2023 Design, Automation & Test in Europe Conference (DATE 2023)

Data Freshness Optimization on Networked Intermittent Systems

Hao-Jan Huang*, Wen Sheng Lim*†, Chia-Heng Tu*, Chun-Feng Wu§, Yuan-Hao Chang†
*Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan
†Institute of Information Science, Academia Sinica, Taipei, Taiwan
§Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
{NE6081030, chiaheng}@ncku.edu.tw {tundergod1882, johnson}@iis.sinica.edu.tw, {cfwu417}@cs.nycu.edu.tw

***Abstract*—A networked intermittent system (NIS) is often deployed in the field for environmental monitoring, where sink nodes are responsible for relaying the data captured by sensors to a central system. To evaluate the quality of the captured monitoring data, Age of Information (AoI) is adopted to quantify the freshness of the data received by the central server. As the sink nodes are powered by ambient energy sources (e.g., solar and wind), the energy-efficient design of the sink nodes is crucial in order to improve the system-wide AoI. This work proposes the energy-efficient sink node design to save energy and extend system uptime. We devise an AoI-aware data forwarding algorithm based on the branch-and-bound (B&B) paradigm for deriving the optimal solution offline. In addition, an AoI-aware data forwarding algorithm is developed to approximate the optimal solution during runtime. The experimental results show that our solution can greatly improve the average data freshness for 148% against existing well-known strategies and achieves 91% performance of the optimal solution. Compared with the state-of-the-art algorithm, our energy-efficient design can deliver better A^3oI results by up to 9.6%.**

***Index Terms*—Data freshness, Age of Information (AoI), networked intermittent systems (NESs), energy harvesting (EH)**

I. Introduction

Environmental monitoring tracks real-time physical conditions for a specific purpose, such as wildlife tracking, disaster monitoring, and structural health monitoring. Usually, sensor nodes are deployed in the field for monitoring and transmitting the collected environmental data to a central controller so as to reflect the real-time status of the target environment. In order to fulfill the demand for a longer lifetime and low maintenance overhead, these sensors are often powered by ambient energies (e.g., solar, wind, and radio-frequency) without batteries, and they are also known as *networked intermittent systems* (NISes) [1], [2] owing to the intermittent execution behaviors caused by the unstable energy sources.

A common system hierarchy of NISes is illustrated in Fig. 1, where there is a sink node (K) sets between the sensors (S) and the controller (C) to relay the collected data from K to C. The hierarchical structure offers the flexibility of the system extensibility; that is, it can be extended by duplicating S and K. On such systems, the *data freshness* (i.e., the end-to-end latency of the data collected by S being received by C) becomes an important issue since S and K are powered by harvesting the unstable energy sources, where energy harvesting rates and energy consumptions are different across different application/system combinations. Without proper considerations, the controller may, in the worst-case scenario, not be able to receive any status update when S and K are down at the same time (because of insufficient harvested energy).

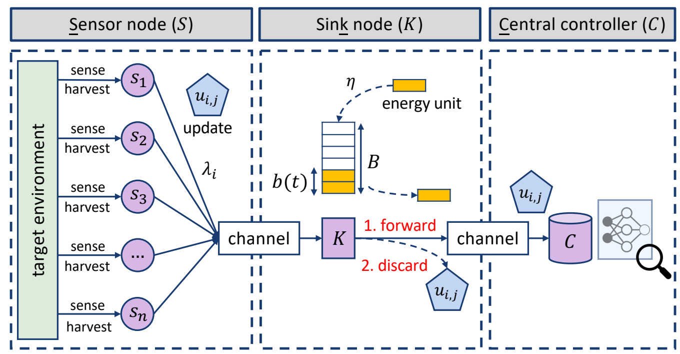

Fig. 1. System model overview.

Research works have been done to optimize the data freshness from the perspective of sensor nodes [3], [4], [5]. For example, Yates et al. [3] propose an online strategy for the sensor nodes to decide the update timing (data forwarding from S to C) under the limited energy arrival rates with different energy buffer sizes (i.e., finite energy, infinite energy, and one unit of energy). Bacinoglu et al. [4] develop both online and offline solutions for sensor nodes with an infinite capacity of energy buffer. Wu et al. [5] considers the long-term averaged data freshness of a single update source (i.e., a sensor node) and shows the optimality when the status updates are sent over uniformly-spaced time intervals.

In order to cope with a larger NIS with massive sensor nodes, the recent research trend is shifting to sink nodes as each sink node can help forward the data updates from many sensor nodes, and the data forwarding strategy on the node is critical to ensure data freshness [6], [7]. Especially, the forwarding algorithms on a sink node have been developed to provide fresh data by considering the characteristics of the energy harvesting rate and the energy consumption of the sink node, which are often referred to as the *energy causality* constraints. Nevertheless, the solutions are based on strong assumptions about the target system models. That is, Zhou et al. [6] assume that a sink node is responsible for the updates from a single sensor node, and they further study a large data buffer on the sink node to buffer updates sent by multiple sensors [7]. The assumptions render them improper for real-world applications.

This work is motivated by the need of the large NIS, where each sink node powered by unstable energy is responsible for forwarding data updates from multiple sensors without requiring large hardware resource consumptions. In particular, the sink node is required to *immediately* forward or discard the received updates (i.e., without the data buffer) while the optimization of the average data freshness is done by considering the energy causality constraints (i.e., a limited energy buffer and a non-deterministic energy harvesting rate). Based on the need, we propose an offline forwarding algorithm to provide the upper-bound performance of the data freshness problem (the *Age of Information*, AoI [8], [9] is used to quantify the performance) based on the branch-and-bound (B&B) algorithm, which we believe is the first work to explore the optimal update forwarding policy considering the AoI calculation and the energy causality constraints on top of the classical B&B algorithm. In addition, to facilitate the deployment in the field, we present an online algorithm to approximate the presented upper-bound performance, where the decision can be made in a constant time. The experimental results show that our online solution greatly improves the performance by 148% and 7.9%, in terms of the average data freshness, against the two well-known strategies, and achieves 91% of the performance delivered by the ideal solution. Besides, the performance delivered by the online algorithm is close to that by the prior work relying on a data buffer in most cases [7], and saves a significant amount of energy which can be used for forwarding extra status updates or extending the system execution time while improving the A^3oI by up to 9.6%.

The rest of this paper is organized as follows. Section II presents the background and motivation of this work, which is followed by the proposed solutions in Section III, including the offline performance estimation and the online update forwarding algorithm. Section IV provides a series of experimental results to demonstrate the efficiency of the proposed solutions, and Section V concludes this paper.

II. Background and Motivation

A. System Model

The target system model is illustrated in Fig. 1, where a multi-hop network is adopted to cover a wide region. In this work, we use the NIS with a single sink node to present our ideas, and the developed algorithms can be applied to every other sink node presented in the system network, if any. The key components of the NIS are described as follows.

Sensor node. We consider n sensor nodes, each of which is an intermittent system s_i driven by the ambient energy. A sensor generates a sequence of status updates according to a Poisson process of rate λ_i different from the others since each sensor might be deployed at a different location exhibiting a distinctive energy harvesting rate in energy harvesting-based application scenarios [10].

Sink node. A sink node operates intermittently, forwarding the data updates sent from multiple sensor nodes when it has sufficient energy. When a sink node receives a status update $u_{i,j}$, it immediately decides to either *forward* or *discard* the status update based on the designated strategy, and its energy expenditure is normalized to an energy unit for each data forwarding. We assume a sink node is equipped with an energy buffer containing B energy units, and an energy unit is charged based on a Poisson process with the rate of η, according to the prior work [5].

Central controller. A central controller is backed by a stable energy source and is responsible for monitoring the real-time status of the target environment through the data collected by the sensor nodes. The accuracy of the monitoring process can be assessed by the freshness of the updated data, in terms of the Age of Information [8], [9], which is formally defined as below.

Definition 1: The Age of Information (AoI) $\Delta_i(t)$ of an update source s_i at time t is defined as the difference between the current time t and the timestamp $L_i(t)$ of the latest time the central controller received the status update from sensor node s_i:

$$\Delta_i(t) = t - L_i(t). \tag{1}$$

Based on Definition 1, the *average AoI* (A^2oI) Δ_i of a single update source s_i within the time interval $[0, T]$ can be formulated as:

$$\Delta_i = \frac{1}{T}\int_0^T \Delta_i(t)dt. \tag{2}$$

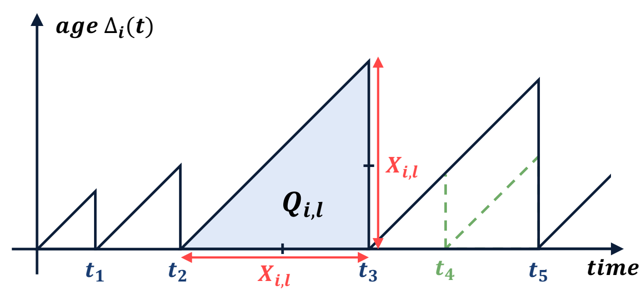

Fig. 2. An example of the age of information.

Fig. 2 depicts the course of an evolution process of $\Delta_i(t)$ observed at a central controller. It is assumed that $\Delta_i(t)$ is initialized to zero at the beginning of the monitoring process, i.e., $t = 0$. Upon the arrival of a new update, the corresponding age of s_i ($\Delta_i(t)$) is reset to zero; otherwise, it grows as time elapses. $X_{i,l}$ denotes the time interval between the l-th and $(l-1)$-th time points, where the central controller receives the status updates of the sensor node s_i. For example, there is an update sent by s_i at t_2 and the central controller receives the update at t_3.

As the areas of isosceles triangles in Fig. 2 represent the A^2oI, Equation 2 can be rewritten as the sum of the isosceles triangle areas $Q_{i,l}$, and the A^2oI of a sensor node s_i in the time interval $[0, T]$ becomes:

$$\Delta_i = \frac{1}{T}\sum_{l=1}^{Y_i} Q_{i,l} = \frac{1}{2T}\sum_{l=1}^{Y_i} X_{i,l}^2, \tag{3}$$

where Y_i is the total number of status updates received by the central controller from sensor node s_i.

Following Equation 3, the system-wide AoI (i.e., multiple sensor nodes) can be measured as the average of A^2oI (denoted as A^3oI), which can be formulated as the average of A^2oI of the total number n of sensor nodes served by a sink node.

$$\frac{1}{n}\sum_{i=1}^{n}\Delta_i = \frac{1}{2nT}\sum_{i=1}^{n}\sum_{l=1}^{Y_i}X_{i,l}^2. \tag{4}$$

B. Motivation

The design of sink nodes in NIS is important for the system to obtain fresh data for further data manipulations required by applications. As the sink nodes are powered by intermittent energy, it is crucial to minimize hardware resource usage/utilization in order to have a longer lifetime and lower maintenance overheads since a higher resource requirement means larger power consumption and more hardware components, which increase the hardware failure rate, the frequency for system maintenances, and hence the total cost of ownership [11], [12].

This work aims at the sink node design with limited hardware usage (i.e., without the presence of the data buffer for storing the updates). The major challenge of such a design is that the sink node has to immediately decide whether forwarding or discarding the arrived status updates, considering the current energy budget and the non-deterministic input energy. Specifically, the decisions are made sequentially over uncertain discrete-time periods for optimizing the system-wide AoI (i.e., A^3oI). An *aggressive* forwarding policy can minimize the AoI at the current time point, but it would miss a better forwarding timing, which produces a lower A^3oI in the long run. On the other hand, a *conservative* forwarding strategy may waste the harvested energy since the harvested energy will be dropped when the energy buffer is full. Besides, in order to obtain a good A^2oI across the system (i.e., A^3oI), the decisions should be made according to the current remaining energy on the sensor nodes since unbalanced A^2oI among sensors would degrade the overall performance A^3oI.

This work is motivated by the challenges of the design of the forwarding strategy for the sink node within a NIS. Our goal is to minimize the A^3oI of the status updates sent by sensors in the NIS to provide the freshest data of a monitoring application. As the AoI value is computed based on the previous forwarding time (according to Definition 1), the solution to the AoI optimization problem cannot be found in linear time, where the AoI optimization problem can be considered as a sequential decision problem. To tackle the challenge of making the decisions immediately, we propose an *AoI-aware Branch-and-Bound (B&B) algorithm* to derive the upper-bound performance of the AoI optimization problem. To the best of our knowledge, we are not aware of any other work that provides such information considering the *AoI calculation characteristics* and *energy causality constraints* on a classical B&B algorithm to find the optimal solution. Furthermore, we propose an *AoI-aware online update forwarding algorithm* to approximate the optimal solution in constant time to provide immediate decisions by considering the dynamic AoI values during runtime and the stochastic update arrival pattern under the energy causality constraints.

III. Data Freshness Optimization

A. Problem Formulation

The objective of the proposed AoI optimization problem is to minimize the A^3oI subject to the energy causality constraints by finding an update forwarding set F from the status update arrival set U in time T, as shown in Equation 5.

$$\begin{aligned}&\text{Minimize } \frac{1}{n}\sum_{i=1}^{n}\Delta_i,\ \forall i \in \{1, ..., n\},\\ &\text{Subject to } 0 \le b(t) \le B,\end{aligned} \tag{5}$$

where $b(t)$ denotes the number of energy units stored in the energy buffer at time t. Following Equation 3, our objective can be also presented as minimizing the total area of all isosceles triangles among n sensor nodes in the given time interval $[0, T]$.

The energy causality constraint consists of two general principles obtained from the energy consumption and energy absorption behavior in the sink node. Specifically, at time $f.t$, the instant before the sink node forwards an arrival status update, the sink node must have at least one unit of energy stored in its energy buffer.

$$b(f.t) \ge 1,\ \forall f \in F, \tag{6}$$

where F is the update forwarding set and f denotes the status update decided to be forwarded at time $f.t$. Moreover, the harvested energy unit increases over time unless the number of energy units reaches the maximum capacity B of the energy buffer. Thus, with the energy charging rate η and $E(\eta)$ denoting the number of energy units arrives in $[t-1, t)$, the energy unit is defined as Equation 7.

$$b(t) = min\{b(t-1) + E(\eta), B\}. \tag{7}$$

B. AoI-aware Branch-and-Bound Algorithm

1) Observations and Limitations: There exist many algorithms proposed for searching the exact solution of optimization problems, such as the Branch-and-Bound (B&B) algorithm. However, these algorithms may be *inefficient* and, more seriously, *incorrect* when deployed on the proposed AoI optimization problem. This is because they are designed for stable-powered conditions without considering AoI as their performance metric. Specifically, the *A^2oI monotonically increases as the time elapses* since it is calculated by adding all the convolution areas (i.e., $\frac{1}{2}X_{i,l}^2$), as illustrated in Equation 3. For example, the area constituted by the evolution of AoI at time t is always smaller than the area at time $t + a$, no matter how long the time goes.

Moreover, the *A^2oI always becomes smaller when the total number of update forwarding is larger*. This is because any forwarding decision divides the area of an isosceles triangle into two smaller isosceles triangles and a parallelogram, where the parallelogram can represent the performance gain. For example, in Figure 2, the isosceles triangle between the time interval $[t_3, t_5]$ is divided into three parts by the green dashed line if the sink node decides to forward the arrival status update at time

t_4. These characteristics allow us to *efficiently* reduce a large amount of searching time by halting the searching process of the non-optimal solution spaces on a sub-problem.

On the other hand, the solution may *wrong* if the consumed energy exceeds the harvested energy even the total energy consumption is not, without considering the causation of energy harvesting condition. Therefore, we propose to introduce the *AoI calculation characteristics* and *energy causality constraint* into the B&B algorithm.

2) AoI-aware Branch-and-Bound Algorithm: A branch-and-bound (B&B) based algorithm design paradigm leverages a tree search strategy to enumerate all possible solutions systematically to find the optimal solution for a given problem. It naturally fits the A^3oI optimization problem since each decision (forward or discard) is *binary* and the problem can be formulated as the binary tree to represent the sequence of the binary decisions (i.e., forward or discard), where the B&B can be applied on the tree to find the solution. With the profiles of the harvested energy and the status updates, the made decisions can be constructed as a completed B&B binary tree that represents the entire solution space to the optimization problem. That is, the optimal solution can be identified by visiting all decision sequences (tree paths) with B&B algorithm.

Fig. 3 illustrates an example of mapping the A^3oI optimization problem onto the B&B algorithm. For example, the nodes #2 and #9 represent the forwarding/discard decisions made for the first status update $u_{1,1}$, and the nodes #6 and #7 are for the fourth update (i.e., $u_{1,3}$). Comparing the A^3oI values for the nodes #6 and #7, one can simply know that the forwarding decision made for $u_{1,3}$ achieves a better result (with a lower A^3oI value). The search path from the root node toward node #6 can be regarded as a candidate for the optimal solution. On the other hand, for the nodes with larger A^3oI values, they will be *bounded*, e.g., the nodes #8 and # 10 are determined to be non-optimal candidate solutions because the A^3oI monotonically increases as the time elapses.

The depth-first search algorithm is adopted along with the B&B paradigm on the binary tree to look for the optimal solution, where the tree node number is the same as the sequence number during the depth-first search, and the node label (e.g., $\mathtt{b} = 1$) represents the remaining energy at the time point of making a decision.

It is worth noting that the improper states will produce inaccurate and inefficient results, and they should be bounded too. The underflow and overflow situations are the improper states, and they render the system as a lack of energy ($b(t) < 0$) that incurs wrong result, or waste of energy ($b(t) > B$) that can be regarded as non-optimal candidate solutions since the A^3oI always becomes smaller when the total number of update forwarding is larger. For instance, the nodes #3 and #11 are examples of the underflow ($b < 0$) and overflow ($b > 2$) situations, given that the capacity of the energy buffer is two units. By pruning off the sub-trees rooted by the nodes with improper states, the proposed B&B algorithm finds the result correctly and efficiently in the depth-search fashion.

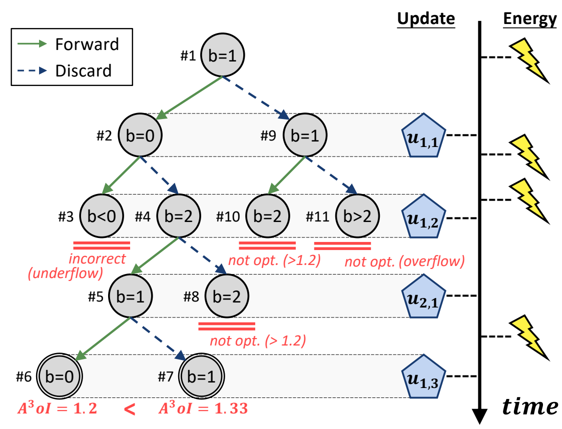

Fig. 3. Illustration of the AoI-aware B&B algorithm.

C. AoI-aware Update Forwarding Algorithm

The optimal A^3oI can be approximated by minimizing Equation 4, which means the area of each isosceles triangle ($\frac{1}{2}X_{i,l}^2$) should be minimized that further implies the minimization of $X_{i,l}$ denoted as X_{opt}. According to Lemma 1, X_{opt} can be obtained when a sink node can forward the updates from each sensor node with the equal opportunity. As an energy unit is used to forward a status update, $(\eta * T)/n$ is used for n sensor nodes to equally share the harvested energy at the sink node within a time interval T, where $(\eta * T)$ represents the total number of harvested energy unit. To further determine X_{opt} for each sensor node, $T/((\eta * T)/n)$ is thus used to estimate the best forwarding time interval based on the provided energy on the sink node. The derivations of the above concept is provided in Equation 8.

$$X_{opt} \approx \frac{T}{(\eta * T)/n} = \frac{n}{\eta}. \tag{8}$$

It is worth noting that the equation is designed based on the concept of equally spending the energy resource on each sensor node. Hence, X_{opt} is the same across sensor nodes, implying that all the sensor nodes have the same priority in the NIS and they have the same aging rate for the target application.

Lemma 1: Assume that the sum of n terms is fixed and equal to M. The minimum of the sum of squares is M^2/n when each term is equal to M/n, according to the Cauchy–Schwarz inequality [13].

The proposed AoI-aware update forwarding algorithm is developed with the derived X_{opt}, as listed in Algorithm 1. The sink node forward the status update only if the time interval generated by the current update (i.e., $|u_{i,j} - u_{i,j-1}|$) is closer to X_{opt} ($= \frac{n}{\eta}$) than the time interval generated by the next status update (i.e., $|u_{i,j+1} - u_{i,j-1}|$). The exception is that in order to avoid the energy waste, the forwarding is triggered when the energy buffer is full, as depicted in Line #6. Note that the timestamp of the next arrival status updates $u_{i,j+1}$ of node i is calculated according to λ_i under the stochastic analysis model. It is important to note that in order to achieve the time complexity of O(1), the forwarding interval is compared with the sensor with the updated status, rather than other sensors (which results in O(n)).

Algorithm 1: AoI-aware Update Forwarding Algorithm

input : $E = \{e_1, e_2 ..., e_m\}$, $U = \{u_{1,1}, u_{1,2}, ..., u_{1,j}, ...u_{i,j}\}$, B
output: Update forwarding decision

$b(t) \leftarrow 0$
while *true* **do**
  if *harvested energy unit arrives* **then**
    $b(t) \leftarrow \min\{b(t) + 1, B\}$
  if *update arrives and* $b > 0$ **then**
    if $b(t) = B$ *or* $|n/\eta - (u_{i,j} - u_{i,j-1})| < |n/\eta - (u_{i,j-1} - u_{i,j+1})|$ **then**
      Forward the current update
      $b(t) \leftarrow b(t)$ - 1
    else
      Discard the current update

IV. Performance Evaluation

A. Experimental Setup

We develop an AoI-based simulation for evaluating performance by comparing the proposed AoI-aware online update forwarding algorithm (denoted as *Online*) with the well-known strategies which are *best-effort update forwarding algorithm* [3] (denoted as *Best-effort*) and *energy-balancing update forwarding algorithm* [14] (denoted as *Balanced*). Specifically, the best-effort algorithm always chooses to forward the received status updates as long as it has sufficient energy, while the balanced algorithm decides to forward the updates when the data freshness of the corresponding source exceeds a pre-defined A^3oI threshold. Furthermore, we compare the performance with the state-of-the-art design of an energy-harvesting sink node in [7], denoted as *EH_offline* and *EH_online* for their offline and online algorithms, respectively, where a data buffer is dedicated to store the arrived updates when the sink node does not have sufficient energy for forwarding.

In the simulation, we consider n sensor nodes s_i continuously generate a sequence of status updates with an independent rate λ_i and send them to the central controller via a sink node. The sink node harvests energy at a rate η, and it is equipped with an energy buffer, which can store B units of energy. Note that both λ_i and η are generated followed by the Poisson process [5], [7]. The system is assumed to start with an empty energy buffer and zero ages.

B. Comparing to the Optimal Solution

Figure 4 shows the A^3oI of five different update forwarding strategies normalized to the optimal solution, i.e., the proposed AoI-aware B&B algorithm (denoted as *AoI-B&B*). The performance difference between our online algorithm (denoted as *AoI-forwarding*) and AoI-B&B is only 9%, and AoI-forwarding achieves about 8% and 161% performance improvement over the balanced and best-effort strategies, respectively. It is important to note that AoI-B&B performs better than EH_offline (which renders that EH_offline is not an optimal solution) and AoI-forwarding performs as close as EH_online (about 1%)

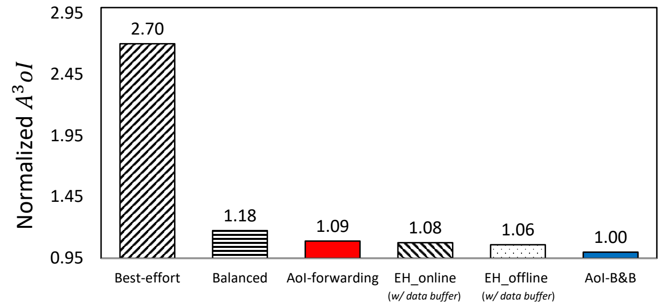

Fig. 4. The A^3oI normalized to the optimal solution.

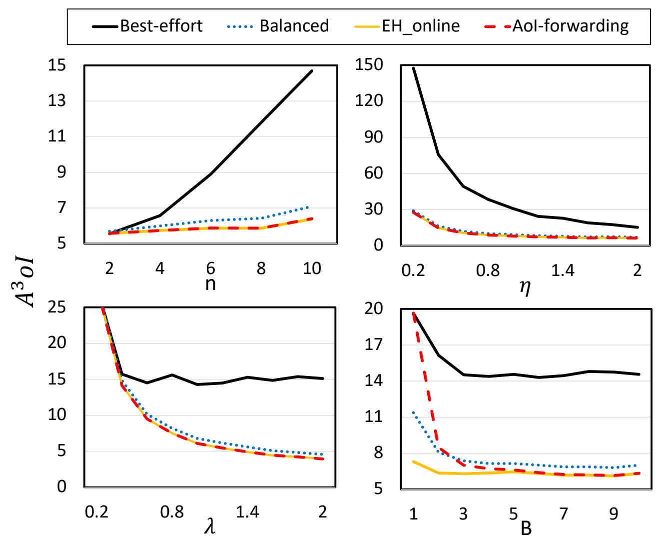

Fig. 5. The delivered A^3oI with different n, η, λ_i, and B settings.

without the presence of a data buffer for storing the updates. A closer comparison between our work and the prior work (EH_online) is given in Section IV-D.

C. Evaluating Performance under Different Parameters

In order to demonstrate the capability of the proposed solutions in different scenarios, we perform the experiments with different settings for the four system design parameters (i.e., n, η, λ_i, and B). By default, the parameters n, η, λ_i and B will be fixed in 10, 2, 1 and 10, respectively. Figure 5 shows that AoI-forwarding (and EH_online) outperforms the well-known strategies (148% for the best-effort and 7.9% than the balanced). The best-effort algorithm, which is not specifically designed for energy harvesting conditions, performs similarly with our online solution when the harvested energy is enough to cover all update forwarding, i.e., when the absorbed energy is sufficient to cover the forwarding of arrived updates. Otherwise, the best-effort strategy performs much worse than AoI-forwarding. That is why best-effort has the worst performance most of the time.

The balanced strategy utilizes the harvested energy conservatively with the fixed AoI threshold, but it does not consider the limited capacity of the energy buffer (i.e., an energy causality constraint). For example, when the capacity of the energy buffer is small, the performance of the balanced strategy is close to AoI-forwarding, but with the increase of B, AoI-forwarding performs better, as shown in the bottom-right of Figure 5. This experiment results further demonstrate that the balanced strategy cannot fully utilize the capacity of the energy buffer.

When the energy buffer size sets to one as shown in the bottom-right of Figure 5, AoI-forwarding performs worse than others since the design philosophy at this extreme case is to make the best use of absorbed energy (to not waste energy) and AoI-forwarding forwards the arrived updates whenever possible. In such a case, AoI-forwarding has the same performance as the best-effort algorithm. On the other hand, EH_online uses a data buffer to forward the stored updates when the energy is enough for the transmission. Our results suggest when the energy buffer size becomes larger, AoI-forwarding is competitive or similar to that delivered by EH_online without the energy consumed by the data buffer. For example, when $B = 3$, the performance difference between AoI-forwarding and EH_online is only 10.27%, and less than 1% when $B > 5$.

D. Performance Impact of the Design without Data Buffer

TABLE I
PERFORMANCE ACHIEVED BY AOI-FORWARDING AGAINST EH_ONLINE [7].

η	Saved power (mW)	A^3oI difference (%)
5	21.84	-2.296
4	37.17	-1.846
3	46.79	-2.052
2	50.03	+0.002
1	50.49	+9.682

While EH_online benefits from the data buffer when B is small (as described in Section IV-C), it consumes a significant amount of energy to store the received data, especially when the absorbed energy is not sufficient for transmitting a relatively large size of incoming data. Table I lists the saved energy and the difference of the A^3oI of AoI-forwarding against EH_online [7], using the default settings described in Section IV-C under the different settings of η. The bottom of the table shows that AoI-forwarding is an energy-saving solution, achieving a better A^3oI value, which is about 9.6% better than that delivered by EH_online. Specifically, EH_online requires the extra 50.4 mW of power to buffer the data when the harvested energy ($\eta = 1$) cannot afford to the data transmission. Concretely, this amount of power can support an intermittent device (e.g., TI MSP430) to run for 10 more seconds, or it can be used to forward 400 pieces of status update data (e.g., with the CC2420 radio transceiver). Furthermore, EH_online is not aware of the data freshness for the stored data as it always decides to forward the longest buffered update without comparing the AoI among sensor nodes, and hence, its A^3oI is 9.6% larger than ours.

V. CONCLUSION

This paper studies the update forwarding strategies of the EH-based sink node to optimize the data freshness among multiple sensor nodes in NISs. Specifically, the sink node has to decide which status updates received from multiple sensor nodes should be discarded or forwarded to the central controller for monitoring. We develop an AoI-aware B&B algorithm as the optimal solution and a constant time online update forwarding strategy to optimize the data freshness without being over-relevant on the need of data buffer. Our online solution can greatly improve the average data freshness for 148% against the well-known solutions, and it is only 9% worse than the optimal solution. It is important to note that the performance of AoI-forwarding is close to that of the prior work relying on a data buffer in most cases (i.e., except when the capacity of energy buffer is extremely low), and save a significant amount of energy for more update forwarding and extend the system execution time.

VI. ACKNOWLEDGEMENT

This work was supported in part by National Science and Technology Council under grant no. 111-2221-E-006-116-MY3 and Ministry of Education under Yushan Young Fellow Program. In addition, this work is financially supported in part by the "Intelligent Manufacturing Research Center" (iMRC) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

REFERENCES

[1] K. S. Adu-Manu, N. Adam, C. Tapparello, H. Ayatollahi, and W. Heinzelman, "Energy-harvesting wireless sensor networks (eh-wsns): A review," *ACM Transactions on Sensor Networks (TOSN)*, vol. 14, no. 2, 2018.

[2] W. S. Lim, C.-H. Tu, C.-F. Wu, and Y.-H. Chang, "icheck: Progressive checkpointing for intermittent systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 40, no. 11, pp. 2224–2236, 2021.

[3] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 3008–3012.

[4] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *Information Theory and Applications Workshop (ITA)*, 2015, pp. 25–31.

[5] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Transactions on Green Communications and Networking (TGCN)*, vol. 2, no. 1, 2018.

[6] Z. Zhou, C. Fu, C. J. Xue, and S. Han, "Transmit or discard: Optimizing data freshness in networked embedded systems with energy harvesting sources," in *ACM/IEEE Design Automation Conference (DAC)*, 2019.

[7] Z. Zhou, C. Fu, C. Xue, and S. Han, "Energy-constrained data freshness optimization in self-powered networked embedded systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 39, pp. 2293–2306, 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," 2020.

[9] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1897–1910, 2016.

[10] S. Ahmed, M. Nawaz, A. Bakar, N. A. Bhatti, M. H. Alizai, J. H. Siddiqui, and L. Mottola, "Demystifying energy consumption dynamics in transiently powered computers," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 19, no. 6, 2020.

[11] M.-C. Yang, Y.-M. Chang, C.-W. Tsao, P.-C. Huang, Y.-H. Chang, and T.-W. Kuo, "Garbage collection and wear leveling for flash memory: Past and future," in *2014 International Conference on Smart Computing*, 2014.

[12] M.-C. Yang, Y.-H. Chang, C.-W. Tsao, and P.-C. Huang, "New era: New efficient reliability-aware wear leveling for endurance enhancement of flash storage devices," in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013.

[13] J. M. Steele, *The cauchy-schwarz master class: An introduction to the art of mathematical inequalities*. Cambridge University Press, 2004.

[14] S. Abdelhak, C. S. Gurram, S. Ghosh, and M. Bayoumi, "Energy-balancing task allocation on wireless sensor networks for extending the lifetime," in *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2010, pp. 781–784.