon_{min}\) and \(\max\{\gamma_{uv}(L^m(u)) | v \in S^m(u, t)\} \leq \omega(K(u, t))/\varepsilon_{min}\). Therefore
\[
\phi(L^m, T^m, S^m) \leq \sum_{u \in \lambda(T^m)} \sum_{t \in \tilde{T}} L^m(u) \cdot \omega(K(u, t))/\varepsilon_{min} \\
\leq \frac{\varepsilon_{max}}{\varepsilon_{min}} \cdot \omega(U).
\]
Note that any two different nodes \(u \in \mathcal{V}\) and \(v \in \mathcal{V}\) cannot be adjacent in the TRS space. Besides, rules M1-M2 imply that when \(u\) is adjacent to certain element \(y \in \partial(v)\), there must exist a symmetric \(x \in \partial(u)\) which has the same weight as \(y\). Therefore, from lines 4-14 of Algorithm 1 we know that the total weight of the elements in \(U' \cup U\) is no more than that of \(U'\) after Algorithm 1 finishes. Hence, it follows that \(\omega(U) \leq \omega(U')\) and the lemma is proved. \(\square\)

**Proof of Lemma 6:** Let \(W_1 = \mathcal{R} \cup \{s\} \cup \{(u, t, r, l) | u \in \lambda(T^m) \wedge t \in \tilde{T} \wedge l = L^*(u) \wedge S^*(u, t) \neq \emptyset \land [r =

max\{γ_{uv}(l)|v \in \tilde{S}^*(u, t)\}\}. Clearly we have \(w(W_1) = \phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)\).

Initialize \(W_2\) to the empty set, and then we check each \(v \in \lambda(\hat{T}^*)\). Suppose that the parent node of \(v\) is \(u\). Since \((L^*, T^*, S^*)\) is a VDS, we can find \(y_1 = \langle u, t_1, r_1, l_1 \rangle \in W_1\) and \(y_2 = \langle v, t_2, r_2, l_2 \rangle \in W_1\) such that \(v = \partial_t(u, l_1)\), \(r_1 \geq \gamma_{uv}(l_1)\) and \(u \in N_v(l_2)\). Let \(t_3\) be an arbitrary time slot in \(A(u)\) and we add \(y_3 = \langle v, t_3, \gamma_{uv}(l_2), l_2 \rangle\) into \(W_2\). Note that \(y_1\) and \(y_3\) are accessible to each other according to \(M_1\), and \(w(y_3)\) is no more than \(\lambda \cdot w(y_2)\). After all the nodes in \(\lambda(\hat{T}^*)\) are checked, \(w(W_2)\) is at most \(\lambda \cdot w(W_1)\). Moreover, for any pair of nodes in \(R \cup \{s\}\), there exists a TRP path between them which is embraced by \(W_1 \cup W_2\).

Suppose that the weight of an optimal NWST spanning \(R \cup \{s\}\) in the TRP space \((\hat{L}^*, \hat{T}^*, \hat{S}^*)\) is \(w^*\). Since the approximation ratio of the NWST algorithm used in line 1 of Algorithm 1 is \(2 \ln k + \log n\), we get

\[
w(\hat{L}^*) \leq 2 \ln k \cdot \log n \leq 2 \ln k \cdot w(W_1 \cup W_2) \leq 2 \ln k \cdot (1 + \lambda \cdot w(W_1)) = 2 \ln k \cdot (1 + \lambda) \cdot \phi(\hat{L}^*, \hat{T}^*, \hat{S}^*).
\]

Hence the lemma follows.

Proof of Lemma 7: According to line 8 of Algorithm 2, it can be seen that \(T^b\) is a broadcast tree in \(G\) rooted at \(s\). Next, we prove that \(L^b\) and \(S^b\) are valid for \(T^b\). We note that, for any \(u \in \lambda(\hat{T}^b)\) and any \(v \in C_{T^b}(u)\), there must exist \(\langle u, t_1, r_1, l_1 \rangle \in X\) and \((v, t_2, r_2, l_2) \in \theta(v)\) such that \(((u, t_1, r_1, l_1), (v, t_2, r_2, l_2)) \in W_{b}\) according to rules \(B_1\) and \(B_2\). This implies \(v \in \partial_t(u, l_1)\), \(r_1 \geq \gamma_{uv}(l_1)\) and \(u \in N_v(l_2)\). Based on this observation and line 9 of Algorithm 2, it can be proved that \(\forall t \in \mathcal{I} : S^b(u, t) \subseteq \partial_t(u, L^b(u)) \cap C_{T^b}(u)\) and \(C_{T^b}(u) \subseteq \bigcup_{t \in \mathcal{I}} S^b(u, t)\) using a similar reasoning with that in the proof of Lemma 4. Hence the lemma follows.

Proof of Lemma 8: Let the path from \(s\) to any \(z \in Z_j\) in \(\hat{T}^*\) be \((u_1, u_2, ..., u_m)\) where \(u_1 = s\) and \(u_m = z\). For \(1 \leq i \leq m\), we can find \(x_i = \langle u_i, t_i, r_i, l_i \rangle\) such that \(t_i \in \mathcal{I}\), \(u_{i+1} \in S^b(u_i, t_i)\), \(l_i = L^b(u_i)\) and \(r_i = \max\{\gamma_{uv}(l_i)|v \in S^b(u_i, t_i)\}\). Let \(P_z = \{s, x_1, x_2, ..., x_{m-1}, z\}\). According to the construction rules of \(W_{b}\), we know that there exists \(P_z \subseteq \bigcup_{z \in Z_j} P_{z}\) such that \(P_{z}\) embraces a TRP path from \(s\) to any \(z \in Z_j\) and \(w(P_{z}) \leq \phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)\). Then, using a similar method as the “spider decomposition” method in [27], \(P_{z}\) can be partitioned into subsets \(P_{z1}, ..., P_{zw}\) such that \(P_z \cap N_z \neq \emptyset\), then \(P_{z}\) embraces disjoint TRP paths from certain \(r_i \in P_{z}\) to the nodes in \(P_{z} \cap Z_j\) (\(1 \leq i \leq w\)). Based on line 3 of Algorithm 2 we know that, if \(P_{z} \cap Z_j \neq \emptyset\), then

\[
\frac{w(X_{j+1}) - w(X_j)}{|D_{j+1}|} \leq \text{avl}(r_{ti}, P_{z} \cap Z_j) \quad (1 \leq i \leq w) \quad (13)
\]

Combining (13) with

\[
\sum_{i=1}^{w} \text{avl}(r_{ti}, P_{z} \cap Z_j) \cdot |P_{z} \cap Z_j| \leq w(P_{z}) \leq \phi(\hat{L}^*, \hat{T}^*, \hat{S}^*)
\]

and \(\sum_{i=1}^{w} |P_{z} \cap Z_j| = |Z_j|\), the lemma follows.
Kai Han received the B.S. and Ph.D. degrees in computer science from the University of Science and Technology of China, Hefei, China, in 1997 and 2004, respectively. From 2005 to 2008, he was a post-doctoral research fellow at the School of Mathematics, University of Science and Technology of China.

He is currently an associate professor at the School of Computer Science, Zhongyuan University of Technology, China. This work is done when he works as a visiting fellow at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include wireless ad hoc and sensor networks, social networks, combinatorial and stochastic optimization, machine learning theory, as well as algorithmic game theory. He is a Member of both IEEE and ACM.

Jun Luo got his PhD degree in computer science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland, in 2006. From 2006 to 2008, he has worked as a post-doctoral research fellow in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the School of Computer Engineering, Nanyang Technological University in Singapore, where he is currently an Assistant Professor. His research interests include wireless networking, distributed systems, multimedia protocols, network modeling and performance analysis, applied operations research, as well as network security. He is a Member of both IEEE and ACM.

Liu Xiang received the B.S. degree in Electronic Engineering from Tsinghua University, Beijing, China, and the Ph.D. degree in Computer Science from Nanyang Technological University, Singapore. She is currently a research scientist with Institute for Infocomm Research (I2R), A*STAR, Singapore. Her research interests include network modelling, analysis, and optimization for wireless networks and smart grid.

Mingjun Xiao is an associate professor in the School of Computer Science and Technology at the University of Science and Technology of China (USTC). He received his Ph.D. degree from USTC in 2004. In 2012, he was a visiting scholar at Temple University. He has served as a reviewer for many journal papers. His main research interests include wireless sensor networks, delay tolerant networks and mobile social networks.

Liusheng Huang is a professor in the School of Computer Science and Technology at the University of Science and Technology of China. He received his M.S. degree in computer science from the University of Science and Technology of China in 1988. He serves on the editorial board of many journals. He has published 6 books, and more than 200 papers. His main research interests include wireless sensor networks, delay tolerant networks and Internet of things.