

Minimizing Age of Information in Multi-hop Energy-Harvesting Wireless Sensor Network

Kunyi Chen, Fatma Benkhelifa, Hong Gao, Julie McCann and Jianzhong Li

Abstract—Age of information (AoI), a metric measuring the information freshness, has drawn increased attention due to its importance in monitoring applications in which nodes send time-stamped status updates to interested recipients, and timely updates about phenomena are important. In this work, we consider the AoI minimization scheduling problem in multi-hop energy harvesting (EH) wireless sensor networks (WSNs). We design the generation time of updates for nodes and develop transmission schedules under both protocol and physical interference models, aiming at achieving minimum peak AoI and average AoI among all nodes for a given time duration. We prove that it is an NP-Hard problem and propose an energy-adaptive, distributed algorithm called MAoIG. We derive its theoretical upper bounds for the peak and average AoI and a lower bound for peak AoI. The numerical results validate that MAoIG outperforms all of the baseline schemes in all scenarios and that the experimental results tightly track the theoretical upper bound optimal solutions while the lower bound tightness decreases with the number of nodes.

Index Terms—Age of Information, Energy Harvesting WSN, Multi-hop WSN.

I. INTRODUCTION

WIRELESS sensor networks (WSNs), bringing the fundamental component of the Internet of Things (IoT), are deployed in a wide range of environments to collect data from the physical world, providing information for remote monitoring, target tracking and anomalous phenomena detection. Frequent battery replacement can be extremely expensive or even impractical in hard to reach locations and cause material waste. As such, energy harvesting (EH) technologies can be exploited where devices can harvest energy from ambient sources such as solar power and radio frequency (RF) signals. Energy Harvesting devices have enabled WSN (EH-WSN) to reach energy self-sustainability and operate perpetually [1]. However, the energy harvesting capacities of such devices are limited. Moreover, environmental energy is inherently time-varying and exhibits a high degree of heterogeneity over different nodes. These properties bring challenges to networking algorithm design which has been attracting a growing research interest in recent years [2]–[7].

Most of these works focus on maximizing throughput (e.g., [2]–[4]) or minimizing transmission latency (e.g., [5]–[7]). The key issue in the throughput maximisation problem is to maximise the volume of sensing and transmission data as much as

Kunyi Chen, Hong Gao and Jianzhong Li are with the Department of Computing, Harbin Institute of Technology, Harbin, China. E-mail: {kunyichen, honggao, lijzh}@hit.edu.cn (Corresponding author: Hong Gao)

Fatma Benkhelifa is with the School of Computing, Electronics and Mathematics, Coventry University, Coventry, CV1 5FB, United Kingdom. E-mail: ad8904@coventry.ac.uk.

Julie McCann is with the Department of Computing, Imperial College London, London, UK. E-mail: j.mccann@imperial.ac.uk

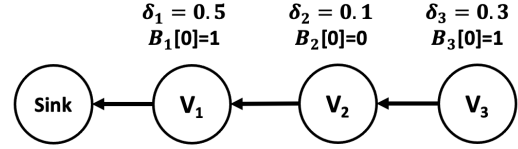


Fig. 1. A Line EH-WSN consisting of a sink and 3 nodes, where for $i \in \{1, 2, 3\}$, δ_i is the energy harvesting rate and $B_i[0]$ is the initial battery level. For object i , $AoI_i(t) = t - g_i^{latest}$, where t is the current time and g_i^{latest} is the generation time of the freshest updated from object i that received by the sink.

possible; while the latency minimisation problem investigates how to transmit those packets promptly. For both, the problem of when to generate an update packet is not addressed directly. Earlier, traditional sensing applications could not afford real-time data transmission, or indeed, did not require it. Therefore throughput was the more critical metric.

However, with the proliferation of online sensor-based systems such as in autonomous systems, security systems, control and real-time applications, the generation time of an update packet have a significant effect on the freshness of information about the objects they monitor. This is especially important in applications where timely updates regarding a targeted phenomena are essential for a control function to enact upon that data and where this sense/actuate reaction time is key. Recently, a new metric called age-of-information (AoI) was proposed to measure the timeliness of this information update [8]. Defined as the time elapsed since the generation time of the last packet received by the sink, AoI describes the timeliness of the process of the target objects.

To highlight the difference between AoI, latency and throughput, we present in Fig. 1 an example of a line topology EH-WSN consisting of a sink and three nodes v_1, v_2, v_3 monitoring three objects. The initial battery level of each node v_i is denoted by $B_i[0]$. For simplicity, only in this example, we assume that for each node v_i , its energy harvesting rate δ_i is constant during the time duration and it spends one unit of energy to send or receive one packet. When considering latency minimisation, the goal is to minimise the time cost for transmitting at least one packet from each node v_i to the sink. Since v_2 needs to send out its own packet and relay a packet for v_3 , the total cost is 3 units of energy which takes 30 time slots to harvest. Similarly, v_1 needs to send out its own packet and relay two packets for v_2 and v_3 , the total cost is 5 units of energy which takes 10 time slots for it to harvest. The minimum latency is 32 time slots, while the throughput is three packets. Case 1: If all nodes generate their packets $\{p_1, p_2, p_3\}$ at time slot 0, all packets are received by the sink at time slot $t = 32$. The information is stale by $t = 32$. The AoIs of the

objects are $\{32, 32, 32\}$. Case 2: If we set the generation time of $\{p_1, p_2, p_3\}$ to be the beginning of time slots $\{27, 28, 30\}$. Let v_1 send p_1 to the sink at $t = 27$, v_2 sends p_2 to v_1 at $t = 28$ and v_1 send p_2 to sink at $t = 29$. The remaining time slot for p_3 . By $t = 32$, AoI of the three objects are $\{5, 4, 2\}$. Hence, this example clearly shows that even when the latency and throughput are fixed, we can achieve different AoI by adjusting the generation time of packets. We can also see that generating an update when all relays have enough energy can obtain smaller AoI.

A. Related Works

Over the past few years, there has been a growing body of research works on the AoI minimization scheduling problem EH-WSN [9]–[14] and in battery powered WSN (BP-WSN) in [15]–[24]. EH-WSN research, [9]–[13] focuses on single-hop networks where their source nodes communicate directly to destination nodes. The problem here is to determine the generation time of the updates and the time to activate the links between the source nodes and their one-hop far destination nodes. [14] considered a two-hop EH-WSN where a source is communicating to a destination through a relay and propose age-minimal policies to solve the AoI minimization problem. However, a two-hop EH-WSN is a very simple topology; the above methods cannot be easily extended to multi-hop scenarios.

In applications like remote monitoring, sensors are scattered across a broad area far away from the control centre. In such scenarios, single-hop transmission is unsuitable or even not feasible. Some works considered AoI minimization for multi-hop BP-WSN [21]–[24]. [21] focuses on a simple solution space where links can only be activated according to a stationary probability distribution. Under their settings, the optimal stationary scheduling policy was derived.

[22] considers the scenario where a sender periodically sends a collection of data to a receiver. The objective is to minimise peak/average AoI subject to throughput requirements. In that paper, a pseudo-polynomial optimal algorithm is proposed. [23] considers a scenario where each node is both a source and a monitor of information; here all nodes can receive fresh updates from all other nodes in the network. Based on this setting, lower bounds on the peak and average AoI are derived. [24] minimized the age of a single information flow in an interference-free multi-hop BP-WSN. The authors assumed that the transmission times of packets are exponentially distributed, and then focused on finding the optimal queuing policy. The Last-Generated First-Served policy was shown to be optimal when preemption is allowed.

However, [21]–[24] focus on BP-WSN where nodes are available for transmission at any time. To the best of our knowledge, there is no existing paper investigating the AoI minimisation problem in multi-hop EH-WSN. The focus on EH-WSN brings the added complexity inherent in the collaborative nature of the network nodes and heterogeneous energy harvesting opportunities across the network. The extra challenges are:

- 1) The schedule is required to be energy-adaptive and time-varying in terms of path choice and packet update frequency. Since relaying packets costs energy, if a

node in a poor energy condition (e.g. v_4 in Fig .4) is assigned to be the relay for many other nodes, the delay will be huge, as will the AoI. In addition, the environmental resource is time-varying. For example, in a solar-powered WSN, for a given duration a node has a high energy harvesting rate meaning it can relay packets for other nodes. Later, the node might be in the shade of a tree or a building, and its energy harvesting rate may decrease greatly. In this scenario, using a fixed data gathering tree leads to large latency. On the other hand, the updating frequency on sunny days should be higher than for cloudy days; as on sunny days the node energy harvesting rates are higher.

- 2) The scheduling algorithm should be distributed and decentralised. This is because if each node were to send their current battery level to the sink this will bring enormous energy and time cost due to the communication and potential interference.
- 3) In addition, avoiding transmission interference in a distributive way is also challenging as the transmissions among EH-nodes depend on the energy conditions of those nodes which is time-varying.

B. Contributions

To address the aforementioned problems, two key questions need to be investigated: (i) For each node (i.e. each target area), under which conditions do we generate an update packet? (ii) For a generated packet, how do we schedule nodes to transmit to the sink? To address (i) and (ii), both energy and interference issues need to be considered. In this paper, we propose that: (1) A node v_i senses and generates a new packet only if all nodes in its path to the sink have enough estimated energy to relay their current workload and the new packet, and (2) divide the nodes into interference-free groups and maximise the simultaneous transmissions to have freshly generated packets. To the best of our knowledge, this paper is the first work addressing the AoI optimisation scheduling problem in multi-hop EH-WSN. Our main contributions are as listed below:

- 1) We formulate the maximum peak AoI Minimization scheduling (EH-PAMS) problem and the average AoI Minimization Problem (EH-AAMS) in multi-hop EH-WSN. Specifically, each node is a source node providing updates to the sink about a target area. The goal is to minimise the peak/average AoI for all target areas (i.e. all nodes) in a long time duration $[0, T]$.
- 2) We propose a distributed and energy-adaptive algorithm called MAoIG where nodes make updates cyclically; in each cycle, every node makes one update.
- 3) We prove that: a) The peak AoI achieved by MAoIG verifies $AoI_{MAoIG}^{Peak}(T) \leq RG^{Max} + 2R$, where RG^{Max} is the time cost of the longest updating cycle and R is the depth of the network, that is, the hop-count of the node that has the farthest distance to the sink. b) The average AoI achieved by MAoIG verifies $A_{MAoIG}^{Ave}(T) \leq \frac{T}{2K} + 2R$, where R is the depth of the network and K is the number of update cycles during the time duration $[0, T]$.

TABLE I
SYMBOLS

Symbols	Meanings
n	The number of nodes in the network
$B_i(t)$	Battery level of v_i in time slot t
$\delta_i(t)$	Energy harvesting rate of v_i in time slot t
NB_i	The neighbour nodes of v_i
$A_i(t)$	AoI of object i in time slot t
p_i^k	The k -th updating packet sensed by v_i
g_i^k	The generation time of p_i^k
t_i^k	The time slot that p_i^k received by the sink
$delay_i^k$	$t_i^k - g_i^k$
$[v_i, v_j, p_x^k, t]$	The transmission schedule indicates v_i sends packet p_x^k to v_j in time slot t
CA_i	The Collision-Avoiding Matrix of v_i
$F2S_i$	If $F2S_i(t) = 1$, v_i can not send any data in time slot t
$F2R_i$	If $F2R_i(t) = 1$, v_i can not receive any data in time slot t
P_i	The transmission path for any packet generated by v_i
hop_i	The length of P_i

- 4) Solar-Powered and dedicated/environmental RF powered EH-WSN have been simulated to validate the performance of our algorithms.

C. Paper Organization

The rest of the paper is organised as follows. Section II introduces the network model. Section III presents the problem formulation. Section IV and Section V show the detailed design and theoretical analysis of the proposed algorithm for the line network and the general network, respectively. The simulation results are shown in Section VI. Section VII concludes the paper.

II. NETWORK MODEL

We consider a multi-hop wireless sensor network $G = (V, E)$ with $V = \{v_1, \dots, v_n\} \cup \{S\}$, where S is the sink and v_i s are the nodes. Each sensor v_i monitors one target object i and the data sensed by v_i can provide information about object i . Here, the communication range of each sensor is time-independent. For $v_i, v_j \in V$, if $D(v_i, v_j) \leq r$, where $D(v_i, v_j)$ is the Euclidean distance between v_i and v_j . r is the communication range, we have $(v_i, v_j) \in E$.¹ For a node v_i , we denote its neighbors as $NB(v_i) = \{v_j | (v_i, v_j) \in E, v_j \in V\}$. We consider a monitoring time duration $[0, T]$, which is partitioned into multiple equal time slots $\{1, 2, \dots, T\}$.

Let p_i^k denote the k -th updating packet sensed by v_i . p_i^k contains all measurements collected by the v_i since the time it sent its k -th update packet. A transmission schedule can be denoted by a vector $sch(v_i, v_j, p_x^k) = [v_i, v_j, p_x^k, t]$ indicating that node v_i is scheduled to send p_x^k to node v_j at time slot t . It is worth highlighting that, each node can operate as a source node and a relay node. It has a buffer to store packets

¹For complexity reduction, we assume that the weights of all links in the network are equal. In a real WSN, the quality of the link is varying in terms of lost packets, delays, error rates, etc. The quality of the link is a function of the distance between two nodes, the background noise, the transmission power, the propagation model, etc. In our paper, we chose the simplified model and assumed all links were similar. Also, the proposed algorithm can be used in the heterogeneous link quality scenario with slightly modified. The state of the art work also assumes that all links are similar, so this assumption allows us to compare fairly [7], [25].

generated by itself or received from others. Thus, p_x^k might be generated by another node v_x .

We assume that the size of each updating packet is the same. The energy consumption of sending and receiving a packet are denoted by e_s and e_r , respectively. When packets are in different sizes, our algorithm works. Details are presented in the Discussion Section. We make this assumption to simplify the description of our algorithm.

A. Energy Model

Due to limited space, we have simplified the energy model of BF-node. The details can be seen in [26], [27]. Each node harvests energy from environmental sources and stores such energy in its battery. For a node $v_i \in V$, let $B_i[t]$ denote the battery level of v_i at the beginning of time slot t and $\delta_i[t]$ is the average recharge rate of v_i during time slot t . The battery capacities of all nodes are equal. They are denoted by B_{max} . $\forall t \in \{1, 2, \dots, T\}$, $i \in \{1, \dots, n\}$, we have $0 \leq B_i[t] \leq B_{max}$. The current energy of node v_i at the beginning of time slot t is

$$B_i[t] = \min\{B_i[t-1] + \delta_i[t-1] - e_x - e_c, B_{max}\}, \quad (1)$$

where $e_x \in \{e_r, e_s, 0\}$, if v_i is sending (or receiving) a packet, $e_x = e_r$ (or e_s); otherwise, $e_x = 0$. Here e_c is energy consumed due to clocking synchronization [28], sensing, computing, broadcasting beacons and other signal processing tasks. In a real system, we always have $e_c \ll e_r$ or e_s . Without loss of generality, we ignore e_c in the remainder of this paper.

Note that equation (1) is suitable for conventional linear EH model like solar-powered EH-WSN and RF-Powered EH-WSN. It can also be used in novel non-linear EH-WSN like those introduced in [29].

Although environmental energy is time-varying, for the most widely-used energy sources: solar power and Radio Frequency signals, a node can roughly estimate its harvesting rates in the future tens of minutes. Some existing works study the energy harvesting model for EH-nodes. For example, for solar-powered nodes, [30] proposed a model which can be used to calculate the amount of harvested energy by the node from the value of Solar Radiation which is measured by the node itself. Since solar radiation will not change a lot rapidly and the weather forecast information can also be sent to these nodes every day, we can use the current measurement to predict the future recharge rate. [31] can leverage past energy observations to provide accurate estimations of future energy availability.

For Ratio-Frequency powered nodes, the energy received by a node can be estimated by its location [26], [27]. Based on these papers, we assume that an EH node knows its battery level and can predict its energy harvesting rate in tens of minutes. It is important to know that: 1) The battery level and energy harvesting rate are only known by the node itself but not by any other nodes. 2) Our scheduling algorithm is strong enough to handle the error between the predicted energy level and the real one.

B. Transmission Model

In a BF-WSN, one schedule is executable only if both of the sender and receiver have enough energy.

Definition 1. (Executable Transmission Schedule) For nodes v_i and v_j , $[v_i, v_j, p_x^k, t]$ is executable if it satisfies

- 1) $(v_i, v_j) \in E$, i.e. $v_j \in NB(v_i)$;
- 2) $B_i[t-1] - e_s \geq 0$ and $B_j[t-1] - e_r \geq 0$;
- 3) p_x^k is in v_i 's buffer.

where e_s and e_r are the energy consumption of sending and receiving a packet, respectively.

Both protocol interference and physical interference models are considered [32]. We assume that the transmission power of nodes in the network are the same and let P denote the power. In the protocol interference model, the interference range of a node is denoted by r_I . In this model, a node v_i can successfully receive a packet from v_j at time slot t if $(v_i, v_j) \in E$ meanwhile no other nodes v_k satisfying $D(v_k, v_i) \leq r_I$ is sending packet simultaneously.

In the physical interference model, a node v_i can successfully receive a packet from v_j at time slot t if $(v_i, v_j) \in E$ meanwhile the signal-to-interference-plus-noise ratio between v_i and v_j , i.e., $SINR_{ij}(t)$, is larger than a threshold β , where

$$SINR_{ij}(t) = \frac{P \cdot h(v_i, v_j) D(v_i, v_j)^{-\alpha}}{N_0 + \sum_{v_k \in S(t), v_k \neq v_i} P \cdot h(v_k, v_i) D(v_k, v_i)^{-\alpha}} \quad (2)$$

where P is the transmit power for all nodes, N_0 is the additive white Gaussian noise (AWGN), $h(v_k, v_i)$ is the multi-path block fading channel between v_k and v_i , α is the path loss exponent that satisfies usually $\alpha \geq 3$ for urban areas and $S(t)$ is the set of nodes that transmit packet at time slot t .

Even a schedule $[v_i, v_j, p_x^k, t]$ is executable, it may lead to interference with other schedules and cause packet loss.

Definition 2. (Interference Delay) For two nodes v_i and v_j , v_i has a packet p_x^k in its buffer to send to v_j . At time slot $t_{EH}(v_i, v_j, p_x^k)$, both of them have enough energy. However, it is possible that $[v_i, v_j, p_x^k, t_{EH}(v_i, v_j, p_x^k)]$ can not be implemented since there is another packet being transmitted in v_i or v_j 's interference area at that time slot. Let $t(v_i, v_j, p_x^k)$ be the time slot when v_j received the packet from v_i ,

$$\text{delay}_{In}(v_i, v_j, p_x^k) = t(v_i, v_j, p_x^k) - t_{EH}(v_i, v_j, p_x^k)$$

is the interference Delay between v_i and v_j in the transmission of p_x^k .

III. PROBLEM FORMULATION

Each time the sink receives a new packet, it will update the information about the corresponding object. For a time slot t , $g_i^{\text{latest}}(t)$ denotes the generation time of the freshest packet sensed by v_i and has been received by the sink. The AoI function of object i is

$$A_i(t) = t - g_i^{\text{latest}}(t)$$

If the sink has not received any packet from object i by time t , we have $A_i(t) = t$. As shown in Fig .2, for a monitored object i , the AoI increases with time t until the sink receives a packet containing fresher information.

Given p_i^k being the k -th update packet sensed by v_i , the generation and arrival time of p_i^k are denoted by g_i^k and t_i^k , respectively. We denote the transmission delay of p_i^k as $\text{delay}_i^k = t_i^k - g_i^k$. Notice that, if $\exists k_1 < k_2$ and $t_i^{k_1} > t_i^{k_2}$,

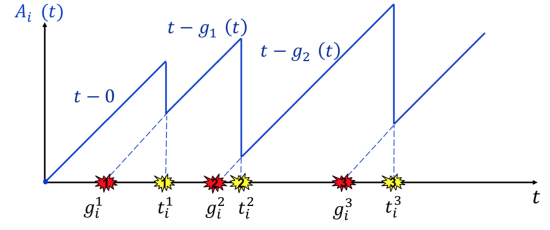


Fig. 2. Age of Information (AoI) of target area i .

the arrival of p_i^{k+1} brings no change in $A_i(t)$. This packet can be ignored and dropped. Therefore, we always have $0 = t_i^0 < t_i^1 < t_i^2 < \dots$

Lemma 1. For the AoI function $A_i(t)$, there are some properties to satisfy:

- 1) $\forall t \in \{1, \dots, T\}$, if $t = t_i^k$, $A_i(t) = t_i^k - g_i^k = \text{delay}_i^k$.
- 2) $\forall t \in [t_i^k, t_i^{k+1})$, $A_i(t) = t - g_i^k = t - t_i^k + \text{delay}_i^k$.

During the time duration $[0, T]$, let m_i denote the number of updates generated from v_i and received by the sink. In a time duration $[0, T]$, the average of AoI of target object i is

$$\bar{A}_i(T) = \frac{1}{T} \int_0^T A_i(t) dt = \frac{1}{T} \sum_{t=0}^{m_i-1} \int_{t_i^k}^{t_i^{k+1}} A_i(t) dt \quad (3)$$

The average AoI over all target areas/objects is

$$A^{\text{ave}}(T) = \frac{1}{n} \sum_{i=1}^n \bar{A}_i(T) \quad (4)$$

For an object, the peak value of AoI function during time duration $[0, T]$ is

$$A_i^{\text{Peak}}(T) = \max_{t=1, \dots, T} A_i(t) \quad (5)$$

In our paper, we consider the fairness issue among nodes and minimize the worst case of peak AoI, which is defined as follows:

$$A^{\text{peak}}(T) = \max_{i=1, \dots, n} \{A_i^{\text{Peak}}(T)\}. \quad (6)$$

When the goal is to minimize the worst case of peak AoI, we can guarantee that at any time slot, for any node, the AoI is less than the value of $A^{\text{peak}}(T)$.²

A. Mathematical Formulation

At a time slot t , let $f_i^{\text{out}}(t)$ be the packet(s) sent out by v_i and $f_i^{\text{in}}(t)$ be the packet(s) received by v_i . $p_i(t)$ denotes the packets that generated by v_i at time slot t . For $f_i^{\text{out}}(t)$, $f_i^{\text{in}}(t)$ and $p_i(t)$, we use $|f_i^{\text{out}}(t)|$, $|f_i^{\text{in}}(t)|$ and $|p_i(t)|$ to denote the size of these packets. We always have $|f_i^{\text{out}}(t)| \geq 0$, $|f_i^{\text{in}}(t)| \geq 0$ and $|p_i(t)| \geq 0$. Let \mathcal{I} be the indicator function. If $|f_i^{\text{out}}(t)| > 0$, we have $\mathcal{I}(f_i^{\text{out}}(t)) = 1$; else $\mathcal{I}(f_i^{\text{out}}(t)) = 0$. So as for $f_i^{\text{in}}(t)$ and $p_i(t)$.

²Some existing papers studied the average peak AoI minimization problem where they define the average of peak AoI as $A^{\text{Ave-Peak}}(T) = \frac{1}{n} \sum_{i=1}^n A_i^{\text{Peak}}(T)$. However, if the optimization objective is to minimize the $A^{\text{Ave-Peak}}(T)$, it may end up with the value of AoI function of some nodes at a given time slot can be significant, but the $A^{\text{Ave-Peak}}(T)$ is still small.

1) *Data Buffer Model*: Each node $v_i \in V$ maintains a data queue $Q_i(t)$ to store its own sensed data packets and data packets received from its neighbors in $NB(v_i)$. We use $|Q_i(t)|$ to denote the total size of packets in $Q_i(t)$.

Since there is no packet loss, we have

$$Q_i(t) = \cup_{k=1}^{k=t} (\{f_i^{in}(k)\} \cup \{p_i(k)\}) \setminus (\cup_{k=1}^{k=t} \{f_i^{out}(k)\}) \quad (7)$$

A node v_i can send out a packet or packets at time slot t only if $f_i^{out}(t)$ are in v_i 's data buffer or v_i just sense the new data. Therefore, we have

$$f_i^{out}(t) \subseteq Q_i(t) \cup \{p_i(t)\} \quad (8)$$

Since nodes normally have limited resources, we consider a finite buffer size Q_{max} and we have $\forall v_i \in V, \forall t \in \{1, 2, \dots, T\}, |Q_i(t)| \leq Q_{max}$.

2) *Transmission Issue*: At time slot t , for a node v_i , its battery level should be higher than the energy cost for sending or receiving data from any other node.

$$B_i(t) - |f_i^{out}(t)| * e_s - |f_i^{in}(t)| * e_r \geq 0 \quad (9)$$

In this paper, we consider half-duplex communication. That is, at a time slot t , v_i can receive a packet from another node or send out a packet to another node, but v_i cannot send and receive packets simultaneously. Therefore,

$$\mathcal{I}(f_i^{out}(t)) + \mathcal{I}(f_i^{in}(t)) \leq 1 \quad (10)$$

Let $S(t)$ denote the set of nodes sending out data at time slot t , we have

$$S(t) = \cup_{i=1}^n \mathbb{I}(f_i^{out}(t)) \quad (11)$$

where $\mathbb{I}()$ is another indicator function, if $f_i^{out}(t) > 0$, $\mathbb{I}(f_i^{out}(t)) = \{v_i\}$; if $f_i^{out}(t) = 0$, $\mathbb{I}(f_i^{out}(t)) = \emptyset$.

In the protocol interference model, for any node $v_i \in V$, if v_i is sending a packet to $Re(v_i, t) \in V$ at time slot t , then nodes in $Re(v_i, t)$ cannot send any packet at that time slot. Thus

$$|\mathbb{I}(f_i^{out}(t))| + \sum_{v_k \in IN(Re(v_i, t)) \setminus \{Re(v_i, t)\}} |\mathbb{I}(f_k^{out}(t))| \leq 1 \quad (12)$$

In the physical interference model, for any node $v_i \in V$, if v_i is sending a packet to $Re(v_i, t) \in V$ at time slot t , then node $Re(v_i, t)$ can successfully receive the packet from v_i only if the signal-to-interference-plus-noise ratio between v_i and $Re(v_i, t)$, i.e. $SINR_{i, Re(v_i, t)}(t)$ is larger than a threshold β . That is

$$\mathcal{I}(f_i^{out}(t)) \times SINR_{i, Re(v_i, t)}(t) \geq \beta \quad (13)$$

where the value of $SINR_{i, Re(v_i, t)}(t)$ can be obtained by equation (2).

$\forall v_i \in V$, at the end of the time duration $[0, T]$, the total amount of v_i 's incoming data must be no more than that of its outgoing data.

$$\frac{1}{T} \sum_{t=1}^T (|f_i^{in}(t)| + |p_i(t)| - |f_i^{out}(t)|) \leq 0 \quad (14)$$

3) *Optimization Objective*: In the sequel, we investigate the peak AoI minimization scheduling problem, namely the EH-PAMS problem. Also, we study the average AoI minimization scheduling problem, namely the EH-AAMS problem. For EH-PAMS, the optimization objective is

$$\min_{p_i(t), f_i^{in}(t), f_i^{out}(t)} A^{peak}(T) \quad (15)$$

For EH-AAMS, the Optimization Objective is

$$\min_{p_i(t), f_i^{in}(t), f_i^{out}(t)} A^{ave}(T) \quad (16)$$

For both of (15) and (16), $\forall i \in \{1, 2, \dots, n\}, \forall t \in \{1, 2, \dots, T\}, f_i^{out}(t), f_i^{in}(t), p_i(t)$ subject to Constraints (2), (7)-(14).

The challenges brought by energy harvesting are as follows: 1) In Battery-Powered WSN, nodes can transmit a packet at any time slot. However, the schedule in EH-WSN is not always executable, as we mentioned in Definition 1. 2) The transmission paths should be energy-adaptive. In BP-WSN, we can use a fixed transmission tree for packet transmissions. However, a fixed transmission tree will bring massive delays and stale information. Because it costs enormous energy to relay packets. If a node is in poor energy condition (low energy level and low energy harvesting rate), it will cost so much time to harvest enough energy. 3) The transmission path should be time-varying. Since the environmental energy is time-varying, e.g. solar-powered scenario, a node with a high energy harvesting rate in this time duration is the relay for many other nodes. It may suffer poor energy conditions in the following one hour. If it is still working as a relay for those nodes, AoI will be high. Although the problems have been formulated mathematically, it is not practical to solve the optimization problem in a centralized way. In the following, we will present distribution scheduling algorithms for EH-PAMS and EH-AAMS. Thus, their solution will be presented by a set of executable and collision-free schedules like $[v_i, v_j, p_x^k, t]$.

Theorem 1. *The EH-PAMS problem is NP-Hard.*

Proof. The proof is in Appendix A. \square

In the literature on data gathering scheduling in WSN, regarding the real requirements of a practical system, the theoretical upper/lower bounds of the performance metrics are important. For example, in [7], [25], [33], the researchers analyzed the bounds of the worst latency, in [34] researchers proposed the lower bound of weighted fairness; [35] considers the bounds of throughput. Therefore, in our paper, we will present scheduling algorithms with lower bounds of the worst cases in performance.

B. Problem Analysis

There are two fundamental questions when minimizing AoI:

- 1) For each node (i.e. each target area), under which condition do we make an update packet?
- 2) For a generated packet, how to schedule nodes to transmit to the sink?

The key to AoI minimization is to minimize the time that an update packet stays in the node buffer. In EH-WSN, there are two causes: 1) Energy Delay: The sender who is hosting the packet does not have enough energy to send it out quickly from its buffer, or the candidate relay nodes do not have enough energy to receive and relays the packet. 2) Interference Delay: The sender/receiver cannot send/receive the packet since another packet is being transmitted in its interference area, which can bring considerable delay without good scheduling. Indeed, avoiding interference delay is one of the most critical issues in scheduling strategy for WSNs.

IV. SCHEDULING ALGORITHM FOR LINE NETWORK

We first introduce how to schedule nodes in a Line network and then extend the method to general networks. An example is shown in Fig .3. The IDs assigned to all EH-nodes are assigned corresponding to their distances from the sink node v_0 . That is, we have $D(v_0, v_1) < D(v_0, v_2) < \dots < D(v_0, v_n)$. In a Line Network $\{v_0, v_1, \dots, v_n\}$, in our scheduling algorithm, for each node v_i , its intended receiver is v_{i-1} . If a node v_i device fails(lack of energy is not included since it can get recharged), node v_{i+1} will be the intended receiver of v_{i-1} . v_{i-1} will need to use higher transmission power. The transmission path of any update packet generated by v_i is $\mathcal{P}_i = \{v_i, v_{i-1}, \dots, v_0\}$.

we will introduce a distributed scheduling algorithm which is divided in to the following two parts for

In the following, we will introduce a distributed scheduling algorithm which is divided in to the following two parts:

- 1) How to avoid interference delay in a distributed way?
- 2) How to avoid energy delay in a distributed way?

To avoid interference delay, we divide nodes into subareas according to their locations and interference range. We apply the division method proposed in [7], the details of which are given as follows:

Definition 3. (Subareas in Line Network) A Line BF-WSN is divided into $M + 1$ subareas A_0, A_1, \dots, A_M . We denote the length of each subarea as d , $M = \lceil \frac{D(v_0, v_n)}{d} \rceil$ and $A_0 = \{v_0\}$. $\forall i = \{1, 2, \dots, n\}$, v_m will be assigned to A_m if and only if $(m - 1)d < D(v_i, v_0) \leq md$.

Definition 4. (Odd Subareas). All subareas $A_{2\hat{m}-1}$ with $\hat{m} = 1, 2, \dots, \lfloor \frac{M}{2} \rfloor$ are the odd subareas in a Line network with $M + 1$ subareas.

Definition 5. (Even Subareas). All subareas $A_{2\hat{m}}$ with $\hat{m} = 1, 2, \dots, \lfloor \frac{M}{2} \rfloor$ are the even subareas in a Line network with $M + 1$ subareas.

Definition 6. (Exit/Entrance Node in a subarea). Each subarea A_m has an Exit node and an Entrance node: 1) $v_i \in A_m$ is the exit node of A_m if it satisfies $\forall v_j \in A_m, D(v_i, v_0) \leq D(v_j, v_0)$; 2) $v_i \in A_m$ is the entrance node of A_m if it satisfies $\forall v_j \in A_m, D(v_i, v_0) \geq D(v_j, v_0)$.

An example is shown in Fig .3. The subareas are A_0, \dots, A_3 , where A_1, A_3 are Odd subareas, A_2 is the Even ones. A_0 contains only one node, the sink v_0 and we call A_0 as sink subarea. Specifically, for A_1 , the exit node is v_1 while the entrance node is v_4 .

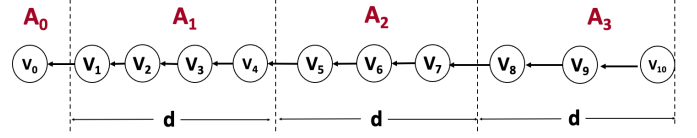


Fig. 3. Linear EH-WSN.

Definition 7. (Active State of subareas) If A_m is in active state at time slot t , nodes in A_m can be scheduled to generate, send or receive a packet in time slot t .

Definition 8. (Idle State of subareas) If A_m is in an idle state at time slot t , only the entrance node of A_m is allowed to receive beacon or packet from the existing node in A_{m+1} , meanwhile, other nodes in A_m can only harvest energy without any transmission in time slot t .

Theorem 2. The following statements show that the interference among subareas can be avoided with appropriate d :

- 1) In protocol interference model, if $d \geq r + r_I$, where r and r_I are the transmission range and interference range of an EH-node, transmissions among nodes in A_m will not interfere with any simultaneous transmissions in A_j ($1 \leq j \leq m - 2$ or $m + 2 \leq j \leq M$).
- 2) In physical interference model, if $d \geq (\frac{n * P\beta}{P_r - \alpha - N_0\beta})^{1/\alpha} + r$, similarly, transmissions among nodes in A_m will not interfere with any simultaneous transmissions in A_j ($1 \leq j \leq m - 2$ or $m + 2 \leq j \leq M$).

Proof. The theorem has been proved in [7]. \square

A collision-avoid matrix proposed in our previous work [5] is used in each BF-node. Let CA_i denote the Collision-Avoiding Matrix of node v_i and CA_i has following attributes:

- 1) **Forbidden-to-Send (F2S)** $F2S_i$. $F2S_i[t] = 1$ if $B_i(t) < e_s + e_c$ or there is a node in v_i 's interference range that has been scheduled to receive a packet from other node at time slot t .
- 2) **Forbidden-to-Receive (F2R)** $F2R_i$. $F2R_i[t] = 1$ if $B_i(t) < e_r + e_c$ or there is a node in v_i 's interference range that has been scheduled to send out a packet at time slot t .

In our algorithm, Odd subareas and Even subareas will be active alternatively. At time slot t , if t is odd (even), all Odd (Even) subareas will be active while Even (Odd) subareas will be idle. We achieve this by using our Collision-avoiding matrix. If t is Odd (Even), let node v_i be the entrant node in Even (Odd) subareas, we set $F2S_i(t) = 1$ and $F2R_i(t) = 1$. The maintenance of such a matrix will be introduced in our scheduling algorithm.

A. Scheduling Strategy for Minimizing AoI

We propose a distributed algorithm to minimize AoI for the line network, which is called (MAoIL). We will analyze its performance for both the peak AoI and average AoI. In this algorithm, nodes make updates cyclically. In the k -th cycle, each node in the network will make one update. Specifically, for each v_i , the update packet p_i^k will be transmitted through

TABLE II
SYMBOLS

Symbols	Meanings
Sch_i^k	Transmission Schedule set of p_i^k
$t_s^L(k)$	The first time slot in the k -th cycle
$t_e^L(k)$	The time slot that the sink received the last update packet in the k -th cycle
$f_{TE}(v_i, t_{start}, E)$	The number of time slots v_i needs to harvest E units of energy from time slot t_{start}
$V_{TE}(v_i, t_s^L(k))$	A vector, $V_{TE}(v_i, t_s^L(k))[j]$ is the number of time slots that the j -th nodes on path \mathcal{P}_i needs to harvest enough energy for transmissions in k -th cycle
$t_{energy}(p_i^k)$	The max value in $V_{TE}(v_i, t_s^L(k))$
$t_{LSA}(p_{i-1}^k)$	The time slot that p_{i-1}^k has been send out from v_i 's subarea
$t_{collision}(p_i^k)$	The earliest time slot that p_i^k is collision-free with p_{i-j}^k , where $j = 1, 2, \dots, i-1$
$t_{E2S}(j, p_i^k)$	For a relay $v_j \in \mathcal{P}_i$, $t_{E2S}(j, p_i^k)$ is the earliest time slot that v_j has enough energy, collision-free with other transmission and p_i^k is in its buffer
$TS_{E2R}(j, p_i^k)$	The time slot set that v_j 's intended receiver can receive a packet from v_j
$t_{E2S}(j, p_i^k)$	The time slot that v_j sends p_i^k out
$[t_s^L(k), t_e^L(k)]$	The time duration of k -th cycle
$RL^L(k)$	The length of time duration in k -th cycle
RL^{Max}	The length of the longest cycle
L_r	The set of nodes whose shortest path form it to the sink is a r -hop path
w_i^k	The current work load of v_i in k -th cycle
$t_{E2F}(v_i, k)$	The earliest time slot for v_i to harvest enough energy for finish its workload in the k -th cycle
$\mathbf{T}(V, E_T, k)$	The transmission Tree in the k -th cycle
R	The depth of the BFS tree which we term the depth of the network
RG^{Max}	The cost for the longest update cycle
K	he number of update cycles during the time duration $[0, T]$

the path $\{v_i, v_{i-1}, \dots, v_0\}$, which is denoted as P_i . The generation time g_i^k and transmission schedule set Sch_i^k will be developed in our MAoIL algorithm. The details are as follows:

- 1) Let $t_s^L(k)$ be the first time slot in the k -th cycle, we set $t_s^L(1) = 0$.
- 2) In each cycle k , the generation time g_i^k and transmission schedule set Sch_i^k of $p_i^k, i \in \{1, 2, \dots, n\}$ will be made in an ascending order. That is, the generation time and transmission time will be developed first for v_1 , then for v_2, \dots, v_n . The details of how to develop g_i^k and Sch_i^k will be explained later.
- 3) Let $t_e^L(k)$ denote the time slot that the sink v_0 has received the last update in k -th cycle, we have $t_e^L(k) = \max\{t | t \in Sch_i^k |_{time}, i = 1, 2, \dots, n\}$, where $Sch_i^k |_{time}$ is the set of all transmission time slots in Sch_i^k . The beginning time slot for the next cycle is $t_s^L(k+1) = t_e^L(k) + 1$.

In the MAoIL algorithm, when developing the generation time for each update, the key idea is to avoid energy delay. In k -th cycle, v_i will generate an update only if all nodes in $\{v_{i-1}, \dots, v_1\}$ have harvested enough energy to transmit $\{p_i^k, p_{i-1}^k, \dots, p_1^k\}$. Each node can only know its own battery level and predict its own energy harvesting rate, it know nothing about the energy condition of the nodes in its path. Thus before transmitting these updates, nodes need to send

beacons for exchanging energy conditions. Before presenting the details, we introduce some definitions first.

Definition 9. (Time-for-Harvesting-Energy Function) Consider a node v_i and a time duration $[t_{start}, T]$. $B_i(t_{start})$ is v_i 's battery level at t_{start} . $\{\delta_i(t) | \forall t \in [t_{start}, T]\}$ is the set of estimated energy harvesting rates of v_i at every time slot in $[t_{start}, T]$. For a given energy price E , we define a function for the calculation of the earliest time slot that v_i can afford E units of energy:

$$f_{TE}(v_i, t_{start}, E) = \min\{t | B_i(t) \geq E, t \geq t_{start}\}$$

$f_{TE}(v_i, t_{start}, E)$ can be calculated by using equation (1).

Lemma 2. The time complexity of Time-for-Harvesting-Energy-Function is $O(\frac{E}{\delta_{min}})$, where E is the energy cost for the transmission and δ_{min} is the minimum value of the energy harvesting rate among all nodes during the updating cycle.

Proof. The proof is in Appendix B. \square

For example, in Fig .1, if $e_s = e_r = 1$, we have $f_{TE}(v_1, 0, e_s) = 0$ and $f_{TE}(v_2, 0, (e_s + e_r) + e_s) = 30$.

Definition 10. (Time-for-Harvesting-Energy Vector) For a node v_i in a Line network, in each cycle k , let $t_s^L(k)$ be the beginning time slot for cycle k , we denote a $|\mathcal{P}_i|$ dimensional vector $V_{TE}(v_i, t_s^L(k))$.

- 1) $V_{TE}(v_i, t_s^L(k))[1] = f_{TE}(v_i, t_s^L(k), e_s)$;
- 2) $V_{TE}(v_j, t_s^L(k))[j] = f_{TE}(v_i, t_s^L(k), ((i-j) \times (e_s + e_r) + e_s))$, where $j = \{i-1, i-2, \dots, 0\}$.

For example, in Fig .1, if $e_s = e_r = 1$. For v_3 , $\mathcal{P}_3 = \{v_3, v_2, v_1\}$ and $V_{TE}(v_3, 0) = \{4, 30, 10\}$. By the time $t = \max\{V_{TE}(v_i, t_s^L(k))\} + t_s^L(k)$, each EH-node $v_j \in \mathcal{P}_i$ has enough energy to send out its update and relay updates for other nodes in $\{v_x | v_x \in \mathcal{P}_i, x > j\}$.

We assume that nodes can sense data just before sending it, i.e, for node v_i , the first transmission for p_i^k can be $[v_i, v_{i-1}, p_i^k, g_i^k]$. Given the beginning time of k -th schedule cycle $t_s^L(k)$, for p_i^k , the generation time g_i^k and transmission schedule set Sch_i^k are developed by the two phases: **Update Time Assigning Phase** which is for the energy information exchange and determining the transmission time of updates and **Transmission Phase** which is for the update transmission.

In **Update Time Assigning Phase**, we have the following steps:

- 1) **Step 1 (Energy Status Beacon):** $\forall i \in \{1, 2, \dots, n\}$, v_i calculates $f_{TE}(v_i, t_s^L(k), e_s)$, $f_{TE}(v_i, t_s^L(k), (e_s + e_r) + e_s)$, $f_{TE}(v_i, t_s^L(k), 2(e_s + e_r) + e_s)$, ..., $f_{TE}(v_i, t_s^L(k), (n-i) \times (e_s + e_r) + e_s)$. Each node v_i broadcasts a beacon containing those results to v_{i+1}, \dots, v_n .
- 2) **Step 2 (Calculate the Generation Time):** After each node v_i received these beacons, it can calculate its Time-for-Harvesting-Energy Vector $V_{TE}(v_i, t_s^L(k))$ locally. We denote $t_{energy}(p_i^k) = \max(V_{TE}(v_i, t_s^L(k)))$, i.e., $t_{energy}(p_i^k)$ is the maximum value in $V_{TE}(v_i, t_s^L(k))$. The generation time for p_i^k is $g_i^k = \max\{t_{collision}(p_i^k), t_{energy}(p_i^k)\}$, where $t_{collision}(p_i^k) = t_{LSA}(p_{i-1}^k) + 1$. $t_{LSA}(p_{i-1}^k)$ is

the time slot that p_{i-1}^k is sent out from its subarea. v_i will receive a beacon containing the value of $t_{LSA}(p_{i-1}^k)$ at the end of that time slot (see Step 2 in the Transmission Phase).

Transmission Phase has the following steps:

- 1) **Step 1 (Schedule between Sender and Receiver):** For $v_j \in \mathcal{P}_i$, when v_j has received p_i^k at time slot t^* . It will calculate its earliest available sending time slot $t_{E2S}(j, p_i^k) = \min\{t | t > t^*, F2T_j(t) = 0, B_j(t) - e_s \geq 0\}$, then send a beacon containing a transmission plan $[v_j, v_{j-1}, t_{E2S}(j, p_i^k)]$ to v_{j-1} . After receiving the plan, v_{j-1} replies with a time set containing all of its available time slots $TS_{E2R}(j, p_i^k) = \{t | t_{E2S}(j, p_i^k) \leq t \leq t_{E2S}(i, j, k) + T_w, F2R_{j-1}(t) = 0, t_{E2R} = 0\}$, where T_w is a time window. The final transmission time is $t_{T2T}(j, p_i^k) = \min\{t | t \in TS_{E2R}, F2T_i(t) = 0\}$. The schedule is $[v_j, v_{j-1}, p_i^k, t_{T2T}(j, p_i^k)]$.
- 2) **Step 2 (Update Collision Matrices):** v_j and v_{j-1} broadcast a beacon containing $[v_j, v_{j-1}, p_i^k, t_{T2T}]$ to the nodes sharing the same subarea. For nodes receiving this beacon, they update their $F2S$ or $F2R$ to avoid collisions. Specifically, if v_j is the exit node in the subarea containing v_j , v_{j-1} sends a beacon containing i and the t_{T2T} to the node v_{i+1} which will generate the next packet. v_{i+1} can derive $t_{collision}(p_{i+1}^k) = t_{T2T} + 1$.

It is worth highlighting that, in step 1 of Transmission Phases, the sender and the receiver will exchange energy information before transmitting the packet. Therefore, even the predicted energy used for calculating the generation time is inaccurate, the schedules are still feasible. The inaccuracy of energy prediction only makes nodes generating packets too early or too late.

Note that this work focus on scheduling strategy. The details about sending beacon for schedule information, sending acknowledgements after transmission are omitted. It is worth noting that at one time slot, there is at most one packet to be transmitted in one subarea. On the other hand, even the generation and schedules for each packet are developed in order, there can exist simultaneous transmissions in the Line network. For example, $[v_7, v_1, p_3^k, t]$ and $[v_7, v_6, p_7^k, t]$ can be executed simultaneously.

B. Performance Analysis

Theorem 3. In MAoIL, in each updating cycle, we have

- 1) (Time Complexity) For each node $v_i \in V$, the time complexity of the local scheduling algorithm is $O(\frac{n^2(es + er)}{\delta_{min}})$, where n is the number of nodes in the network, es and er are the energy cost of sending or receiving a packet, δ_{min} is the minimum value of the minimum energy harvesting rate among all nodes during that data collection cycle.
- 2) (Transmission Complexity) During the scheduling process, the number of beacons is $O(n^2)$.

Proof. The proof is in Appendix C. \square

Lemma 3. In MAoIL algorithm, for each packet $p_i^k, i = 1, 2, \dots, n$, delay $t_i^k = t_i^k - g_i^k$, where t_i^k is the time when p_i^k is received by the sink and g_i^k is its generation time, we have

$$\text{delay}_i^k = 2 \times \text{hop}_i, \quad (17)$$

Algorithm 1: Minimising AoI Scheduling Algorithm for Line Network (MAoIL)

- 1 **Input:** Network $G(V, E)$; Energy cost e_s, e_r, e_c ; The length of subarea d ; The length of time T .
 - 2 **Output:** A schedule set \mathcal{S}_L .
 - 3 Step 0: For all time slot t , if t is odd (even), for all node v_i expect the entrant node in even (odd) subareas. We set $F2S_i(t) = 1$ and $F2R_i(t) = 1$.
 - 4 $t_s^L(k) = 0$;
 - 5 **while** $t_s^L(k) \leq T$ **do**
 - 6 **Update Time Assigning Phase:**
 - 7 Step 1: **for** $i \in 1, 2, \dots, n$ **do**
 - 8 Calculate its
Time-for-Harvesting-Energy-Function Set and Broadcast Energy status Beacon to
 v_{i+1}, \dots, v_n .
 - 9 Step 2: After receiving the beacons, each node calculates its update generation time g_i^k .
 - 10 Initial time $t = t_s^L(k)$.
 - 11 **while** The sink has not received all
 $\{p_i^k, i = 1, \dots, n\}$ by the time t **do**
 - 12 **for** $i \in 1, 2, \dots, n$ **do**
 - 13 **if** $t == g_i^k$ OR v_i has a packet p_x^k in its
buffer **then**
 - 14 Go into **Transmission Phase:**
 - 15 Step 1: v_i sends the packet to v_{i-1} in
the earliest time slot which satisfies
that nodes are feasible and
collision-free. We get
 $[v_i, v_{i-1}, p_x^k, t_{T2T}]$ and put it into \mathcal{S}_L .
 - 16 Step 2: Send beacons to update the
collision matrices.
 - 17 Update $k = k + 1$;
 - 18 Update $t_s^L(k) = t_e^L(k) + 1$, where $t_e^L(k)$ is the
arrival time of the last packet in cycle k .
-

where hop_i is the hop-count of v_i and $\text{hop}_i = i$ in a Line Network.

Proof. The proof is in Appendix D. \square

In our algorithm, the time duration $[0, T]$ can be divided into $[t_s^L(1), t_e^L(1)] \cup [t_s^L(2), t_e^L(2)] \cup \dots \cup [t_s^L(K), t_e^L(K)] \cup [t_s^L(K+1), T]$, where $t_s^L(k)$ and $t_e^L(k)$ are the beginning and ending time slots of the k -th cycle and $t_s^L(1) = 0$. We denote $RL(k)$ to be the length of the k -th cycle and we have $RL^L(k) = t_e^L(k) - t_s^L(k)$. We denote by $RL^{Max} = \max_k \{t_e^L(k) - t_s^L(k)\}$.

Theorem 4. (Peak AoI in MAoIL) Let $AoI_{MAoIL}^{Peak}(T)$ denote the peak AoI achieved by MAoIL algorithm in time duration $[0, T]$, we have $AoI_{MAoIL}^{Peak}(T) \leq RL^{Max} + 2n$, where RL^{Max} is the time length of the longest update cycle and n is the number of nodes.

Proof. The proof is in Appendix E. \square

Theorem 5. (Average AoI in MAoIL) Let $AoI_{MAoIL}^{Ave}(T)$ denote the average AoI achieved by MAoIL algorithm in time duration $[0, T]$. We have $A_{MAoIL}^{Ave}(T) \leq \frac{T}{2K} + 2n$, where n is

the number of nodes in the line network and K is the number of update cycles during the time duration $[0, T]$.

Proof. The proof is in Appendix F. \square

V. SCHEDULING ALGORITHM FOR GENERAL NETWORK

In Line network topology, each node has its intended receiver. However, in general network topology, for a node v_i , all nodes in its neighbour set $NB(v_i)$ can be its relay. That is, there are multiple paths available for the transmission of its updates. Intuitively, we can build a spanning tree in the network and let the nodes transmit their updates using these fixed paths, then call MAoIL. However, such a strategy will suffer from poor scalability and robustness. When few nodes are added to the network, many nodes need to update their paths. If a node fails, which usually happens in WSNs, all nodes using the node as a relay cannot make updates. Moreover, in EH-WSNs, the path selection should be energy adaptive because if a node with poor energy condition is assigned a heavy transmission workload, the packets will suffer huge delay.

The scheduling algorithm proposed for the general network topology is called MAoIG. We first introduce how to select paths in an energy-adaptive way, then we show how to schedule nodes to achieve smaller AoI.

A. Path Selection

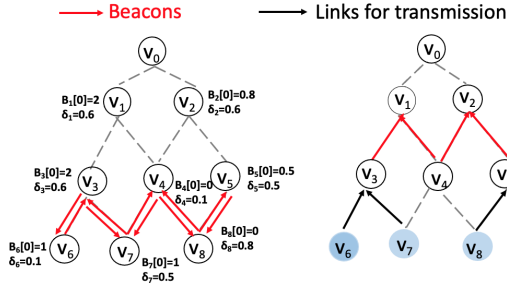


Fig. 4. The blue ones are the nodes who get their relaters. v_4 will not be selected as a relay due to its poor energy condition

In MAoIG, we divide the network into layers, a node v_i is in the r -th layer if the shortest path from it to sink is a r -hop path. Let L_r denote the nodes in the r -th layer, we have $\bigcup_{r=1}^R L_r = V$. For a node v_i , whose neighbours are $NB(v_i)$, if $v_i \in L_r$, all nodes in $NB(v_i) \cap L_{r-1}$ are v_i 's candidate relays. At the beginning of each cycle, each node chooses its relay. Specifically, we propose to consider a bottom-to-top manner, nodes in L_R choose their relays first. Each layer $r \in \{R, R-2, \dots, 1\}$ has a time window to broadcast beacons for selecting relays. During the time window, when nodes broadcasting their beacons, we used the Time-division multiple access (TDMA) channel access method to guarantee that each node can get its relay. Fig. 4 shows an example. The transmission tree is built from bottom to up.

After L_R, \dots, L_{r+1} finishing the path selection phase, each node in L_r has been assigned its relay workload and a sub-tree rooted in itself. For a node v_i , let $St(v_i, k)$ denote the sub-tree rooted at v_i in k -th cycle and wl_i^k be the number nodes in $St(v_i, k)$. The earliest time slot for v_i to finish its workload is

$t_{E2F}(v_i, k) = f_{TE}(v_i, t_s^k, es + wl_i^k \times (es + er))$. For example, in Fig. 4, workload for v_3 is 2.

When nodes in L_r , $r \in \{R, R-2, \dots, 1\}$ are selecting relays, the details are as follows:

- 1) Each node $v_i \in L_r$ broadcast a beacon containing its workload $[wl_i^k, t_{E2F}(v_i, k)]$.
- 2) After receiving beacons from nodes in L_r , each node $v_j \in L_{r-1}$ will broadcast a beacon containing $t_{E2F}(v_j, k) = f_{TE}(v_j, t_s^k, es + (wl_j + 1) \times (es + er))$ to nodes in $L_r \cap NB(v_j)$, where $t_{E2F}(v_j, k)$ is the earliest time slot it harvest enough energy for the added workload $(wl_j^k + 1)$ if it is selected to be the relay.
- 3) After receiving all feedback, each node $v_i \in L_r$ will select the node v_{j^*} which satisfies $v_{j^*} \in NB(v_i) \cap L_{r-1}$ and $t_{E2F}(v_{j^*}, k) \leq t_{E2F}(v_j, k), \forall v_j \in NB(v_i) \cap L_{r-1}$. Then v_i will tell v_{j^*} that it as been selected.
- 4) Nodes in L_{r-1} will update and their workload and its children set $St(v_j, k)$ in this cycle.

Lemma 4. For the Path Selection Phase in MAoIG, we have:

- 1) For each node, the time complexity of the scheduling algorithm is $O(\frac{n\Delta(e_s + e_r)}{\delta_{min}})$,
- 2) The transmission complexity is $O(n^2)$.

where Δ is the maximum degree of G , n is the number of nodes in the network. $e_s(e_r)$ are the energy cost for sending(receiving) unite data. δ_{min} is the minimum energy harvesting rate.

Proof. The proof is in Appendix G. \square

It is worth highlighting that, there are no updates that are transmitted in the path selection phase. After the construction stage, we get a transmission tree $\mathbf{T}(V, E_T, k)$.

In MAoIG, we also divide the network into subareas. A node v_i is in subarea m if its distance to sink $D(v_i, v_0)$ is satisfying $(m-1)d < D(v_i, v_0) \leq md$, where d is a constant whose value can be derived from Theorem 1. At time slot t , if t is odd (even), all Odd (Even) subareas will be active while Even (Odd) subareas will be idle.

In MAoIG, we treat each leaf-to-sink path in $\mathbf{T}(V, E_T, k)$ as a Line network where MAoIL can be used. The "Line" network will be scheduled one by one.

B. Performance Analysis

Lemma 5. In MAoIG, for each packet $p_i^k, i = 1, 2, \dots, n$, $delay_i^k = t_i^k - g_i^k$, where t_i^k is the time when p_i^k is received by the sink, we have

$$delay_i^k = 2 \times |\mathcal{P}_i|,$$

where $|\mathcal{P}_i|$ is the length of the path \mathcal{P}_i from v_i to sink.

Proof. The proof is almost the same as that in Lemma 3 so we omit it here. \square

Theorem 6. (Peak AoI in MAoIG) Let $AoI_{MAoIG}^{Peak}(T)$ denote the peak AoI achieved by MAoIG algorithm in time duration $[0, T]$, we have $AoI_{MAoIG}^{Peak}(T) \leq RG^{Max} + 2R$, where R is the depth of the BFS tree which we term the depth of the network and RG^{Max} is the time cost of the longest update cycle.

Algorithm 2: Minimising AoI Scheduling Algorithm for General Network (MAoIG)

```

1 Input: Network  $G(V, E)$ ; Energy cost  $e_s, e_r, e_c$ ; The
   length of subarea  $d$ ; The length of time  $T$ .
2 Output: A schedule set  $\mathcal{S}_G$ .
3 Step 0: Divide  $G$  into  $\{L_1, \dots, L_R\}$ ,  $t_s^L(k) = 0$ ;
4 while  $t_s^k \leq T$  do
5   Path Selection Phase:
6   for  $r = R : 1 : -1$  do
7     Step1: for  $v_i \in L_r$  do
8        $v_i$  broadcast its workload  $wl_i^k$  and the
       earliest time slot it can harvest enough
       energy for its workload  $t_{E2F}(v_i, k)$  to all
       nodes in  $NB(v_i) \cap L_{r-1}$ 
9     Step2: for  $v_j \in L_{r-1}$  do
10       $v_j$  broadcast the earliest time slot that it
      can harvest enough energy for the added
      workload( $wl_i^k + 1$ ) if it is selected to be
      the relay of  $v_i$ 
11     Step3: for  $v_i \in L_r$  do
12      After receiving feedback,  $v_i$  will select  $v_{j^*}$ 
      which satisfies  $v_{j^*} \in NB(v_i) \cap L_{r-1}$  and
       $t_{E2F}(v_{j^*}, k) \leq t_{E2F}(v_j, k), \forall v_j \in$ 
       $NB(v_i) \cap L_{r-1}$ . Then  $v_i$  will tell  $v_{j^*}$  that
      it as been selected.
13     Step4: for  $v_j \in L_{r-1}$ : do
14       $v_j$  updates its workload and its children set
      ( $St(v_j, k)$ ) in this cycle.
15   After this phase, we got transmission tree
       $\mathbf{T}(V, E_T, k)$ 
16   Update Transmission Phase:
17   for path  $\mathcal{P}_i \in \mathbf{T}(V, E_T, k)$  do
18     Regard  $\mathcal{P}_i$  as a Line network and call MAoIL

```

Proof. The proof is almost the same as that in Theorem 4 so we omit it here. \square

Theorem 7. (Ave AoI in MAoIG) Let $AoI_{MAoIG}^{Ave}(T)$ denote the Average AoI achieved by MAoIG algorithm in time duration $[0, T]$. Let K be the number of updating cycles during $[0, T]$. We have $A_{MAoIG}^{Ave}(T) \leq \frac{T}{2K} + 2R$, where R is the depth of the BFS tree and K is the number of update cycles during the time duration.

Proof. The proof is almost the same as that in Theorem 5 so we omit it here. \square

Indeed, different selected paths bring a different length of a cycle. A good transmission tree achieves a shorter length of a cycle. That is, the influence of Path selection present in RG^{Max} and $\frac{T}{K}$.

VI. NUMERICAL RESULTS

In this section, we present the numerical results to validate our proposed scheduling algorithm.

A. Simulation settings

We consider a general network with a sink and multiple nodes randomly generated in a geographical topology of dimensions $R \times R$. The location of the sink is $(0, 0)$ and the location of a node follows uniform distribution. We consider the Physical interference model in all of the experiments since it characterises the real wireless communication better than the protocol model. We generate the location of nodes randomly. The power consumed for receiving or sending a certain packet is chosen equal to $e_{x,1} = -17dBm = 0.05$ J/s, in accordance with the IEEE 802.15.4 transceiver called AT86RF231 [36], with $x \in \{s, r\}$. Let S be the size of one update (in kb), the energy consumption of sending or receiving a packet of size S is $e_{x,S} = \frac{S}{v_s} e_{x,1}$, where $v_s = 250kb/s$ is the velocity representing the number of packets to send per second. In all the simulations, we vary S . Hence, we use different e_s and e_r in our model.

In all the simulations, we set the initial battery level to $B_i[0] = 0$ for all $\{v_1, v_2, \dots, v_n\}$ and consider the following energy harvesting scenarios:

- 1) **Solar Power:** In a solar-powered EH-WSN, at time slot t , the energy harvesting rate of a node v_i can be computed by the following equation:

$$En_{Solar}(v_i, t) = P_{Solar}(v_i, t) \times S(v_i) \times \xi_{Solar}$$

where $P_{Solar}(v_i)$ is the density of solar power in the location of v_i , $S(v_i)$ is the size of v_i 's solar panel and ξ_{Solar} is the energy transfer efficiency. We consider typical solar-powered sensor nodes, each is equipped with a $3.8cm \times 9cm$ solar panel and the transfer efficiency is 50%.

For the density of solar power, considering that in real solar-powered EH-WSN, some nodes are under shade of trees and buildings while others are not. That is, the density of solar power is time-varying and at that moment some nodes may not even be part of the network. Therefore, we consider an average solar power density $P_{Solar}^{Ave} = 300$ W/m², which provided in [37], respecting the real data in 14:00 pm, January 1, 2021 in Eugene, Oregon. To simulate the heterogeneous solar power density, we use a random variable θ following the uniform distribution in $[0.5, 1.5]$. For a node v_i , at time slot t , its energy harvesting rate can be computed by

$$En_{Solar}(v_i, t) = P_{Solar}^{Ave} \times S(v_i) \times \xi_{Solar} \times \theta(v_i, t)$$

where $\theta(v_i, t)$ is the value of θ at time slot t .

- 2) **Radio Frequency signals:** We consider both dedicated RF Energy Source like TX91501 (Power:3W) and ambient RF Energy Source like TV tower (960kW) [38]. We set the location of the dedicated RF Source as the same as the sink. For the TV tower, we assume that it is 500m away from the sink. In ideal conditions for an RF-Powered node v_i , at time slot t , its harvested energy can be expressed as

$$En_{RF}(v_i, t) = \xi_{RF} \times P_{RF} \times dis(RF, v_i)^{-\alpha}$$

where $\xi_{RF} \in [0, 1]$ is the RF conversion efficiency usually equal to %60, P_{RF} is the transmit power of the RF source, $dis(RF, v_i)$ is the Euclidean distance

between v_i to RF source and $\alpha = 3$ is the path loss exponent.

In the real RF-Powered EH-WSN, the radio waves in the background will superimpose with the charging RF waves. Since the amplitude and phase of those noise waves are random, the superimposed waves received by each node are time-varying. To simulate this feature, we use a random variable θ following the uniform distribution in $[0.5, 1.5]$. For a node v_i , at time slot t , its energy harvesting rate is

$$En_{RF}(v_i, t) = \xi_{RF} \times P_{RF} \times dis(RF, v_i)^{-\alpha} \times \theta(v_i, t)$$

where $\theta(v_i, t)$ is the value of θ at time slot t .

Because this work is the first one addressing the AoI minimization problem in multi-hop EH-WSN, there are no existing algorithms in the literature. For comparison, we propose to compare our algorithm to baseline algorithms and to a lower bound on the peak AoI:

- 1) **Lower Bound on Peak AoI:** In EH-WSN, the lower bound on peak AoI can be derived by the transmission workload of nodes and their energy inputs including initial battery level and energy harvesting rates. Specifically, when the transmission tree has been built, the workload of each node can be determined. For each node v_i , we calculate how many time slots it needs to harvest enough energy for the first cycle of data collection. The maximal energy harvesting time among all ones is the lower bound on the peak AoI.
- 2) **Baseline Algorithms:** We compare MAoIG with other algorithms: For transmission scheduling. We use an algorithm called DCoSG, which was proposed for latency minimization scheduling problem in multi-hop EH-WSN [7], where nodes used a fixed tree in every cycle for transmission. The generation time of updates was not considered in [7]. Hence, we designed the generation in two ways and obtained two baseline algorithms: a) In the first baseline algorithm, we considered cycled updating policy, each node in the network generates a packet at the first time slot of each cycle and we called the algorithm as DCoSG-Cycled-Update. Specifically, in each cycle, the sink receives one and only one update from every node in the network. b) In the second baseline algorithm, we consider random updating policy and we called the algorithm as DCoSG-Random-Update.

B. Results

In Figs. 5-8, we have plotted the peak and average AoI for a $[20m \times 20m]$ network consisting of 20 nodes and with different packet sizes in kbits (kb) during a time duration $T_{max} = 1000s$.

1) **Adaptive Path Selection:** To evaluate the performance of our adaptive path selection strategy, we compare it with the following two baselines: a) FixedBFS+MAoIL: We construct a Breadth-First Search tree in the whole network and regard all leaf-to-sink path as Line network and use MAoIL in every cycle. b) RanAda+MAoIL: Here, each node selects a relay randomly in each cycle. The difference between RanAda+MAoIL and MAoIG is that MAoIG selects the relay with good energy conditions but RanAda+MAoIL is

random. In Fig .5, we can see that adaptive path selection algorithms are much better than the fixed one. Both peak AoI and average AoI reduce about 50%. When comparing MAoIG with RanAda+MAoIL, if we consider the energy issue, the result can be even better.

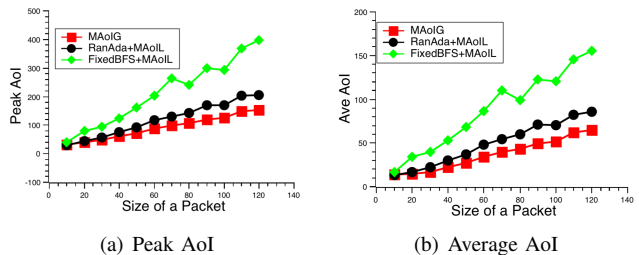


Fig. 5. The performance of adaptive path selection

2) **Packet Sizes:** In Figs. 6, 7 and 8, we have plotted the peak and average AoI versus the packet sizes for a solar source ($300W/m^2$), an ambient RF source from TV Tower transmitting with power 960 kWatts located at $[-500m, -500m]$ and a dedicated RF source transmitting at power 3 Watts, respectively.

We compare the peak AoI performance to its lower bound and to the DCoSG-Cycled-Update algorithm and we compare the average AoI to the DCoSG-Cycled-Update and DCoSG-Random-Update algorithms. Note that for all peak AoI figures, we have purposefully omitted the DCoSG-Random-Update algorithm because its performance is dramatically worse than the others and makes it hard to have clues on how our proposed algorithm is performing compared to the lower bound and the DCoSG-Cycled-Update algorithm. Figs. 6(a), 7(a) and 8(a) show the performance of peak AoI as a function of different packet sizes in three energy harvesting scenarios, we can see that:

- Firstly, in all of these figures, our proposed algorithm achieves a better peak AoI than DCoSG-Cycled-Update. This is because DCoSG-Cycled-Update makes updates at the first time slot but these packets cannot be transmitted timely since nodes do not have enough energy or channel available for transmissions. So packets are getting stuck in buffers and become stale. However, in MAoIG, nodes send an update only if all nodes on their path have enough energy so that delays due to energy scarcity can be avoided. On the other hand, in DCoSG-Cycled-Update, each node uses fixed relays in each cycle while MAoIG selects nodes with good energy as relays in each cycle.
- Secondly, the Peak AoI achieved by MAoIG is very close to the lower bound. The reason is that in EH-WSN, the energy delay is always much larger than the collision delay. In MAoIG, all energy delays are avoided, which makes it really close to the lower bound. On the other hand, in Fig .7(a), there is a gap between the lower bound and MAoIG for the ambient RF source. Because in the TV Tower scenario, nodes are closer to the high power RF source, their energy harvesting rates are the highest among the three scenarios. In Fig .7(a), the delay is mainly caused by interference, which is not counted in the Lower Bound.

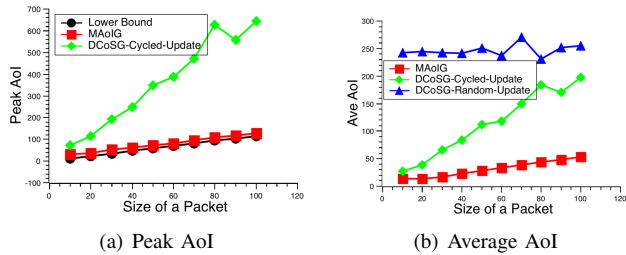


Fig. 6. Solar Power ($300W/m^2$) for different packet sizes (in kbs).

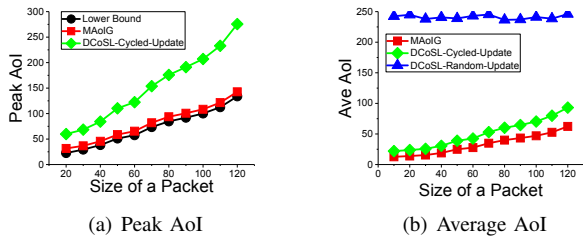


Fig. 7. Ambient RF source from TV Tower transmitting with power 960 kWatts located at $[-500m, -500m]$ for different packet sizes (in kbs).

Additionally, Figs. 6(b), 7(b) and 8(b) show the performance of average AoI as a function of different packet sizes in three energy harvesting scenarios. We can see:

- MAoIG outperforms both the DCoSG-Cycled-Update and DCoSG-Random-Update. Specifically, when the available energy is weak, like in Fig. 8(b), the gap between DCoSG-Cycled-Update and MAoIG increases with the growth of packet size. Due to poor energy conditions, it takes more time for the nodes to harvest enough energy to finish their workload and the delay caused by that rises in DCoSG-Cycled-Update.
- The DCoSG-Random-Update is always the worst one because here the nodes make updates randomly. The results also show that cycled scheduling is better than the random one.

3) *Network Scalability*: In Figs. 9-11, we have plotted the peak and average AoI versus the number of nodes for a solar source, an ambient RF source from TV Tower and a dedicated RF source, respectively. We consider a $[30m \times 30m]$ network topology and we fix the size of one packet to be 20 kb. In these figures, $T_{max} = 2000s$. We also omit the DCoSG-Random-Update algorithm in all peak AoI Figures.

Figs. 9(a), 10(a) and 11(a) show the performance of the peak AoI as a function of different numbers of nodes in three energy harvesting scenarios. We can see that:

- MAoIG outperforms DCoSG-Cycled-Update in all scenarios. In Figs. 9(a) and 10(a), as the number of nodes grows, the workload of nodes increases, especially the nodes that are closer to the sink. Here, the bottleneck nodes which take the longest time to finish their workload are the one-hop neighbours of the sink. It takes more time for those nodes to harvest enough energy. Thus, the gap between MAoIG and DCoSG-Cycled-Update increases as the number of nodes increases.
- Comparing MAoIG and the lower bound on peak AoI, we can see that the gap between them also grows in Figs.

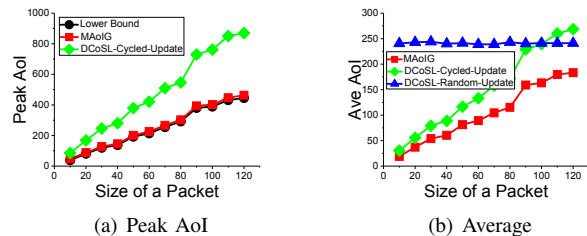


Fig. 8. Dedicated RF Source transmitting at power 3 Watts for different packet sizes (in kbs).

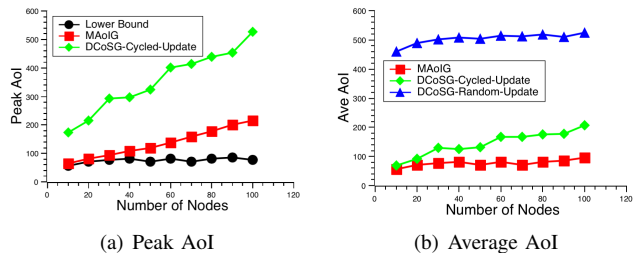


Fig. 9. Solar Power ($300W/m^2$) for different packet sizes (in kbs).

9(a) and 10(a). The lower bound is derived only from the energy workload, the interference between nodes is not considered. In a network with a fixed size, when the number of nodes increases, the delay caused by interference increases but this issue has not been taken into consideration in the Lower Bound.

- In 11(a), the peak AoI of MAoIG does not increase with the number of nodes. Different from Figs. 9(a) and 10(a), the bottleneck nodes which take the longest time to finish their workload are the ones far from the sink as the location of the RF source is the same with the sink. In RF-powered scenario, for a node v_i , its energy harvesting rate $En(v_i) = \xi_{RF}P(\text{source}) \times \text{dis}(\text{Source}, v_i)^{-\alpha}$, where $\alpha = 3$. The energy harvesting rate drops remarkably when the distance increases. Even though nodes close to the sink have higher workloads, they can finish it faster than those far from the sink. When the number of nodes grows, the workload of those nodes will not change much due to our adaptive path selecting method in MAoIG. Therefore, the AoI does not increase when the number of nodes rises.

Figs. 9(b), 10(b) and 11(b) show the performance of the average AoI in the function of different numbers of nodes in three energy harvesting scenarios. We can see that MAoIG outperforms both the DCoSG-Cycled-Update and DCoSG-Random-Update algorithms. When we compare Figs. 9(b) and 10(b), we can see that when the environment is energy scarce (compared with the size of one packet), the benefit of our algorithm is more visible.

4) *Upper and Lower Bounds*: Last but not least, we compare our theoretical upper/lower bounds and experimental results in terms of peak/average AoI versus different packet sizes and numbers of nodes in Fig. 12. Here we consider the solar-powered scenario. In Fig. 12(a), our theoretical bounds are very close to the experimental ones in all scenarios. In Fig. 12(b), we show the gap between theoretical bounds and experimental result for different numbers of nodes in the

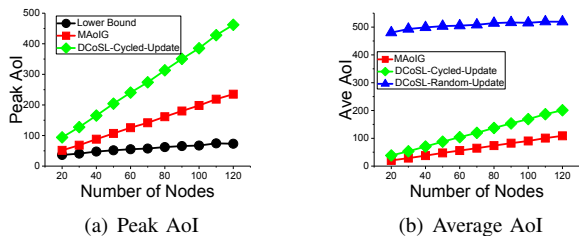


Fig. 10. Ambient RF source from TV Tower transmitting with power 960 kWatts located at $[-500m, -500m]$ for different number of nodes.

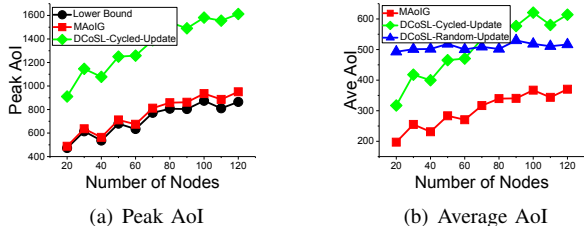


Fig. 11. Dedicated RF Source transmitting at power 3 Watts for different number of nodes.

network. We can see that the upper bounds are tight in all cases. However, the accuracy of the lower bound of peak AoI is affected by the number of nodes, as we have seen in Fig. 9(a). The reason is that the lower bound is derived only from the energy workload and the interference between nodes is not taken into account.

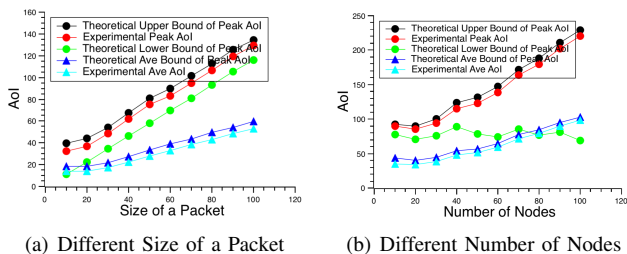


Fig. 12. Theoretical Upper/Lower Bounds and Experimental Result

VII. CONCLUSION AND DISCUSSION:

In this paper, we studied the peak/average AoI minimization problem in multi-hop EH-WSN. We proposed scheduling algorithms for both line network topology and general network topology. For general network topology, we proposed a distributed scheduling algorithm called MAoIG. We proved that: 1) The peak AoI achieved by MAoIG verifies $AoI_{MAoIG}^{Peak}(T) \leq RG^{Max} + 2R$, where R is the depth of the BFS tree of the network and RG^{Max} is the time cost of the longest updating cycle. 2) The average AoI achieved by MAoIG verifies $AoI_{MAoIG}^{Ave}(T) \leq \frac{T}{2K} + 2R$, where K is the number of update cycles during the time duration.

We made a numerical simulation for three kinds of energy harvesting WSN: solar-powered, dedicated RF-powered and TV Tower RF-powered. The numerical results validate that MAoIG outperforms all of the baseline algorithms in all scenarios. The results also show that when the environment

is energy scarce (compared with the size of one packet), the benefit of our algorithm is more visible.

Possible extensions of our proposed algorithm MAoIG are discussed below which can be directly applied or studied in future works.

- 1) Heterogeneous packet size: In event-driven monitoring systems, the sampling rate is determined by the application /phenomena itself and can be heterogeneous among target areas. As an update packet contains all measurements collected by the node since the last update, the update packets here are in different sizes while in our algorithm MAoIG we assume that all packets are of equal size. In addition, measurements can be compressed with the compression ratio depending on the algorithm and the sense data. In this case, for generation time and transmission schedule, we need to adjust the energy cost for sending or receiving a packet according to the size of the packet, then our proposed algorithm MAoIG is effective. The case of the size of a packet being larger than the buffer size of a node is another interesting and important problem that is worth to be studied in real systems.
- 2) Energy-neutral WSN: In Energy-Neutral EH-WSN systems, the goal is to optimize energy efficiency while guaranteeing that none of the EH-node runs out of its energy [4]. Here, the strict cycled update is not feasible. In each update term, only a subset of nodes can make update due to the energy issue. The selection of nodes will substantially affect or impact on both peak AoI and Average AoI. In the selection, for peak AoI, we need to consider the latest update time of each node; for average AoI, the optimal update frequency assignment among nodes. AoI minimization in Energy-neutral WSN is also an interesting and challenging problem, which needs further study for future work.
- 3) Heterogeneous target areas: For the applications where some target areas are more important than others, such nodes need more frequent sampling. In this case, we will consider minimizing the Weighted-Ave or Weighted-Peak AoI minimization problem and this needs further study in future work.
- 4) Power allocation: In this paper, we focus on time scheduling and assume that the transmission power of a node is fixed. Combining the power allocation and time scheduling in the AoI minimization problem is also interesting and challenging.
- 5) Heterogeneous link quality. In a real WSN, the quality of the link is a function of the distance between two nodes, the background noise, the transmission power, the propagation model, etc. In this scenario, we need to consider issues like packet re-transmission and modify our algorithm.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China under Grant NO.61632010 and NO.U19A2059. It is also supported by EPSRC Science of Sensor System Software (S4) under grant EP/N007565/1.

REFERENCES

- [1] D. K. Sah and T. Amgoth, "Renewable energy harvesting schemes in wireless sensor networks: a survey," *Information Fusion*, vol. 63, pp. 223–247, 2020.
- [2] Y. Zhang, S. He, and J. Chen, "Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1632–1646, 2015.
- [3] S. Yang, Y. Tahir, P. Y. Chen, A. Marshall, and J. McCann, "Distributed optimization in energy harvesting sensor networks with dynamic in-network data processing," in *IEEE INFOCOM 2016*, 2016, pp. 1–9.
- [4] S. Yang and J. A. McCann, "Distributed optimal lexicographic max-min rate allocation in solar-powered wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 1, pp. 1–35, 2014.
- [5] K. Chen, H. Gao, Z. Cai, Q. Chen, and J. Li, "Distributed energy-adaptive aggregation scheduling with coverage guarantee for battery-free wireless sensor networks," in *IEEE INFOCOM*, 2019.
- [6] C. Quan, G. Hong, C. Zhipeng, C. Lianglun, and L. Jianzhong, "Energy-collision aware date aggregation scheduling for energy harvesting sensor network," in *IEEE INFOCOM*, 2018.
- [7] T. Zhu, J. Li, H. Gao, and Y. Li, "Latency-efficient data collection scheduling in battery-free wireless sensor networks," *ACM Trans. Sens. Networks*, vol. 16, no. 3, pp. 25:1–25:21, 2020.
- [8] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.
- [9] S. Farazi, A. G. Klein, and D. R. Brown, "Average age of information for status update systems with an energy harvesting server," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2018, Honolulu, HI, USA, April 15-19, 2018*. IEEE, 2018, pp. 112–117.
- [10] A. Arafa, J. Yang, S. Ulukus, and H. V. Poor, "Age-minimal online policies for energy harvesting sensors with incremental battery recharges," in *2018 Information Theory and Applications Workshop, ITA 2018, San Diego, CA, USA, February 11-16, 2018*. IEEE, 2018, pp. 1–10.
- [11] B. T. Bacinoglu and E. Uysal-Biyikoglu, "Scheduling status updates to minimize age of information with an energy harvesting sensor," in *2017 IEEE International Symposium on Information Theory, ISIT 2017, Aachen, Germany, June 25-30, 2017*. IEEE, 2017, pp. 1122–1126.
- [12] C. R. Lorenzo Corneo and P. Gunningberg, "Age of information-aware scheduling for timely and scalable internet of things applications," in *IEEE INFOCOM*, 2019.
- [13] T. Zhu, J. Li, H. Gao, Y. Li, and Z. Cai, "Aoi minimization data collection scheduling for battery-free wireless sensor networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.
- [14] P. Turchi, "Prospects for pulsed power above ten megamperes," in *Digest of Technical Papers. 12th IEEE International Pulsed Power Conference. (Cat. No.99CH36358)*, vol. 1, 1999, pp. 3–7 vol.1.
- [15] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *INFOCOM, IEEE*, 2012, pp. 2731–2735.
- [16] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1807–1827, 2019.
- [17] S. Yin, E. Uysal-Biyikoglu, R. Yates, C. E. Koksall, and N. B. Shroff, "Update or wait: How to keep your data fresh," in *IEEE INFOCOM -the IEEE International Conference on Computer Communications*, 2016.
- [18] M. Costa, M. Codreanu, and A. Ephremides, "Age of information with packet management," in *2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, June 29 - July 4, 2014*. IEEE, 2014, pp. 1583–1587.
- [19] —, "On the age of information in status update systems with packet management," *IEEE Trans. Inf. Theory*, vol. 62, no. 4, pp. 1897–1910, 2016.
- [20] Q. He, D. Yuan, and A. Ephremides, "Optimal link scheduling for age minimization in wireless systems," *IEEE Transactions on Information Theory*, vol. 64, no. 7, pp. 5381–5394, 2017.
- [21] R. Talak, S. Karaman, and E. Modiano, "Minimizing age-of-information in multi-hop wireless networks," in *55th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2017, Monticello, IL, USA, October 3-6, 2017*. IEEE, 2017, pp. 486–493.
- [22] Q. Liu, H. Zeng, and M. Chen, "Minimizing age-of-information with throughput requirements in multi-path network communication," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Mobihoc 2019, Catania, Italy, July 2-5, 2019*. ACM, 2019, pp. 41–50.
- [23] S. Farazi, A. G. Klein, and D. R. Brown, "Fundamental bounds on the age of information in general multi-hop interference networks," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2019, Paris, France, April 29 - May 2, 2019*. IEEE, 2019, pp. 96–101.
- [24] A. M. Bedewy, Y. Sun, and N. B. Shroff, "The age of information in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1248–1257, 2019.
- [25] Q. Chen, H. Gao, S. Cheng, J. Li, and Z. Cai, "Distributed non-structure based data aggregation for duty-cycle wireless sensor networks," in *INFOCOM 2017*, 2017, pp. 1–9.
- [26] F. Benkhelifa, Z. Qin, and J. A. McCann, "Minimum throughput maximization in lora networks powered by ambient energy harvesting," in *2019 IEEE International Conference on Communications, ICC 2019, Shanghai, China, May 20-24, 2019*. IEEE, 2019, pp. 1–7.
- [27] F. Benkhelifa and J. A. McCann, "Resource allocation for noma-based lpwa networks powered by energy harvesting," *arXiv preprint arXiv:2012.14834*, 2020.
- [28] K. Geissdoerfer and M. Zimmerling, "Bootstrapping battery-free wireless networks: Efficient neighbor discovery and synchronization in the face of intermittency," in *18th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2021, April 12-14, 2021*, J. Mickens and R. Teixeira, Eds. USENIX Association, 2021, pp. 439–455.
- [29] E. Boshkovska, D. Ng, N. Zlatanov, and R. Schober, "Practical non-linear energy harvesting model and resource allocation for swipt systems," *IEEE Communications Letters*, vol. 19, no. 12, pp. 2082–2085, 2015.
- [30] G. Jackson, S. Kartakis, and J. McCann, "Accurate models of energy harvesting for smart environments," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2017, pp. 1–7.
- [31] A. Cammarano, C. Petrioli, and D. Spenza, "Pro-energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks," in *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, 2012, pp. 75–83.
- [32] Y. Shi, Y. T. Hou, J. Liu, and S. Kompella, "How to correctly use the protocol interference model for multi-hop wireless networks," in *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2009, New Orleans, LA, USA, May 18-21, 2009*, E. W. Knightly, C. Chiasserini, and X. Lin, Eds. ACM, 2009, pp. 239–248.
- [33] P. J. Wan, C. H. Huang, L. Wang, Z. Wan, and X. Jia, "Minimum-latency aggregation scheduling in multihop wireless networks," in *Tenth ACM International Symposium on Mobile Ad Hoc Networking & Computing, Mobihoc, 2009*.
- [34] G. Li, J. Li, and B. Yu, "Lower bound of weighted fairness guaranteed congestion control protocol for wsns," in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 3046–3050.
- [35] R. Sappidi, A. Girard, and C. Rosenberg, "Maximum achievable throughput in a wireless sensor network using in-network computation for statistical functions," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1581–1594, 2012.
- [36] <https://www.microchip.com>.
- [37] <http://solarlat.uoregon.edu/DataPlottingProgram.html>.
- [38] X. Lu, P. Wang, D. I. Kim, and Z. Han, "Wireless networks with RF energy harvesting: A contemporary survey," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 2, pp. 757–789, 2015.



Kunyi Chen Dr. Kunyi Chen is an algorithm engineer in Alibaba Group. She received her PhD degree in July 2022 from Department of Computing, Harbin Institute of Technology. She was a visiting PhD student in Department of Computing, Imperial College London, UK, since November 2020 to December 2021. She received her BS in 2016 from Department of Mathematics, Harbin Institute of Technology, China. Her research interests include Scheduling and in-network computing in Energy-Harvesting Wireless Sensor Network



Fatma Benkhelifa Dr. Fatma is a research associate in the AESE research lab at Imperial College London since 2018. She obtained her PhD in Electrical Engineering in 2017 from King Abdullah University of Science and Technology (KAUST), Saudi Arabia. She also obtained her Master of Science from KAUST in January 2013. She graduated as a Polytechnician engineer from “Ecole Polytechnique de Tunis” with major in Signals and Systems. Her research interests include, but not limited to, the resource management algorithms, stochastic

geometry-based analysis, spatiotemporal modelling, energy harvesting, physical layer security, reinforcement learning and deep learning approaches of low power wide area networks, wireless sensor networks, and wireless communication systems.



Hong Gao received the BS degree in computer science from Heilongjiang University, Harbin, China, the MS degree from Harbin Engineering University, Harbin, China, and the PhD degree in computer science from the Harbin Institute of Technology, Harbin, China. She is currently a professor with the School of Computer Science and Technology, Harbin Institute of Technology. Her research interests include graph data management, sensor network, and massive data management. She is the winner of National Science and Technology Progress Award.



Julie A. McCann (M'16) is a Professor in Computer Systems with Imperial College London. Her research centres on decentralized and self-organizing schemes for spatial computing e.g., Wireless Sensor systems, Internet of Things, or Cyber-physical systems. She leads the Adaptive Embedded Systems Engineering Research (AESE) Lab, is Deputy Director for the UK-wide PeTraS Centre for IoT Cyber-security, and until recently co-directed the Intel Collaborative Research Institute for Sustainable Cities. She has received significant funding through national and

international bodies such as the UK's EPSRC, EU FP7/H2020 funding and Singapore NRF; she has a sub-lab in Singapore with I2R and HDB. Prof McCann is an Elected Peer for the EPSRC, serves on/chairs/AE for the top international conference committees and journals in the field, and is a Fellow of the BCS and Chartered Engineer.



Jianzhong Li is currently a professor and doctoral supervisor with the School of Computer Science and Technology, Harbin Institute of Technology, China. He worked with the Department of Computer Science, Lawrence Berkeley National Laboratory, as a scientist, from 1986 to 1987 and from 1992 to 1993. He was also a visiting professor with the University of Minnesota at Minneapolis, Minnesota, from 1991 to 1992 and from 1998 to 1999. His research interests include massive data computing and wireless sensor networks. He has published

more than 200 papers in refereed journals and conference proceedings, such as the VLDB Journal, the Algorithmica, the IEEE Transactions on Knowledge and Data Engineering, the IEEE Transactions on Parallel and Distributed Systems, SIGMOD, SIGKDD, VLDB, ICDE, and INFOCOM. His papers have been cited more than 15,000 times. He has been involved in the program committees of major computer science and technology conferences including SIGMOD, VLDB, ICDE, INFOCOM, ICDCS, and WWW. He also served on the editorial boards for distinguished journals such as the IEEE Transactions on Knowledge and Data Engineering.

APPENDIX A PROOF OF THEOREM 1

We prove that EH-PAMS problem is NP-Hard by using polynomial-time Turing reduction¹. To do so we need to prove the following steps: 1) Firstly, we consider the peak AoI minimization scheduling problem in BP-WSN, namely the BP-MPAS problem, which is a special case of EH-PAMS problem by setting initial battery levels to its maximum value $B_i[0] = B_{Max}$ and energy harvesting rates to zero $\delta_i[t] = 0$ for all $i \in \{1, \dots, n\}, t \in \{0, \dots, T\}$. If BP-MPAS is NP-hard, for the more general problem, EH-PAMS is also NP-Hard. 2) Secondly, we used the method of reduction to absurdity to prove that BP-MPAS is NP-Hard. Specifically, we prove that if there exists a polynomial-time algorithm \mathcal{A} for BP-MPAS problem, the optimal solution of the minimum latency collection schedule Problem in BP-WSN (namely BP-MLDCS problem), which is NP-hard, can be obtained in polynomial time by calling \mathcal{A} . Due to the limited space, the details are omitted.

APPENDIX B PROOF OF LEMMA 2

Based on the estimated energy harvesting rate and equation (1), each node maintains a vector recording the battery level at every single time slot during the current updating cycle. When the energy cost/price E is given, it takes at most $\frac{E}{\delta_{min}}$ basic operations for a node v_i to find the minimum t satisfying $B_i(t) \geq E, t \geq t_{start}$.

APPENDIX C PROOF OF THEOREM 3

There are two phases in MAoIL: the Update Time Assigning Phase and Transmission Phase. Each of them contains two steps:

- 1) In the Step 1 of Update Time Assigning Phase: 1. For a node v_i , it need to compute $n - i$ times of Time-for-Harvesting-Energy function. According to Lemma 2, the complexity of Time-for-Harvesting-Energy function here is $O(\frac{n(e_s + e_r)}{\delta_{min}})$. Thus, the total time complexity is $O(\frac{n^2(e_s + e_r)}{\delta_{min}})$. 2. The number of beacons is $n - i$.
- 2) In the Step 2 of Update time Assigning Phase: 1. For a node v_i , it need to compute a Time-for-Harvesting-Energy vector and the length of such a vector is at most n . Thus, the total time complexity is $O(\frac{n^2(e_s + e_r)}{\delta_{min}})$. 2. There are at most n beacons need to be sent.
- 3) In the Step 1 of Transmission Phase: 1. The time complexity is $O(\frac{e_s + e_r}{\delta_{min}})$. 2. There are 3 beacons.
- 4) In the Step 2 of Transmission Phase: 1. There is no computation. 2. There are at most n beacons are sent.

To sum up, the total time complexity is $O(\frac{n^2(es + er)}{\delta_{min}})$ and the transmission complexity is $O(n^2)$.

APPENDIX D
PROOF OF LEMMA 3

There are two reasons for a packet being delayed during its travel to the sink: 1) Energy Delay: The current battery level of the node or its receiver is not sufficient for it to transmit all packets in its buffer timely. 2) Interference Delay: The sender/receiver cannot send/receive the packet since there is another packet being transmitted in its interference area.

- 1) For the Energy Delay: In the **Update Time Assigning Phase**, we have $g_i^k \geq t_s^L(k) + t_{energy}(p_i^k) = t_s^L(k) + \text{Max}(V_{TE}(v_i, t_s^L(k)))$. According to **Definition 8** and **Definition 9**, we can derive that by the time $t = g_i^k$, all nodes in \mathcal{P}_i has enough energy for the transmission of $\{p_1^k, \dots, p_i^k\}$. Therefore, the energy delay for p_i^k is zero.
- 2) For the Interference Delay:
 - a) Interference Delay among nodes in different subareas. According to **Theorem 2**, if multiple subareas are activated at a same time slot, nodes in these subareas are collision-free with each other.
 - b) Interference Delay among nodes in the same subarea. In the **Update Time Assigning Phase**, we have $g_i^k \geq t_{collision}(p_i^k) = t_{LSA}(p_{i-1}^k) + 1$, where $t_{LSA}(p_{i-1}^k)$ is the time that p_{i-1}^k being sent out from its subarea. That is, the packet p_i^k will be generated after p_{i-1}^k has been sent out of the subarea. Thus, in one time slot, there is at most one packet being transmitted in one subarea. If all of the subareas are always active, we have $delay_i^k = hop_i$. However, a subarea will be activated in every other time slot, we have $delay_i^k = 2 \times hop_i$.

Therefore, we have $delay_i^k = 2 \times hop_i$.

APPENDIX E
PROOF OF THEOREM 4

$AoI_{MAoIL}^{Peak}(T) = \max_{i,t} \{A_i(t)\} = A_{i^*}(t^*)$. Since we have $[0, T] = \cup_{k=1, \dots, K} [t_s^L(k), t_e^L(k)]$, $t^* \in [0, T]$, there must exist k^* satisfying $t^* \in [t_s^L(k^*), t_e^L(k^*)]$. Since v_{i^*} will make one update during $[t_s^L(k^*), t_e^L(k^*)]$, we have $A_{i^*}(t^*) \leq RL^L(k^*) + delay_{i^*}^{k^*}$. (The worst case happens when the update is made at time slot $t_s^L(k^*)$, then $A_{i^*}(t^*) = RL^L(k^*) + delay_{i^*}^{k^*}$). According to lemma 3, we have $delay_{i^*}^{k^*} \leq 2 \times hop_{i^*} \leq 2n$. Thus, $A_{i^*}(t^*) \leq RL^L(k^*) + 2n$ and we have $AoI_{MAoIL}^{Peak}(T) \leq RL^{Max} + 2n$.

APPENDIX F
PROOF OF THEOREM 5

$A_{MAoIL}^{Ave}(T) = \frac{1}{n} \sum_{i=1}^n \bar{A}_i(T)$, where $\bar{A}_i(T)$ is the average AoI of node v_i during $[0, T]$. Consider the AoI curve of v_i in the time duration $[0, T]$, e.g. Fig .2. There are at most $K + 1$ packets generated by v_i and received by the sink. We divide $[0, T]$ into $[0, t_i^1] \cup (t_i^1, t_i^2] \cup \dots \cup (t_i^{K+1}, T]$, where t_i^k is the arrival time of the k -th packet generated by v_i and received by the sink. The AoI curve can be divided into one triangle (during $[0, t_i^1]$) and K right-angled trapezoid(during $(t_i^k, t_i^{k+1}]$). We denote Δ_i as the area of the triangle and $\Theta_i(k)$ to be the area of k -th right-angled trapezoid. $\bar{A}_i(T) =$

$\frac{1}{K+1} \sum_{k=1}^K \bar{A}_i((t_i^k, t_i^{k+1}])$, where $\bar{A}_i((t_i^k, t_i^{k+1}])$ is the average AoI of v_i during $[t_i^k, t_i^{k+1}]$ for $\forall k \in \{1, 2, \dots, K + 1\}$.

- 1) The area of the triangle satisfies $\Delta_i = \frac{1}{2} H_i(1)^2$, where $H_i(1) = t_i^1 - 0$. Therefore, $\bar{A}_i([0, t_i^1]) = \Delta_i / H_i(1) = \frac{1}{2} H_i(1)$.
- 2) For the k -th right-angled trapezoid, let $Up_i(k)$ and $Lp_i(k)$ denote the upper line and Lower line of the trapezoid, and $H_i(k)$ denotes its height, i.e. $H_i(k) = t_i^{k+1} - t_i^k$. Its area is $\Theta_i(k) = \frac{1}{2} H_i(k) \times (Up_i(k) + Lp_i(k))$. In MAoIL, we have:
 - a) $Up_i(k)$ is the transmission delay in $k - 1$ -th cycle. According to Lemma 3, for all k , we have $Up_i(k) = 2 \times hop_i$;
 - b) According to Lemma 1, we have $Lp_i(k) = Up_i(k) + H_i(k) = 2 \times hop_i + H_i(k)$.
 - c) Based on a)-b), we have $\Theta_i(k) = 2H_i(k) \times hop_i + \frac{1}{2} H_i(k)^2$.

According to a)-c), we have $\bar{A}_i((t_i^k, t_i^{k+1}]) = 2 \times hop_i + \frac{1}{2} H_i(k)$

- 3) According to 1)-2), we have

$$\bar{A}_i(T) = \frac{1}{K+1} \left(\sum_{k=1}^{K+1} \frac{1}{2} H_i(k) + 2 \times hop_i \times K \right) \quad (18)$$

Since $\sum_{k=1}^{K+1} H_k = T$, we have $\bar{A}_i(T) \leq \frac{T}{2(K+1)} + 2 \times hop_i$

- 4) $\forall i \in \{1, 2, \dots, n\}$, we have $hop_i \leq n$, and v_i makes at least K updates during $[0, T]$, thus, $\bar{A}_i(T) \leq \frac{T}{2K} + 2 \times n$. Therefore, $A_{MAoIL}^{Ave}(T) \leq \frac{T}{2K} + 2n$

APPENDIX G
PROOF OF LEMMA 4

- 1) For each node v_i , it need to compute $t_{E2F}(v_i, k) = f_{TE}(v_i, t_s^k, e_s + (wl_i + 1) \times (e_s + e_r))$, which is a Time-for-Harvesting-Energy Function and wl_i is the number of nodes in the sub-tree rooted at v_i . Obviously, $wl_i < n$. According to Lemma 2, the time complexity of computing a Time-for-Harvesting-Energy Function is $O(\frac{E}{\delta_{min}})$, where E is the energy overhead. Therefore, the time complexity to compute one $t_{E2F}(v_i, k)$ is $O(\frac{n(e_s + e_r)}{\delta_{min}})$. On the other hand, v_i have at most $\Delta - 1$ candidate children, which means it need to compute at most $\Delta - 1$ times $t_{E2F}(v_i, k)$. Thus, the total complexity of the local scheduling is $O(\frac{n\Delta(e_s + e_r)}{\delta_{min}})$.
- 2) When selecting a parent for a node v_i , v_i need to send out 3 beacons to each of its candidate parents. For each node v_i , it has at most $\Delta < n$ candidate parents. Thus, a node needs to send out $3n$ beacons at most. There are n nodes in the network, thus, the transmission complexity is $O(n^2)$.