Balancing Age-Energy Tradeoff in Sleep-Wake Server Systems

Jin Xu, Xinyuan Wu, Qisheng Huang Member, IEEE, and Peng Sun

Abstract—The surging demand for fresh information from various Internet of Things (IoT) applications requires oceans of data to be transmitted and processed timely. How to guarantee information freshness while reducing energy consumption thus becomes imperative. We consider a multi-source single-server queueing system, where we aim to design the optimal sleep-wake strategy for the server to reduce its energy consumption while guaranteeing users’ information freshness. We propose a sleep-wake strategy that relies on an idling scheme called Conditional Sleep (CS) scheme. We show that the proposed CS scheme can achieve a smaller Age of Information (AoI) than the widely-used Hysteresis Time (HT) scheme and Bernoulli Sleep (BS) scheme, while retaining the same power consumption and Peak Age of Information (PAoI). Moreover, we find that increasing the sleep period length can always reduce energy consumption and enlarge the PAoI, but it does not always increase AoI. We also find that using PAoI as the information freshness metric in designing the optimal sleep-wake strategies would make the server sleep infinitely long. Our analysis reveals that this result is due to the PAoI being a first-order statistic. We further extend our discussion to the scenario where data sources choose sampling rates strategically based on the sleep-wake strategy of the server. We show that increasing the sleeping period length for the server while guaranteeing users’ PAoI could lead to a minor reduction of the server’s energy consumption but significantly increase the data sources’ sampling costs.

Index Terms—Information freshness, energy efficiency, sleep-wake server, queueing analysis.

I. INTRODUCTION

A. Motivations

Information freshness has garnered wide attention from academia and industry nowadays due to its influential role in communication theory and decision science. Users in the Internet of Things (IoT) applications such as smart manufacturing systems and autonomous vehicles [1]–[3] can better infer the actual physical process status with fresher information. To guarantee information freshness for users, the sampling devices such as cameras and monitors need to sample the physical process closely. The transmission and processing of these sampled data, such as high-resolution images and videos, bring substantial energy consumption for the communication systems (e.g., access points and base stations) and computing entities (e.g., processors and servers). For instance, if the sampled information is transmitted wirelessly, the base stations would account for over 80% of the cellular network energy consumption [4], [5].

In order to reduce energy consumption for base stations and servers, sleep-wake strategies can be adopted in communication systems [5]. Sleep-wake strategies allow a base station (server) to transit to a sleep mode with low energy consumption whenever the traffic load is light. However, having the base station sleeping could also result in system inefficiency, since the sleeping base stations or servers may not wake up as soon as new transmission or processing requests occur. Most of the previous studies were focused on characterizing the delay-energy tradeoff of sleep-wake strategies, i.e., how much energy can be traded off by a tolerable delay. These studies investigated the delay-energy tradeoff in different communication systems, such as small cell networks [6], ultra-dense networks [7], hyper-cellular networks [8], and cellular communication systems with IoT environments [9].

However, information freshness metrics are quite different from the delay metric. First, the delay metric measures each data packet’s waiting and transmission time, and the information freshness metrics measure how timely the users are informed. In many IoT applications, information freshness rather than delay is of people’s major interest. For instance, in smart manufacturing systems, images taken for physical processes over time need to be processed by a centralized computer (server) for defect and anomaly detection [10]. Timely information will assist in on-the-fly decision-making, so the information freshness is of interest in such a scenario. Second, the scheduling policies for the delay and information freshness metrics are distinct. The scheduling policies for reducing delay usually require the base stations and servers to transmit every data packet as quickly as possible. In contrast, to achieve optimal information freshness, the base stations or servers need to transmit or process the freshest packet as soon as possible, and stale data packets can be dropped to reduce system redundancy. These differences make the policies balancing the delay-energy tradeoff inapplicable to the scenarios where energy and information freshness are metrics of interest.

This paper aims to analyze the tradeoff between information freshness and energy consumption. As in [11], [12], we characterize information freshness using Age of Information (AoI) and Peak Age of Information (PAoI), which we will formally define in Section III. Specifically, we aim to answer the following research questions.

- How to characterize the AoI, PAoI, and energy...
consumption for communication systems with sleep-wake base stations and servers?

- Does the AoI-energy or PAoI-energy tradeoff always exist?
- How to design the optimal strategy that balances the energy and information freshness tradeoff, and what are the differences in strategies between considering AoI and PAoI as information freshness metrics?

The questions above are challenging for the following two main reasons: 1) It is difficult to characterize the information freshness metrics for multiple data sources associated with the base station (see [3], [13]); 2) Since the processing time for data packets and setup time for the base station/server can be generally distributed, we cannot apply analytical methods that rely on exponential assumptions, such as continuous-time Markov Chain analysis [14] and Stochastic Hybrid System analysis [13].

To answer the above questions, we model the communication system as a queueing system with multiple information sources and a single server. We then derive the closed-form expressions of AoI, PAoI, and the energy consumption under different idling schemes based on a renewal analysis for queueing systems. We further investigate the AoI-energy and PAoI-energy tradeoffs and develop insights.

B. Key Contributions

The main contributions of our paper are summarized as follows.

- **Closed-form expressions of information freshness metrics:** We use a renewal-type analysis to provide the closed-form expressions for AoI, PAoI, and the energy consumption for the queueing system with a sleep-wake server and multiple data sources. This analytical method can be adopted in other queueing systems to evaluate the information freshness. The closed-form expressions we derive can be further used to evaluate system performance or design communication protocols.

- **Comparative analysis on idling schemes:** We propose a novel idling scheme called Conditional Sleep (CS) scheme and compare it with two widely used idling schemes, namely Hysteresis Time (HT) scheme and Bernoulli Sleep (BS) scheme. We show that CS scheme achieves a smaller AoI than HT and BS but retains the same PAoI and energy consumption.

- **Age-energy tradeoff:** We find that extending the sleeping period of the server will reduce the energy consumption, increase the PAoI, but will not always increase the AoI. This result is due to the peculiar definition of the AoI. We further provide the conditions under which the AoI does not increase as the sleeping period length increases. Our analytical results provide practitioners with guidance to evaluate how much energy can be saved by sacrificing users’ information freshness.

- **Difference between metrics AoI and PAoI in optimization problems:** We find that minimizing the energy consumption under a PAoI constraint can induce an optimal sleep-wake strategy that sleeps infinitely long. The optimization problem with an AoI constraint does not have this issue, but it becomes difficult to solve. Our analysis reveals these differences are due to PAoI being a first-order statistic, and AoI being determined by second-order statistics.

- **Impact of strategic behaviors:** We construct a Stackelberg game between the server and data sources, where the server is the service provider, and the data sources are service takers. We show that extending the sleeping length of the server while guaranteeing users’ information freshness would lead to a high sampling cost for the data sources, while incurring only minor energy reduction for the server. This result suggests that keeping a relatively short sleeping length for the server could lead to the socially optimal solution for the server and data sources altogether.

The rest of this paper is organized as follows. In Section II, we introduce the work related to this study. In Section III, we present the system model. We derive the closed-form expressions using queueing analysis in Section IV. In Section V, we obtain the optimal sleep-wake strategies based on the closed-form expressions for system performance. We extend our discussion to strategic data sources in Section VI. Section VII further develops insights by conducting numerical studies. Finally, we provide concluding remarks and discuss future research in Section VIII.

II. RELATED WORK

This section will first review the studies investigating the delay-energy tradeoff in different communication systems in Section IIA. We will then discuss the papers that focus on the sleep-wake strategies in queueing systems in Section IIB and review the recent work about sleep-wake strategies with information freshness consideration in Section IIC.

A. Delay-energy Tradeoff in Communication Systems

Sleep-wake strategies have been designed and investigated in many communication applications. Guo et al. [13] studied sleep-wake strategies in heterogeneous networks (HetNet) and hyper-cellular networks (HCN). Pei et al. [7] investigated the sleep-wake base stations in ultra-dense networks and modeled the system as an M/G/1/N processor sharing vacation queueing system. Liu et al. derived the coverage probability, achievable rate, and energy efficiency for the sleep-wake base stations in small cell networks [6]. Verma et al. [8] considered a paging mechanism where IoT devices transmit and receive data with sleep-wake cycles. Wu et al. [16] studied the sleep control and power match strategies for cellular networks with bursty traffic. The relationship between sleep depth and energy consumption was modeled and studied by Onireti et al. [17]. In [18], Feng et al. studied the problem of base station ON-OFF switching, user association, and power control in HetNet with massive multiple-input-multiple-output (MIMO). The other applications of sleep-wake strategies in communication networks were also discussed in [19]. However, all the studies mentioned above focused on the
delay-energy tradeoff without discussing the tradeoff between energy and information freshness. The tradeoff between energy and information freshness in sleep-wake server systems has not been fully understood.

B. Different Sleep-wake Strategies in Queueing Systems

Communication networks are usually modeled as queueing systems. A sleep-wake strategy in queueing systems usually consists of two parts: a wakeup scheme that determines when the server should wake up and an idling scheme that determines when the server should sleep.

There exist different wakeup schemes in queueing systems, such as N-policy [20], single-sleep scheme [21], multiple-sleep scheme [17]. Under the N-policy, the base station continues sleeping until the queue accumulates N data packets. Under the single-sleep scheme, the server sleeps for a certain period and then wakes up. For the multiple-sleep scheme, the server sleeps for multiple periods until the system becomes non-empty. All these studies [17], [20], [21] assumed that the server would enter the sleeping period once the system becomes empty.

Some studies also discussed the idling scheme in sleep-wake strategies. Niu et al. [5] and Guo et al. [15] discussed Hysteresis Time (HT) scheme under which the server stays idling until either a threshold time is reached or a new packet arrives. Studies like [22], [23] investigated the Bernoulli Sleep (BS) scheme, where the server takes vacations with a probability after completing a task.

However, these studies mainly evaluated classic queueing performance metrics such as mean delay, throughput, idling probability, and energy consumption. The information freshness metrics under different wakeup and idling schemes have not been fully studied.

C. Sleep-wake Design with Information Freshness

Several studies have recently shed light on the tradeoff between energy and information freshness. Bedewy et al. [24] studied the sleep-wake scheduling for sensors to balance the information freshness and energy tradeoff from the sampler’s perspective. Huang et al. [25] studied the information freshness and energy tradeoff in fading channels, where strategies were designed to minimize the weighted summation of AoI and energy consumption for sensors. These two papers mainly considered the energy consumption of samplers. The tradeoff between information freshness and energy in sleep-wake server systems has not been fully investigated. Xu and Chen [26] analyzed the PAoI in single-source systems with the Last Come First Serve (LCFS) scheme and N-policy, single-sleep, and multiple-sleep as sleep-wake strategies, without further discussing the information freshness and energy tradeoff.

In summary, the tradeoff between energy and information freshness in systems with sleep-wake servers has not been fully understood. It is still unclear how the server should sleep and wake to achieve the optimal information freshness and energy tradeoff. Investigating how much energy can be traded off by an acceptable information freshness loss in communication systems can further help people design efficient algorithms and better utilize the communication facilities.

III. SYSTEM MODEL

In this section, we will first describe the queueing model and sleep-wake strategies and then introduce the information freshness and energy metrics. Table I contains most of the notations used in this paper. Throughout this paper, we let \( F_X(u) \) be the cumulative distribution function (CDF) for random variable \( X \), \( X^*(s) \) be the Laplace Stieltjes Transform (LST) of random variable \( X \), and \( X^{(k)}(s) \) be the \( k \)th derivative of \( X^*(s) \).

A. Queueing System

We consider a system with \( k \) data sources and a single server. Each data source \( i \) generates data packets following a Poisson process with rate \( \lambda_i \), and we also call it the sampling rate. Let \( \lambda = \sum_{i=1}^{k} \lambda_i \) be the total sampling rate of all the data sources. The data packets generated by each single data source have a particular receiver (user). Before being received by the user, each packet needs to be processed by the server. The processing times for packets from data source \( i \) are independent and identically distributed (i.i.d.), denoted as \( H_i \). We let \( H_i \) be generally distributed. We assume the buffer at the server can hold at most one packet at a time. New packets that arrive when the server is processing will be rejected. So the model we study is an M/G/1/1 multi-class system with a sleep-wake server. Keeping one buffer in the system can achieve a better information freshness and energy tradeoff, as we will show later.

One can regard the server as the base station in wireless communication systems, and the processing time in this scenario is the data transmission time. One can also regard the server as a computing entity, where processing time is needed to extract useful information. So the queueing system we study could be the abstraction of a wide range of communication systems.

B. Sleep-wake Strategies

We consider the sleep-wake strategies consisting of two parts: an idling scheme that determines whether the server...
should sleep or idle when no packet is waiting, and a wakeup scheme that determines when the server should wake up after sleeping. In this paper, we consider three following idling schemes:

**Hysteresis Time (HT) scheme:** The server would wait for \( D_i \) amount of hysteresis time if a packet from source \( i \) was just processed, where \( D_i \) is a random variable. If there is any arrival during this period, the server will resume working immediately upon the packet’s arrival. If there is no arrival during this period \( D_i \), the server will sleep. This scheme was discussed in studies such as [15], [26].

**Bernoulli Sleep (BS) scheme:** After processing a packet from source \( i \), the server transits to the sleep mode with probability \( \theta_i^{BS} \), or stays idling with probability \( 1 - \theta_i^{BS} \) until next packet arrives. This scheme is easy to implement and can be found in many queueing systems (see [22], [23]).

**Conditional Sleep (CS) scheme:** After processing a packet from source \( i \), if \( H_i < B_i \) with \( B_i \) being a random variable, then the server sleeps immediately. Otherwise, the server remains idling until the next arrival occurs. In this work, we assume \( B_i \) is exponentially distributed with rate \( b_i \).

We note that both HT and BS are idling schemes studied in other papers, but CS is a new scheme proposed in this paper. The idea of CS is that if the server already experiences a long processing time, then we let the server stay idling so that the new arrival can be processed as soon as possible. We will later show that CS has the advantage in minimizing AoI over the other two schemes.

This paper will mainly focus on the N-policy as the wakeup scheme due to its tractability. Under the N-policy, the server will wake up whenever \( N \) packets have arrived during the current sleeping period [3], [15]. We assume that the buffer will only keep the last packet among those \( N \) packets. After waking up, the server would experience a setup time \( U \). During the setup time, newly arrived packets can still enter the buffer, but the buffer will only keep the last one. This practice of keeping the freshest packet in the buffer has been proven effective in improving information freshness (see [3], [11], [12]). A demonstrative graph of the system model is given in Fig. 1.

![Fig. 1. System Model](image)

We will provide our analysis for other wakeup schemes such as single-sleep scheme and multi-sleep scheme in Appendix A of the supplementary material.

### C. Information Freshness Metrics

In this work, we use AoI and PAoI as metrics to characterize information freshness. To formally define the AoI and PAoI, we first define the age for receiver \( i \) at time \( t \) as \( \Delta_i(t) = t - T_i(t) \), where \( T_i(t) \) is the generation time of the freshest packet received by receiver \( i \) before time \( t \). By assuming the system being stationary and ergodic, we define the AoI (denoted as \( E[\Delta_i] \)) as the time-averaged value (stationary expectation) of \( \Delta_i(t) \), i.e., \( E[\Delta_i] = \lim_{T \to \infty} \frac{1}{T} \int_0^T \Delta_i(t) \, dt \).

The age process \( \Delta_i(t) \) is piecewise linear, and each age peak occurs right before a fresh packet is received by receiver \( i \) (completes processing). By letting \( A_{i,l} \) be the \( l \)th age peak of \( \Delta_i(t) \) since time 0, we can then define the PAoI (denoted as \( E[A_i] \)) as the time-averaged (i.e., expected) age peaks, i.e., \( E[A_i] = \lim_{k \to \infty} \frac{1}{k} \sum_{l=1}^{k} A_{i,l} \). Both AoI and PAoI can be used to characterize information freshness as they share some similar properties [3], [11]. However, as we will show later, their properties in sleep-wake server systems could be distinct.

### D. Energy Consumption Rate

We also aim to characterize the server’s expected energy consumption rate \( E[P] \), i.e., the amount of energy consumed per unit time. Specifically, we assume that the server has different energy consumption rates when it is at different status, namely \( P_B \) for processing, \( P_{ID} \) for idling, \( P_{SL} \) for sleeping, and \( P_{ST} \) for setup. The server’s expected energy consumption rate \( E[P] \) is thus a function of \( P_B, P_{IS}, P_{SL}, \) and \( P_{ST} \). As we consider the scenario where sleeping can reduce the energy consumption for the system, we assume that \( P_{SL} \leq \min\{P_B, P_{ID}, P_{ST}\} \). We also assume that \( P_{ID} \leq P_B \), as the idling period usually has a smaller energy consumption than the processing period (see [15], [17]).

### IV. Queueing Analysis

This section will derive the closed-form expressions of AoI, PAoI, and energy consumption rate for the system under different idling schemes introduced in Section III. Using those closed-form expressions, we further show how system parameters jointly determine each policy’s AoI, PAoI, and energy consumption rate. We will then compare the performance of these idling schemes.

#### A. Hysteresis Time Scheme

To derive the closed-form expressions for system performance metrics for HT, we first introduce the concept of regenerative cycles. We then derive the closed-form expression of energy consumption and information freshness metrics based on the LST of regenerative cycles.
1) Regenerative Cycles: We define the time span from processing a packet from data source \( i \), to the next time when the server starts processing a packet, as a Class \( i \) regenerative cycle \( V_i \). In order to derive the distribution of \( V_i \), we first consider different types of period that the server experiences within each cycle. Notice that \( V_i \) begins with processing a packet with time \( H_i \). After processing a packet, the server under HT will remain idling until either of the following two cases occurs: 1) an arrival occurs before the hysteresis time \( D_i \) is over, or 2) it has idled for time \( D_i \).

In the first case, \( V_i \) is over when an new arrival occurs. Following the superposition of Poisson arrival processes (see [27]) and the memoryless property of exponential inter-arrival times of packets, we obtain the LST of the idling period as

\[
E[e^{-sL_i}|D_i \geq L]P(D_i \geq L) = \frac{\lambda}{\lambda + s}(1 - D_i^*(s + \lambda)).
\]

The LST of \( V_i \) in this case is then given by \( H_i^*(s)\frac{\lambda}{\lambda + s}(1 - D_i^*(s + \lambda)) \).

In the second case, no arrival occurs during \( D_i \). The period \( V_i \) will contain an idling period, a sleeping period, and a setup period. The idling period's LST in this case is given by

\[
E[e^{-sL_i}|D_i \leq L]P(D_i \leq L) = D_i^*(\lambda + s).
\]

After the idling period, the server will further experience a sleeping period until \( N \) packets have arrived, and a setup period \( U \). So the LST of \( V_i \) in the second case is \( H_i^*(s)D_i^*(\lambda + s)\frac{\lambda}{\lambda + s}\)\( N U^*(s) \).

By combining the LST of \( V_i \) in the above two cases, we have

\[
V_i^*(s) = H_i^*(s)\left( \frac{\lambda}{s + \lambda}(1 - D_i^*(s + \lambda)) + D_i^*(s + \lambda)\frac{\lambda}{s + \lambda} \right)^N U^*(s). \tag{1}
\]

Moreover, we can derive the probability that the server sleeps within a class \( i \) regenerative cycle as

\[
\theta_{HT}^i = P(D_i \leq L) = D_i^*(\lambda).
\]

We will rely on the closed-form expressions of \( V_i^*(s) \) and \( \theta_{HT}^i \) in our derivations later.

2) Energy Consumption Rate: We now use the results of regenerative cycles to derive the energy consumption rate. Notice that each regenerative cycle starts by processing a packet, and this data packet should be either 1) the only data packet arrived during the idling period of the previous regenerative cycle, or 2) the last packet that arrived during the sleeping or setup period in the previous regenerative cycle. By the Bernoulli splitting of Poisson processes [27], this packet has probability \( \frac{\lambda_i}{\lambda} \) to belong to data source \( i \). As a result, the probability of having a Class \( i \) regenerative cycle is \( \frac{\lambda_i}{\lambda} \). The expected length for Class \( i \) regenerative cycle is

\[
E[V_i] = -V_i^{(1)}(0),
\]

then the expected length of all the regenerative cycles is given as

\[
\sum_{i=1}^{k} \frac{\lambda_i}{\lambda} E[V_i] = \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ E[H_i] + \theta_{HT}^i (\frac{N - 1}{\lambda} + E[U]) + \frac{1}{\lambda} \right]. \tag{2}
\]

Letting \( \lambda_c \) be the arrival rate of the regenerative cycles, and from the fact that \( \lambda_c \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} E[V_i] = 1 \), we have

\[
\lambda_c = 1 \left\{ \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} [E[H_i] + \theta_{HT}^i (\frac{N - 1}{\lambda} + E[U]) + \frac{1}{\lambda}] \right\}. \tag{3}
\]

Then the expected energy consumption rate is given by

\[
E[P_{HT}] = \left\{ \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ P_B E[H_i] + \frac{1 - \theta_{HT}^i}{\lambda} P_{TD} + \theta_{HT}^i (\frac{N - 1}{\lambda} + E[U]) + \frac{1}{\lambda} \right] \right\}. \tag{4}
\]

3) Information Freshness: We now introduce the way to derive AoI and PAoI. In each regenerative cycle, the server can only process one packet. The server will reject all the other packets when it is processing, sleeping, or setting up. We call these packets that are processed the “informative packets”. Note that only informative packets result in age drops (see [11]). We let \( G_i \) denote the waiting time of an informative packet from data source \( i \), and \( I_{si} \) be the time span from processing a packet from source \( i \), to the next time the server is about to process a packet from source \( i \). Then we can derive the LST of age peak \( A_i^*(s) \) for data receiver \( i \) as

\[
A_i^*(s) = G_i^*(s)I_{si}^*(s)H_i^*(s). \tag{5}
\]

Equation [5] holds because the peak age occurs only before an informative packet completes the service. Before the packet completes service, the age at that time instance is equal to the period from the previous informative packet from source \( i \) was sampled to the current time. This time span is comprised of the waiting time \( G_i \) of the previous packet, period \( I_{si} \) and the processing time \( H_i \) of the current packet. The three components \( G_i, I_{si}, \) and \( H_i \) are mutually independent. The reason is that \( G_i \) is the waiting time of the last packet, which is independent of the time span \( I_{si} \) and the processing time \( H_i \) of the current packet. The processing time \( H_i \) of the current packet is independent of the period \( I_{si} \). By a similar argument to Eq. (1) and (2) in [23], the expected PAoI for data receiver \( i \) can be given as

\[
E[A_i] = E[G_i] + E[I_{si}] + E[H_i], \tag{6}
\]

and the AoI for data source \( i \) is

\[
E[\Delta_i] = \frac{E[I_{si}^2]}{2E[I_{si}]} + E[G_i] + E[H_i]. \tag{7}
\]

We now introduce the way to calculate \( E[G_i] \). Notice that an informative packet does not need to wait if it arrives during an idling period. It only has to wait if it arrives when the server is setting up. So the informative packet is either the \( N^{th} \) packet that arrives during the sleeping period, or the last packet that
arrives during $U$. The probability that the server experiences a setup period within a regenerative cycle is $\sum_{i=1}^{k-1} \theta^HT \frac{\lambda_i}{\lambda}$. Since the expected waiting time of an informative packet does not depend on the data source that it comes from, using Lemma 1 of [3], we have $E[G_i] = \sum_{i=1}^{k-1} \theta^HT \frac{\lambda_i}{\lambda} + \frac{l}{\lambda}$.

To obtain $E[A_i]$, we further need to obtain the first and second moment of $\Lambda_i$. Define $I_{ji}$ as the time span from processing a source $j$ packet, to the next time a source $i$ packet starts service. We then have

$$I_{ji}^*(s) = V_j^*(s) \left[ \sum_{i \neq j} \frac{\lambda_i}{\lambda} I_{ij}^*(s) + \frac{\lambda_i}{\lambda} \right],$$

so that

$$I_{ij}^*(s) = \frac{V_j^*(s) \frac{\lambda_j}{\lambda}}{1 - \sum_{i \neq j} \frac{\lambda_i}{\lambda} V_i^*(s)}.$$  \hspace{1cm} (8)

By Equation (6), the PAoI of data receiver $i$ under HT can be obtained as

$$E[A_i^HT] = \frac{E[I_{i1}^*]}{E[I_{11}]} + E[H_1] + \theta^HT \frac{1-U^*(\lambda_1)}{\lambda_1},$$

where

$$E[I_{11}] = E[H_1] + \frac{1-\theta^HT}{\lambda_1} + \theta^HT \frac{N-1}{\lambda_1} + E[U],$$

and

$$E[I_{i1}^*] = \frac{2(1-\theta^HT)}{\lambda_i} + \frac{2D_1^*(1)(\lambda_1)}{\lambda_1} -(2N-D_1^*(1)(\lambda_1)) \frac{\lambda_i}{\lambda_1}$$

$$+ 2D_1^*(1)(\lambda_1) U^*(0) + \frac{\theta^HT N(N+1)}{\lambda_1^2}$$

$$- \frac{2\theta^HT N U^*(1)}{\lambda_1} + \theta^HT U^*(2)(0) + H_1^{(2)}(0)$$

$$+ 2E[H_1] \left[ \frac{1-\theta^HT}{\lambda_1} + \theta^HT \frac{N}{\lambda_1} + E[U] \right].$$

(9)

From Equation (4) and Equation (10) we can see that the influence of hysteresis time $D_i$ on $E[A_i^HT]$ and $E[P_i^HT]$ only reflects through the sleeping probability $\theta^HT$. So that different distributions for $D_i$ would result in the same $E[A_i^HT]$ and $E[P_i^HT]$, as long as the distributions are selected to have the same $\theta^HT = D_1^*(\lambda)$. However, from Equations (11)–(13) we can see that $E[A_i^HT]$ is determined by both $\theta^HT = D_1^*(\lambda)$ and $D_1^*(1)(\lambda)$. So distinct distributions of $D_i$ usually result in different AoI.

\[ B. \text{Bernoulli Sleep Scheme} \]

We now derive the performance metrics of BS. Under BS, the server sleeps with probability $\theta^BS$ after processing a packet from source $i$. Following [22], [23], BS is easy to understand and implement in practice. Similar to the discussion of HT, the expected length of each regenerative cycle under BS is

$$\sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ E[H_i] + \theta^BS \frac{N-1}{\lambda} + E[U] \right] + 1.$$  \hspace{1cm} (14)

Letting $\lambda_c$ be the arrival rate of regenerative cycles, we have

$$\lambda_c = 1 / \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ E[H_i] + \theta^BS \frac{N-1}{\lambda} + E[U] \right] + 1.$$  \hspace{1cm} (15)

We can the obtain the expected energy consumption rate as

$$E[P^{BS}] = \left\{ \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ P_B E[H_i] + \frac{1-\theta^BS}{\lambda} P_{TD} \right] + \theta^BS \frac{N-1}{\lambda} + E[U] \right\} / \left\{ \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \right\}.$$  \hspace{1cm} (16)

Equation (16) shows that once $\theta^BS = \theta^HT$ for each $i$, then BS has the same energy consumption rate as HT.

We then derive the AoI and PAoI under BS. Similar to HT, the waiting time of informative packets under BS only appears when the server is sleeping or setting up, thus $E[G_i] = \sum_{i=1}^{k} \theta^BS \frac{\lambda_i}{\lambda} + 1-U^*(\lambda_i)$. Now we introduce the way to derive $I_{ij}^*(s)$ for BS. The LST of a class $i$ regenerative cycle is given as

$$V_j^*(s) = H_i^*(s) \left[ \frac{\lambda}{s+\lambda} (1-\theta^BS) + \theta^BS \frac{\lambda}{s+\lambda} N U^*(s) \right].$$

(17)

We then have $I_{ij}^*(s) = \frac{V_j^*(s) \frac{\lambda_j}{\lambda}}{1-\sum_{i \neq j} \frac{\lambda_i}{\lambda} V_i^*(s)}$ following the same argument in Section IV-A. We thus derive the PAoI under BS as

$$E[A_i^{BS}] = \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ E[H_i] + \theta^BS \frac{N-1}{\lambda} + E[U] \right]$$

$$+ \sum_{i=1}^{k} \theta^BS \frac{\lambda_i}{\lambda} 1 - U^*(\lambda) + E[H_i].$$  \hspace{1cm} (18)

We also provide the expression of AoI for the single data source scenario as follows.

$$E[\Delta_i^{BS}] = \frac{E[I_{i1}^*]}{E[I_{11}]} + E[H_1] + \theta^BS \frac{1-U^*(\lambda_1)}{\lambda_1},$$

(19)

where

$$E[I_{11}] = E[H_1] + \frac{1-\theta^BS}{\lambda_1} + \theta^BS \frac{N}{\lambda_1} + E[U],$$

(20)

and

$$E[I_{11}^*] = 2E[H_1] \left[ \frac{1-\theta^BS + N \theta^BS}{\lambda_1} - \theta^BS U^*(1)(0) \right]$$

(21)
From the fact that \( I \) is the energy consumption rate and \( \Delta_i \), as we observe from probabilities for both schemes are equivalent (i.e., \( \theta_i^{HT} = \theta_i^{BS} \) for all \( i \in \{1,\ldots,k\} \)), both schemes will result in the same energy consumption rate and PAoI, as we observe from Equations (4), (10), (16), and (18). However, \( E[\Delta_i^{BS}] \) and \( E[\Delta_i^{HT}] \) may not be equivalent even when their sleeping probabilities are equivalent. As shown in Equations (11) and (13), \( E[\Delta_i^{BS}] \) is determined by \( \theta_i^{BS} \), but \( E[\Delta_i^{HT}] \) is determined by both \( \theta_i^{HT} \) and \( D_i^{(1)}(\lambda) \).

C. Conditional Sleep Scheme

In this section, we will discuss the performance of CS. The server under CS only sleeps when the service time \( H_i \) is smaller than the threshold variable \( B_i \). The idea of CS is to remain idling when the server has processed a packet for a long time, so that to reduce the peak age of the next regenerative cycle. For the convenience of analysis, we assume \( B_i \) is exponentially distributed with rate \( b_i \). The sleeping probability after serving a source packet is thus given as

\[
\theta_i^{CS} = \mathbb{P}(H_i < B_i) = H_i^*(b_i).
\]

Similar to the argument of BS and HT, we can obtain the energy consumption rate under CS as

\[
E[P^{CS}] = \left\{ \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ P_B E[H_i] + \frac{1 - \theta_i^{CS}}{\lambda} P_{ID} \right] + \theta_i^{CS} \left( \frac{N}{\lambda} P_{SL} + E[U] P_{ST} \right) \right\} / \left\{ \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ E[H_i] + \theta_i^{CS} \left( \frac{N - 1}{\lambda} + E[U] \right) + \frac{1}{\lambda} \right] \right\}.
\]

The LST of class \( i \) regenerative cycle under CS is then given by

\[
V_i^*(s) = E[e^{-sH_i}|H_i > B_i] \frac{\lambda_i}{s + \lambda_i} \mathbb{P}(H_i > B_i) + E[e^{-sH_i}|H_i \leq B_i] \left( \frac{\lambda_i}{\lambda + s} \right)^{N - 1} U^*(s) \mathbb{P}(H_i \leq B_i) = \left[ H_i^*(s) - H_i^*(s + b) \right] \frac{\lambda_i}{s + \lambda_i} + H_i^*(s + b) \left( \frac{\lambda_i}{\lambda + s} \right)^{N - 1} U^*(s).
\]

From the fact that \( I_i^*(s) = \frac{V_i^*(s) \lambda_i}{1 - \sum_{j=1}^{k} \frac{\lambda_i}{\lambda_j} V_j^*(s)} \), we have the PAoI of data source \( i \) as

\[
E[A_i^{CS}] = \sum_{i=1}^{k} \frac{\lambda_i}{\lambda_i} \left[ E[H_i] + \theta_i^{CS} \left( \frac{N - 1}{\lambda} + E[U] \right) \right] + \frac{k}{\lambda} \sum_{i=1}^{k} \theta_i^{CS} \left[ \frac{1 - U^*(\lambda)}{\lambda} + \frac{1}{\lambda_i} + E[H_i] \right].
\]

When there is only one data source, we can obtain the AoI under CS as

\[
E[\Delta_i^{CS}] = \frac{E[I_i^2]}{E[I_i]} + E[H_i] + \theta_i^{CS} \frac{1 - U^*(\lambda)}{\lambda},
\]

where

\[
E[I_i] = E[H_i] + \frac{1 - \theta_i^{BS} + \theta_i^{CS} N}{\lambda_i} + \theta_i^{CS} E[U],
\]

and

\[
E[I_i^2] = 2(E[H_i] + H_i^*(1) + 1 - \theta_i^{CS})(N - 1) + \frac{2H_i^*(1)}{\lambda} + \frac{N(N + 1) \theta_i^{CS} E[U]}{\lambda} + 2\theta_i^{CS} \frac{N - 1}{\lambda} U^*(0) + \theta_i^{CS} E[U^2] + E[H_i^2].
\]

The closed-form expressions for energy consumption rate, AoI, and PAoI will be useful in comparing the performance of different idling schemes, as we will show in the next subsection.

D. Comparative Analysis

This subsection compares the energy consumption rate, PAoI, and AoI under different idling schemes. From the previous derivations, we find that the energy consumption rate and PAoI under CS are the same as those under BS and HT if \( \theta_i^{CS} = \theta_i^{BS} = \theta_i^{HT} \). This means that by carefully choosing the distribution of \( D_i \) for HT and the threshold \( B_i \) for CS, they can achieve the same PAoI and energy consumption rate as BS. However, the AoI under CS is usually different from that under BS or HT, even when the sleeping probabilities are identical. In the following theorem, we show that CS has an advantage in minimizing AoI over the other two schemes.

Theorem 1. For the single data source scenario with \( \theta_i^{CS} = \theta_i^{BS} = \theta_i^{HT} \) being fixed for HT, BS, and CS, then

\[
E[P^{CS}] = E[P^{BS}] = E[P^{HT}], \quad E[A_i^{CS}] = E[A_i^{BS}] = E[A_i^{HT}], \quad E[\Delta_i^{CS}] \leq E[\Delta_i^{BS}] \leq E[\Delta_i^{HT}].
\]

Proof: See Appendix B of the supplementary material for the detailed proof.

Theorem 1 proves that CS outperforms BS and HT in minimizing AoI, and BS outperforms HT. An intuitive explanation for Theorem 1 is as follows. As shown in Equation (6), AoI is determined by \( E[I_i^2], E[I_i], E[G_i], \) and \( E[H_i] \). When \( \theta_i^{CS} = \theta_i^{BS} = \theta_i^{HT} \), BS, and HT have the same \( E[I_i], E[G_i], \) and \( E[H_i] \). The idling scheme that results in the smallest \( E[I_i] \) would have the smallest AoI.

Under HT, the server would idle for a period after the system becomes empty. If no arrival occurs during the idling period, then the server falls asleep. In this case, the server would experience a large \( I_i \) that consists of a processing period, an idling period, a sleeping period, and a setup period. If the server does not sleep, the \( I_i \) is short since it only consist of a...
processing period and an idling period. The variation of $I_{ii}$ is thus large. Under BS, the server would “toss a coin” to decide whether to sleep. If the decision is to sleep, the server will sleep immediately, without incurring an idling period like HT. Its second moment of $I_{ii}$ under BS is thus smaller than that under HT.

CS has an AoI smaller than BS and HT because it is designed to reduce the second moment of $I_{ii}$. The server under CS will remain idling if the packet processing time within this regenerative cycle turns out to be large. It only sleeps when the processing time is short. This way can prevent $I_{ii}$ from being either too large or too small, thus reducing the second moment of $I_{ii}$. In Section VII we will show numerically that this result also holds in scenarios with multiple data sources.

V. OPTIMAL SLEEP-WAKE STRATEGY

Since CS has the advantage over BS and HT in achieving a smaller AoI, we fix CS as the idling scheme in this section. We aim to derive the threshold $b_i$ and parameter $N$ to achieve the minimum energy consumption rate while guaranteeing information freshness for users. For notation simplicity, we let $\theta_i$ denote the sleeping probability under CS in this section. We will first discuss the conditions under which the Aol-energy tradeoff and PAoI-energy tradeoff exist in Section V-A and then discuss the optimal sleep-wake design in Section V-B.

A. Age-Energy Tradeoff

In this subsection, we will first introduce two useful corollaries and then characterize the difference in age-energy tradeoff when using Aol and PAoI as metrics to measure information freshness.

**Corollary 2.** For a fixed $N$, $E[A_i^{CS}]$ is an increasing function of $\theta_i$ for any $i$. For fixed $\theta_i$ and $\min_i\{\theta_i\} > 0$, $E[A_i^{CS}]$ is an increasing function of $N$.

Corollary 2 holds because Equation (22) is a linear function of both $\theta_i$ and $N$. It shows that increasing the probability of sleeping, and increasing the sleeping period length would increase the PAoI. Note that Corollary 2 holds for HT and BS as well, since HT, BS, and CS have the same expressions for PAoI. In the next corollary, we characterize how the energy consumption rate changes as a function of $\theta_i$ and $N$.

**Corollary 3.** When $P_{SL} < \min\{P_{TD}, P_{ST}, P_B\}$ and $\theta = (\theta_1, \ldots, \theta_k)$ is fixed with $\min_i\{\theta_i\} > 0$, then $E[P^{CS}]$ is decreasing on $N$. When $N$ is fixed, the minimal $E[P^{CS}]$ is reached at either $\theta_i = 0$ or $\theta_i = 1$ for each $i$. If $P_{ST} \leq P_{TD}$, then the minimal $E[P^{CS}]$ is achieved at $\theta_i = 1$ for all $i \in \{1, \ldots, k\}$.

**Proof:** See Appendix C of the supplementary material for the detailed proof.

Corollary 3 shows that increasing the sleeping period length can always reduce the energy consumption, since the sleeping mode has the lowest energy consumption rate than the other statuses of the server. However, increasing the sleeping probabilities does not always reduce energy consumption for a fixed $N$. Since each sleeping period is followed by a setup period, if the energy consumption for the setup period is high and the sleeping period is short, then the energy saved in the sleeping period could be offset by the energy consumed in the setup period.

Following from Corollaries 2 and 3 we can characterize the PAoI-energy tradeoff in the following way: When the server’s sleeping probabilities are fixed with $\min_i\{\theta_i\} > 0$, then increasing the sleeping length (i.e., $N$) would increase the PAoI, but decrease the energy consumption. When the sleeping length (i.e., $N$) is fixed, then increasing the sleeping probability $\theta_i$ would increase PAoI. The energy consumption rate is either increasing or decreasing as $\theta_i$ increases.

The Aol-energy tradeoff is difficult to characterize due to the closed-form expression for Aol being involved. For analysis purposes, we now focus on the single data source scenario. We remove the subscript of the variables for notation simplicity. We can rewrite Aol under CS as

$$E[\Delta^{CS}] = \frac{\eta + \beta N + \frac{\theta}{\lambda} N^2}{2(\gamma + \frac{\theta}{\lambda} N)} + E[H] + \frac{\gamma(1-U^*(\lambda))}{\lambda}, \quad (28)$$

with

$$\eta = H^{*(2)}(0) + 2 \left[ E[H] + H^{*(1)}(b) \right] \frac{1}{\lambda} + (1-\theta) \frac{2}{\lambda^2} - 2H^{*(1)}(b)E[U] + \theta E[U^2], \quad (29)$$

$$\beta = -2H^{*(1)}(b) \frac{1}{\lambda} + 2\theta E[U] + \frac{\theta}{\lambda^2}, \quad (30)$$

and

$$\gamma = E[H] + \frac{1}{\lambda}(1-\theta) + \theta E[U]. \quad (31)$$

We then have

$$\frac{\partial E[\Delta^{CS}]}{\partial N} = \frac{\beta \gamma + \frac{\theta}{\lambda} N \gamma + \frac{\theta^2}{\lambda^2} N^2 - \frac{\eta}{\lambda}}{2(1+\frac{\theta}{\lambda} N)} \quad (32)$$

The solution to $\frac{\partial E[\Delta^{CS}]}{\partial N} = 0$ is given by $N^* = 2\xi\gamma \sqrt{1 - \frac{\theta}{\lambda} \frac{\gamma}{\beta} - \eta^2} - 2\xi^2\gamma$. From the facts that $\frac{\partial E[\Delta^{CS}]}{\partial N}|_{N=0} = 0$ and $\frac{\partial E[\Delta^{CS}]}{\partial N}|_{N=\infty} > 0$, one can conclude that $N^*$ is the minimizer for $E[\Delta^{CS}]$ if $N^* > 0$. As $N$ increases from 0 to $\infty$, $E[\Delta^{CS}]$ decreases when $N \leq N^*$ and increases when $N > N^*$. This implies that increasing the sleeping length (i.e., $N$) does not always increase Aol. Note that $N^*$ is not always positive. For $N^*$ to be positive, we need $\beta \gamma - \eta^2 \leq 0$, which means

$$2E[H e^{-b_H}](E[H] + \frac{1}{\lambda}) + 2\theta E[U]E[H] + (2-\theta)\frac{\eta}{\lambda} E[U] + 2\theta^2 \{E[U]^2\} \leq \theta H^{*(2)}(0) + \frac{\theta E[H]}{\lambda} + \frac{\theta(1-\theta)}{\lambda^2} + \theta^2 E[U^2]. \quad (33)$$

Inequality (33) thus provides a sufficient condition under which the Aol is not a monotone function of $N$. 

When \( b = 0 \) for CS, then Inequality \( (33) \) becomes
\[
2\{E[H]\}^2 + \frac{E[H]}{\lambda} + 2E[U]E[H] + \frac{E[U]}{\lambda} + 2\{E[U]\}^2 \leq E[H^2] + E[U^2]. \tag{34}
\]
By letting \( D = 0 \) for HT and \( \theta^{BS} = 1 \) for BS we can also have Inequality \( (34) \). One can easily show that Inequality \( (34) \) does not hold if the coefficients of variation (CV) of \( H \) and \( U \) are both smaller than 1 (i.e., \( \sqrt{Var[H]/E[H]} < 1 \) and \( \sqrt{Var[U]/E[U]} < 1 \)). In such a case, \( N^* \) is a negative number, so increasing \( N \) can increase AoI. Therefore, whether enlarging the sleeping period length would increase AoI depends on the CV of service and setup time distributions. When the CV of \( H \) and \( U \) are large, it is possible that Inequality \( (34) \) will hold. We will show it numerically in Section \( \text{VII} \).

When \( N \) is fixed, the change of \( E[\Delta^{CS}] \) as a function of \( \theta \) is difficult to be discussed analytically. We will also show it numerically in Section \( \text{VII} \).

B. Optimal Sleeping Probability and Sleeping Length

1) Optimization Problem: We now search for the optimal sleeping probability and sleeping length by considering an optimization problem with the objective to minimize energy consumption and the constraint to keep either PAoI or AoI bounded by a threshold.

Under CS, the energy consumption rate \( E[P^{CS}(N, b)] \), PAoI \( E[A^{CS}(N, b)] \), and AoI \( E[\Delta^{CS}(N, b)] \) are determined by the parameter \( N \) and vector \( b \in \{b_1, \ldots, b_k\} \). When using PAoI as the information freshness metric, we have the following problem:

\[
P1 : \min_{N, \theta} E[P^{CS}(N, b)] \tag{35}
\]
\[
\text{s.t. } E[A^{CS}(N, b)] \leq \tau \tag{36}
\]
\[
b_i \geq 0 \text{ for } i \in \{1, \ldots, k\} \tag{37}
\]
\[
N \in N^+ \tag{38}
\]

The parameter \( \tau \) in \( P1 \) is the PAoI requirement for user \( i \). We can also define a problem \( P2 \) by substituting the PAoI constraint in \( P1 \) by \( E[\Delta^{CS}(N, b)] \leq \tau_i \), with \( \tau_i \) being the AoI constraint for user \( i \).

We hope to characterize the difference in optimizing \( P1 \) and \( P2 \), so we begin by considering a single-source problem to develop insights. Again, we remove the index of random variables in the single-source scenario for notation simplicity. We can rewrite the energy consumption rate \( E[P^{CS}(N, \theta)] \) and PAoI \( E[A^{CS}(N, \theta)] \) as functions of the variable \( N \) and sleeping probability \( \theta \). If we relax the variable \( N \) as a real number, then problem \( P1 \) can be relaxed as problem \( P3 \), shown as follows.

\[
P3 : \min_{N, \theta} \ E[P^{CS}(N, \theta)] \tag{39}
\]
\[
\text{s.t. } E[A^{CS}(N, \theta)] \leq \tau \tag{40}
\]
\[
0 \leq \theta \leq 1 \tag{41}
\]
\[
N \geq 1. \tag{42}
\]

Letting \( N \) be a real number also has practical implications. Suppose \( N \geq 1 \) is a real-number, one can achieve \( E[A^{CS}(N, \theta)] \) and \( E[P^{CS}(N, \theta)] \) in the following way: Let \( \alpha = N-\lfloor N \rfloor \) such that the server with probability \( 1-\alpha \) wakes up when accumulating \( \lfloor N \rfloor \) packets, and with probability \( \alpha \) wakes up when accumulating \( \lfloor N \rfloor + 1 \) packets. The optimal solution to \( P3 \) thus can be achieved in practice, and the optimal value of \( \alpha \) is clearly a lower bound of that of \( P1 \) with a single data source.

2) Optimal Solution Characterization: As \( P3 \) differs from \( P1 \) (with a single data source) only in the integer constraint for \( N \), we aim to use the properties of \( P3 \) to understand the optimal solution to \( P1 \). In the following theorem, we characterize the optimal solution to \( P3 \).

Theorem 4. The optimal \( N \) and \( \theta \) for the optimization problem \( P3 \) belong to one of the three types: (Type 1) \( \theta = \frac{\tau - \frac{1}{2} - 2E[H]}{\frac{1}{2} + E[U]} > 0 \) with \( N \to \infty \); (Type 2) \( \theta = 1 \) with \( N = \lambda(\tau - \frac{1}{2} - 2E[H]) - E[U] + \frac{U^*}{\lambda} \); or (Type 3) \( \theta = \frac{\tau - \frac{1}{2} - 2E[H]}{\frac{1}{2} + E[U]} - \frac{U^*}{\lambda} \) with \( N = 1 \).

Proof: See Appendix D of the supplementary material for the detailed proof.

Type 1 solution in Theorem 4 means the server should sleep with a tiny but positive probability. Whenever the server sleeps, it sleeps for infinitely long. Type 2 solution means that the server sleeps with probability 1, and the sleeping period is determined by a number \( N \) greater than 1. Under the Type 3 solution, the server wakes up whenever a packet arrives during the sleeping period, and the server sleeps with a probability between 0 and 1.

Note that the Type 1 solution shown in Theorem 4 is problematic from two perspectives. First, the feasible region of \( P3 \) is not a closed set. Type 1 solution is located infinitely close to the boundary, but it cannot locate at the boundary. It requires \( N \) to be infinitely large, but \( \theta = \frac{\tau - \frac{1}{2} - 2E[H]}{\frac{1}{2} + E[U]} > 0 \) to be a positive number. Second, this solution is difficult to implement in practice. Under this scheme, the server barely sleeps, but it sleeps for an infinitely long period whenever it sleeps. No packet is processed during the sleeping period, so the data receiver’s temporary age \( \Delta(t) \) will become infinitely large.

As Type 1 solution is an asymptotic solution, when \( N \to \infty \), the energy consumption rate converges to
\[
E[P^{CS}_{P1}] \approx \frac{P_B E[H] + \frac{1}{2} P_{ID} + (\tau - \frac{1}{2} - 2E[H])P_{SL}}{\tau - E[H]}, \tag{43}
\]
which is an energy consumption rate unrelated to \( P_{ST} \). This shows as long as the sleeping period length is large enough and sleeping probability is small enough, the effect of \( P_{ST} \) on the energy consumption rate becomes negligible.

3) Occurrence of Different Types of Optimal Solution: We now discuss when different types of optimal solution characterized in Section \( \text{VIII} \) would occur. Fig. 2 provides a numerical study to show when different types of optimal solutions to \( P3 \) occur as \( E[U] \) and \( P_{ST} \) change. We can see...
from Fig. 2 that Type 1 solution occurs when $E[U] = \frac{1}{\lambda}$ and $P_{ST}$ are both large, as applying Type 1 solution can effectively avoid setup. When $E[U] = \frac{1}{\lambda}$ is large and $P_{ST}$ is small, we have Type 3 optimal solution. When $E[U] = \frac{1}{\lambda}$ and $P_{ST}$ are both small, we have Type 2 optimal solution.

We can also prove rigorously that when $U = 0$ (i.e., the setup time does not exist), Type 1 solution will not occur. When $\tau \geq \frac{1}{\lambda} + 2E[H]$, one can easily verify that Type 3 solution does not exist. The energy consumption rate for Type 2 solution is given by

$$E[P_{CS}^{Type2}[U = 0]] \triangleq \frac{P_{B}E[H] + (\tau - \frac{1}{\lambda} - 2E[H])P_{SL}}{\tau - E[H]}.$$  

(44)

We can easily verify that $E[P_{CS}^{Type2}[U = 0]] \leq E[P_{CS}^{Type1}]$, which means Type 2 solution will occur. When $\tau < \frac{1}{\lambda} + 2E[H]$. Type 2 solution does not exist. We can derive the energy consumption rate for Type 3 solution as

$$E[P_{CS}^{Type3}[U = 0]] \triangleq \left\{ \left( \frac{1}{\lambda} + 2E[H]\right)\left(P_{SL} - P_{ID}\right) + P_{B}E[H] + \frac{1}{\lambda}P_{ID}\right\}/(E[H] + \frac{1}{\lambda}).$$

(45)

One can also verify that $E[P_{CS}^{Type3}[U = 0]] \leq E[P_{CS}^{Type1}]$. Hence when there is no setup time, Type 1 solution does not occur.

Since $P_{1}$ and $P_{3}$ differ only in the integer constraint on $N$, we can infer that Type 1 solution exists in $P_{1}$ as well. Although Type 1 solution can effectively reduce the energy consumed in setup, Type 1 solution is difficult to implement in practice. Moreover, it would incur a large AoI. One can easily verify from Equations (26) and (27) that if $N \to \infty$, then $E[\frac{1}{\lambda}I_{ii}] \to \infty$. This fact also implies that if we use AoI as the metric to measure information freshness, the optimal $N$ in $P_{2}$ should be bounded. It also shows that although PAoI and AoI are both defined to measure information freshness, using them as information freshness metrics in optimization problems would result in distinct optimal solutions. The difference comes from PAoI being a first-order statistic, as shown in Equation (6). AoI is determined by the second moment of $I_{ii}$, as shown in Equation (7).

As $P_{1}$ and $P_{2}$ with multiple data sources are difficult to be solved analytically, we present a numerical study for $k = 2$ in Fig. 3. For each $N$ value in Fig. 3 we solve for the optimal $b = (b_{1}, b_{2})$ to achieve the minimal $E[P_{CS}]$. We find that when using AoI as the information freshness metric, the optimal $N$ is 11. However, when using PAoI as the information freshness metric, the optimal $N$ could be larger than 25. As $P_{ST}$ is greater than $P_{ID}$ and $P_{SL}$ in Fig. 3 we can infer that the optimal solution when using PAoI as the information freshness metric (i.e., $P_{1}$) is to avoid setup as much as possible. That is, the optimal $\theta$ could be very tiny, and the optimal $N$ could be very large, similar to the Type 1 solution in $P_{3}$.

VI. Extensive Discussions to Scenarios with Strategic Data Sources

In the previous sections, we discussed the sleep-wake strategies in the scenarios where the data generation rates of data sources are fixed. We now extend our discussion to the scenarios where the sampling process of each data source is associated with sampling costs, such as battery depletion. Specifically, we formulate the game between the server and data sources as a Stackelberg game in Section VI-A. We show that the equilibrium sampling rates for data sources can be characterized by solving an optimization in Section VI-B. We then analyze the equilibrium strategies of the server and data sources in Section VI-C.

A. Stackelberg Game Formulation

Strategic data sources may select their sampling rates according to the sleep-wake strategy of the server to guarantee the information freshness of the corresponding users and reduce their sampling cost at the same time. In such a scenario, how to design the sleep-wake strategy to reduce the server’s energy consumption while guaranteeing the users’ information freshness remains unknown.

We model the interaction between the server and data sources as a Stackelberg game, where the server decides the optimal sleep-wake strategy at Stage I, and the data sources determine their sampling rates accordingly at Stage II. In Stage I, the server’s decision is the sleeping probability vector.
\( \theta = (\theta_1, \ldots, \theta_k) \) and the sleep period length \( N \), supposing that it adopts CS as the idling scheme. The server’s objective is to minimize its energy consumption. Specifically, since our analysis shows that the closed-form expression for AoI as a function of \( \theta \), to avoid the Type 1 solution that we showed in Section V-B and also to better understand the game between data sources and the server, we let \( N \leq N_{\text{max}} \), where \( N_{\text{max}} \) is a positive integer. We will show later how \( N_{\text{max}} \) is determined. So Stage I problem is formulated as

**Stage I**: \( \min_{N, \theta} \mathbb{E}[P^{CS}(X^e, N, \theta)] \)

s.t. \( N \in \{1, \ldots, N_{\text{max}}\} \)

\( \mathbb{E}[A^{CS}_i(X^e, N, \theta)] \leq \tau_i \)

\( \theta_i \in [0, 1] \) for \( i \in \{1, \ldots, k\} \).

The parameter \( \tau_i \) in Stage I problem is the PAoI requirement by data receiver \( i \). The vector \( X^e = (\lambda^e_1, \ldots, \lambda^e_k) \) is the equilibrium sampling rate vector of data sources given in Stage II.

In Stage II, the values of \( N \) and \( \theta \) are fixed. We assume that each data source \( i \) has a non-decreasing sampling cost function \( c_i(\lambda_i) \), and the objective of each data source is to minimize its sampling cost. The PAoI requirement by each data receiver should be satisfied. Data source \( i \) can only determine its own sampling rate \( \lambda_i \), but the PAoI for receiver \( i \) is determined by \( \lambda_i \) and also the sampling rate of other data sources. The Stage II problem for data source \( i \) is then formulated as

**Stage II**: \( \min_{\lambda_i} c_i(\lambda_i) \)

s.t. \( E[A^{CS}_i(X^e, N, \theta)] \leq \tau_i \)

\( 0 \leq \lambda_i \leq \lambda_{\text{max}} \).

**B. Equilibrium as an Optimization Problem**

We now show that the equilibrium strategy for Stage II (Equations (50)-(52)) can be obtained by solving the following system:

**P4**: \( \min_\lambda \sum_{i=1}^k c_i(\lambda_i) \)

s.t. \( E[A^{CS}_i(\lambda, N, \theta)] \leq \tau_i \)

\( 0 \leq \lambda_i \leq \lambda_{\text{max}} \) for \( i = 1, \ldots, k \).

The reason for Problem P4 being equivalent to Stage II problem is as follows. As \( X^e \) is the equilibrium sampling rate for Stage II, if we replace \( \lambda^e_i \) in \( X^e \) with any \( \lambda_i \in [0, \lambda_{\text{max}}] \), then we would have either \( E[A^{CS}_i(\lambda^e, N, \theta)] > \tau_j \) for some \( j \) or \( c_i(\lambda_i) \geq c_i(\lambda^e_i) \), which shows that \( (\lambda^e_1, \ldots, \lambda^e_{i-1}, \lambda^e_{i+1}, \ldots, \lambda^e_k) \) is not the optimal solution to P4. One can also use a similar argument to show that the optimal solution to P4 is the equilibrium sampling rate to Stage II. Therefore, one can solve P4 to obtain the equilibrium sampling rate for Stage II.

**C. Equilibrium Analysis**

In this subsection, we will first introduce some useful results for the optimal problem P4, and then characterize the equilibrium strategies of the server and data sources based on these results. In the following lemma, we show that the PAoI for each data source decreases as the sampling rate increases.

**Lemma 5. When \( \theta \) and \( N \) are fixed, \( E[A^{CS}_i(\lambda, N, \theta)] \) is a decreasing function of \( \lambda_i \).**

**Proof:** See Appendix E of the supplementary material for the detailed proof.

The reason for Lemma 5 to hold is that the increment of \( \lambda_i \) would reduce the server’s sleeping length, which increases the frequency that the server processes packets from source \( i \).

In many queueing systems, the increment of packet generation rate of a data source would enlarge the PAoI of other sources due to the system congestion. As a result, the server would be more likely to process the packets with large arrival rates. This phenomenon occurs in many queueing systems such as LCFS queue with multiple classes [29] and priority queue systems [3]. However, in our system with a sleep-wake server, increasing the packet generation rate for data source \( i \) does not always increase the PAoI for other data sources with \( j \neq i \). As shown in Fig. 4, increasing the data generation rate for Data Source 1 in our system reduces the PAoI for all the data sources. The reason is that in Fig. 4, we let the sleeping probability vector be \( \theta = (0, 1, 1, 1, 1) \). Enlarging \( \lambda_1 \) would make the server process packets from data source 1 more frequently. After processing a packet from data source 1, the server does not sleep. Thus increasing \( \lambda_1 \) would significantly lower the sleeping period and frequency, thereby reducing the PAoI for other data sources.

However, when \( \theta = (1, \ldots, 1) \), the following lemma shows that increasing \( \lambda_1 \) would increase the PAoI for data sources other than \( i \).

**Lemma 6. If \( \theta = (1, \ldots, 1) \), then \( E[A^{CS}_i(\lambda, N, \theta)] \) is increasing on \( \lambda_i \) with \( j \neq i \).**

**Proof:** See Appendix E of the supplementary material for the detailed proof.

Lemmas 5 and 6 characterize the monotonicity of PAoI as a function of \( \lambda \). Using the results from these two lemmas, we further characterize a property of the equilibrium sampling rate of Stage II in the following lemma. As the scenarios other than
\( \theta = (1, \ldots, 1) \) are complicated in analysis, we only consider the scenario with \( \theta = (1, \ldots, 1) \) to develop insights.

**Lemma 7.** \( \theta = (1, \ldots, 1) \), the equilibrium strategy \( \lambda^* \) for Stage II satisfies \( E[A_i^{CS}(\lambda^*, N, \theta)] = \tau_i \) for each \( i \).

**Proof:** See Appendix [G] of the supplementary material for the detailed proof.

Lemma 7 shows that if \( \lambda_i^* = \lambda_{\text{max}} \) for some \( i \), then \( \lambda^* \) is the solution of a overdetermined system that contains \( k + 1 \) equations. This implies that the solution to Problem (53) may not exist if we require \( \lambda_i^* = \lambda_{\text{max}} \) for some \( i \). Therefore, to guarantee that the equilibrium sampling rate \( \lambda^* \) always exists, we need \( \lambda_i^* < \lambda_{\text{max}} \) for all \( i \). When \( \theta = (1, \ldots, 1) \), by Lemma 7 we have that

\[
N = \lambda_i^* \left[ \tau_i - \frac{1 - U^* (||\lambda^*||_1)}{||\lambda^*||_1} - E[H_i] \right] - \sum_{l=1}^{k} \lambda_i^* \left( E[H_l] + E[U] \right) \quad \text{for } i \in \{1, \ldots, k\}. \quad (56)
\]

Let \( j \) be the index such that \( \tau_j - E[H_j] = \min_i \{ \tau_i - E[H_i] \} \). Letting \( ||\lambda^*||_1 = \sum_{i=1}^{k} \lambda_i \), we thus have

\[
N = \lambda_j^* \left[ \min_i \{ \tau_i - E[H_i] \} - E[H_j] - E[U] \right] - \sum_{l=1, l \neq j}^{k} \lambda_i^* \left( E[H_i] + E[U] \right) \leq \lambda_{\text{max}} \left[ \min_i \{ \tau_i - E[H_i] \} - E[H_j] - E[U] \right].
\]

We can therefore let

\[
N_{\text{max}} = \lambda_{\text{max}} \left[ \min_i \{ \tau_i - E[H_i] \} - E[H_j] - E[U] \right]. \quad (57)
\]

for Stage I problem with \( \theta = (1, \ldots, 1) \). Moreover, when \( \theta = (1, \ldots, 1) \), we propose Algorithm 1 to compute the optimal \( N \) in Stage I problem. In addition, using the lemmas above, we can prove the uniqueness of the equilibrium in Stage II once it exists, as shown in the following proposition.

**Proposition 8.** When \( \theta = (1, \ldots, 1) \) and \( \lambda^* \) is an equilibrium sampling rate in Stage II, then \( \lambda^* \) is the unique equilibrium for Stage II.

**Proof:** See Appendix [H] of the supplementary material for the detailed proof.

Proposition 8 implies that once \( \theta = (1, \ldots, 1) \) and \( N \) are fixed, then \( \lambda^* \) is uniquely determined. So that Algorithm 1 can find the optimal \( N \) in Stage I by enumeration. We will show numerically in Section VII how the equilibrium sampling rate and energy consumption rate change as \( N \) changes.

**VII. Numerical Study**

In this section, we first use the numerical study to show the advantage of CS in Section VII-A. We then numerically discuss the AoI-energy and PAoI-energy tradeoffs in Section VII-B. We will further compare our system with LCFS service discipline in Section VII-C. In Section VII-D we will discuss the strategic behavior of data sources as the sleep-wake strategy of the server varies.

---

**Algorithm 1 Equilibrium Computing**

1. Given the distribution of \( H_i \) and \( U_i \), also given the parameters \( P_B, P_{ID}, P_{SL}, P_{ST}, \tau = (\tau_1, \ldots, \tau_k) \).
2. Let \( N_{\text{candidate}} = 1, \lambda_{\text{candidate}} = 0, N_{\text{max}} = \lambda_{\text{max}} \).
3. for \( N = 1 \) to \( N_{\text{max}} \) do
4. Obtain \( \lambda \) by solving the system \( g(\lambda) = 0 \), where \( g_i(\lambda) = \frac{1 - U^* (\lambda)}{\lambda_i} - \lambda_i (E[H_i] - \tau_i) + \sum_{l=1}^{k} \lambda_l (E[H_l] + E[U]) + N \).
5. if \( \max(\lambda) \leq \lambda_{\text{max}} \) then
6. Compute \( P_{CS}(N) \)
7. if \( P_{CS}(N) < P_{CS}(N_{\text{candidate}}) \) then
8. \( N_{\text{candidate}} \leftarrow N \), \( \lambda_{\text{candidate}} \leftarrow \lambda \)
9. end if
10. end for
11. Return \( N_{\text{candidate}}, \lambda_{\text{candidate}} \)

---

**Fig. 5. Idling Scheme Comparison with \( \lambda = (0.8, 1.2) \), \( H_1 \equiv H_2 \sim \text{exp}(1), \theta_1 = \theta_2 = 0, N = 1, U \sim \text{Gamma}(2, 1) \)**

**A. Idling Scheme Comparison**

In Section IV, we showed that closed-form expressions of power consumption and PAoI under HT, BS, and CS are identical when these strategies have the same sleeping probabilities. However, these strategies achieve the same sleeping probabilities in different ways, making their AoI different. In Theorem 8 we showed that CS outperforms HT and BS in terms of minimizing AoI in the single data source scenario. In Fig. 5, we further show that this result also holds in the multiple-source scenario. In Fig. 5 we let \( D_i \) be a constant for all \( i = 1, \ldots, k \) in HT, so that \( D_i = -\ln \theta_i \). For CS, we let \( H_i \) be exponentially distributed with \( b_i = \frac{1}{\theta_i E[H_i]} \). From Fig. 5 we see that CS has a much smaller AoI than HT and BS when \( \theta_i \) is near 0.5. This is because when \( \theta_i = 0 \) for all \( i \), then the server never sleeps under those three strategies; when \( \theta_i = 1 \) for all \( i \), then the server always sleeps after processing a packet. In these two extreme scenarios, the three idling strategies perform identically. When \( \theta_i = 0.4 \), CS outperforms HT by nearly 10% for data source 1, and 7% for data source 2, which shows the advantage of applying CS.
(c) Multiple-sleep Scheme

(b) Single-sleep Scheme

(a) N-policy

(b) Single-sleep Scheme

(c) Multiple-sleep Scheme

Fig. 7. AoI Under CS with Different Service and Setup Time Distributions. \( \lambda = 0.8, \mathbb{E}[H] = 5, \mathbb{E}[U] = 5. \)

B. Tradeoff between Information Freshness and Energy Consumption

In this subsection, we numerically demonstrate the tradeoff between the information freshness and energy consumption rate for a single data source.

In Fig. 6 we plot the energy consumption rate, PAoI, and AoI as functions of the variable \( b \) and \( N \). Specifically, we let \( H \sim \text{Gamma}(\frac{1}{2}, 4), U \sim \text{Gamma}(\frac{1}{2}, 25), P_B = 2.1, P_{ID} = 1.1, P_{SL} = 0.1, P_{ST} = 2.1 \). We can develop several insights from Fig. 6.

First, we observe from Fig. 6(a) that the energy consumption rate is always a decreasing function of \( N \), which means sleeping for a long time would reduce the average energy consumption. Second, from Fig. 6(a) we observe that the energy consumption is not always a decreasing or increasing function of the threshold \( b \). Under CS, the sleeping probability is \( \theta = H(b) \), so the larger \( b \) becomes, the smaller \( \theta \) is. When \( N \) is small, the energy consumption rate decreases on \( b \), which means a smaller sleeping probability would reduce the energy consumption. This is because the setup period also has a high energy consumption rate, and sleeping for a short period cannot offset the power consumed in the setup. In this case, sleeping less could reduce the server’s energy consumption. When \( N \) is large, the averaged energy consumption during sleep and setup periods thus becomes small. Sleeping more frequently thus becomes beneficial for the server when \( N \) is large. Third, we observe from Fig. 6(b) and Fig. 6(c) that the shapes of the PAoI and AoI functions are distinct. The PAoI declines if \( N \) decreases or \( b \) increases, meaning that sleeping for a shorter period or less frequently can reduce PAoI. However, as shown in Fig. 6(c), enlarging the sleeping period does not always increase the AoI.

In Section V-A we derived the conditions for the non-monotonicity of the AoI function under N-policy. We now numerically show that this non-monotonicity is not due to using the N-policy as the wake-up scheme. In Fig. 7(a) we plot the AoI under N-policy with \( b = 0 \). It can be observed that AoI is an increasing function of \( N \) when \( H \) and \( U \) are both constants (\( CV = 0 \)), exponential (\( CV = 1 \)), and uniform (\( CV = \sqrt{3} \)). When \( H \sim \text{Gamma}(\frac{1}{2}, \mathbb{E}[H])^2 \) and \( U \sim \text{Gamma}(\frac{1}{2}, \mathbb{E}[U])^2 \), we have \( CV \) for \( H \) is \( \mathbb{E}[H]^2 \) and \( CV \) for \( U \) is \( \mathbb{E}[U]^2 \). One can easily verify that Inequality (34) holds, so the optimal \( N^* \) to achieve the lowest AoI is positive. As shown in Fig. 7(a), the optimal \( N^* = 3 \) in this case.

In Fig. 7(b) and 7(c) we plot the AoI under single-sleep scheme and multiple-sleep scheme respectively. The closed-form expressions for information freshness and energy consumption rate are provided in Appendix A of the supplementary material. In Fig. 7 we let the sleeping period \( W \) be exponential for both single-sleep and multiple-sleep schemes. The sleeping span within each regenerative cycle under single-sleep scheme is thus \( \mathbb{E}[W] \), and that under multiple-sleep scheme is \( \mathbb{E}[W] + \frac{1}{\lambda} \). From Fig. 7(b) and 7(c),
we see that the AoI is not monotone on the value of $E[W]$ when both $H$ and $U$ are gamma distributed. This observation means that increasing sleeping length with regenerative cycle does not always increase AoI, and this phenomenon is not unique for using N-policy as the wakeup scheme.

C. Comparison between LCFS and Single Buffer Systems

The PAoI for a single source system with LCFS service discipline, HT as the idling scheme, and N-policy as the sleeping scheme was investigated in [26]. However, the PAoI-energy tradeoff was not discussed in [26]. We now want to answer whether using a single buffer outperforms LCFS in achieving a better PAoI-energy tradeoff. To make a fair comparison, we consider a single-source LCFS system with CS as the idling scheme and N-policy as the sleeping scheme. Based on the analysis in [26] and our discussion in Section IV, we can derive its PAoI as

$$E[A_{LCFS}^{CS}] = E[H] + \frac{\theta(1 - \lambda E[H])(1 - U^*(\lambda))}{1 + \theta(N - 1 + \lambda E[U])} + 2 - H^*(\lambda) + \lambda H^{*(1)}(\lambda) / \lambda [H - H^*(\lambda)]$$

and the energy consumption rate as

$$E[P_{LCFS}^{CS}] = \lambda E[H]P_B + \frac{(1 - \lambda E[H])((1 - \theta)P_{ID})}{\lambda} + \theta \frac{N P_{SL} + \theta E[U]P_{ST}}{1 + \theta(N + E[U])}.$$  \hspace{1cm} (58)

We compare the minimal energy consumption rate under LCFS and single buffer when both of their PAoI is constrained by the same constant in Fig. 8. We observe from Fig. 8 that the energy consumption under the single buffer system is much lower than LCFS when the packet generation rate is large. The reason is that under LCFS, the server will have to process all the arrived data packets. As we see from Equation (58), enlarging the sleeping period and probability does not help the server reduce the energy consumed in processing packets (i.e., $\lambda E[H]P_B$). Especially, when the traffic intensity (i.e., $\lambda E[H]$) is large, we can find from Equations (58) and (59) that by altering the value of $\theta$ and $N$ under LCFS will not change PAoI and energy consumption significantly. This observation indicates when the traffic intensity is large, using the single buffer strategy can achieve a better PAoI-energy tradeoff.

D. Game between the Server and Data Sources

We now discuss the case where data sources strategically choose their sampling rates according to the sleep-wake strategy of the server. We aim to understand whether enlarging the sleeping period for the server can reduce the server’s energy consumption and the sampling cost for data sources. We fix the sleeping probability $\theta = 1$ in Fig. 9 and apply Algorithm I to compute the equilibrium packet generation rates and energy consumption rate under each value of $N$. Fig. 3(a) shows that as the value of $N$ in Stage I increases, the equilibrium packet generation rates in Stage II would increase accordingly. The energy consumption rate in Stage I would decrease as $N$ increases, as shown in Fig. 3(b). However, we notice that the reduction in the energy consumption rate in Stage I is not significant as $N$ increases. The difference in the energy consumption rate between $N = 1$ and $N = 30$ is less than 0.1%, when the data sources are strategic. At the same time, the equilibrium data generation rate for each data source would increase significantly as $N$ increases, which would result in a high sampling cost if $N$ is large.

It is important to note that although the energy reduction for the server seems to be tiny, in reality, the energy consumption of servers and base stations could be much more significant than the energy consumed by IoT sampling devices (data sources). So the 0.1% reduction of energy consumption rate could be significant for the servers and base stations. On the other hand, the sampling cost is also crucial for data sources. Most of the IoT sampling devices such as smartphones and UAVs have finite batteries, so a high sampling rate is costly for those devices. The social welfare can be written as a weighted combination of the data sources’ sampling cost and the server’s energy consumption. In such a case, we can use Algorithm I to find the optimal $N$ to achieve the social optimum by replacing the objective function in Stage I as the social welfare. From the monotonicity of the sampling cost, we can infer that the social optimal $N$ cannot be greater than the optimal $N$ for Stage I. This result indicates that keeping a relatively short sleeping length for the server could lead to the socially optimal solution for the server and data sources altogether.

VIII. Conclusion and Future Research

In this paper, we investigated the information freshness and energy consumption in a single server queueing system where the server could sleep to reduce its energy consumption. We proposed a modeling approach that relies on a renewal type argument to derive the closed-form expressions for information metrics (i.e., PAoI and AoI) and energy consumption rate. Specifically, we proposed an idling scheme called Conditional
Sleep (CS) scheme to achieve the same PAoI and energy consumption rate as two widely used strategies (namely Hysteresis Time (HT) scheme and Bernoulli Sleep (BS) scheme) while achieving a smaller AoI than the other two policies.

We found that extending the server’s sleeping period length can reduce the energy consumption and enlarge the PAoI, but not always increase the AoI. We derived the conditions under which increasing sleeping period length does not increase AoI. We showed that this counter-intuitive phenomenon occurs when the packet processing time and setup time distributions have a large coefficient of variation. Our analysis further shows that optimizing the energy consumption under a PAoI constraint can result in an optimal solution that is difficult to implement in practice, but this issue does not exist when using AoI in the information freshness constraint. We then extended our discussion to the scenario where data sources strategically choose their packet generation rates according to the sleep-wake strategy of the server. Our analysis shows that enlarging the sleeping period will slightly reduce the server’s energy consumption, while significantly raising data sources’ sampling costs.

We will study the joint optimization for data sources and the server in our future research. Moreover, we hope to extend our discussion to the scenario with multiple servers coordinating sleep-wake strategies.

**REFERENCES**

[1] I. Kadota, A. Sinha, and E. Modiano, “Optimizing age of information in wireless networks with throughput constraints,” in IEEE INFOCOM 2018-IEEE Conference on Computer Communications, IEEE, 2018, pp. 1844–1852.

[2] Y. Sun, I. Kadota, R. Talak, and E. Modiano, “Age of information: A new metric for information freshness,” Synthesis Lectures on Communication Networks, vol. 12, no. 2, pp. 1–224, 2019.

[3] J. Xu and N. Gautam, “Peak age of information in priority queuing systems,” IEEE Transactions on Information Theory, vol. 67, no. 1, pp. 373–390, 2021.

[4] F. Richter, A. J. Fehske, and G. P. Fettweis, “Energy efficiency aspects of base station deployment strategies for cellular networks,” in 2009 IEEE 70th Vehicular Technology Conference Fall. IEEE, 2009, pp. 1–5.

[5] Z. Niu, X. Guo, S. Zhou, and P. R. Kumar, “Characterizing energy–delay tradeoff in hyper-cellular networks with base station sleeping control,” IEEE Journal on Selected Areas in Communications, vol. 33, no. 4, pp. 641–650, 2015.

[6] C. Liu, B. Natarajan, and H. Xia, “Small cell base station sleep strategies for energy efficiency,” IEEE Transactions on Vehicular Technology, vol. 65, no. 3, pp. 1652–1661, 2015.

[7] L. Pei, J. Huilin, P. Zhiwen, and Y. Xiaohu, “Energy-delay tradeoff in ultra-dense networks considering bs sleeping and cell association,” IEEE Transactions on Vehicular Technology, vol. 67, no. 1, pp. 734–751, 2017.

[8] S. Verma, Y. Kawamoto, and N. Kato, “Energy-efficient group paging mechanism for qos constrained mobile iot devices over lte-a pro networks under 5g,” IEEE Internet of Things Journal, vol. 6, no. 5, pp. 9187–9199, 2019.

[9] H. Wang, M. Huang, Z. Zhao, Z. Guo, Z. Wang, and M. Li, “Base station wake-up strategy in cellular networks with hybrid energy supplies for 6g networks in an iot environment,” IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5230–5239, 2020.

[10] A. S. Iquebal and S. T. Bukkapatnam, “Consistent estimation of the max-flow problem: Towards unsupervised image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, early access.

[11] 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.

[12] Y. Inoue, H. Masuyama, T. Takine, and T. Tanaka, “A general formula for the stationary distribution of the age of information and its application to single-server queues,” IEEE Transactions on Information Theory, vol. 65, no. 12, pp. 8305–8324, 2019.

[13] R. D. Yates and S. K. Kaul, “The age of information: Real-time status updating by multiple sources,” IEEE Transactions on Information Theory, vol. 65, no. 3, pp. 1807–1827, 2019.

[14] A. Maatouk, M. Assaad, and A. Ephremides, “The age of updates in a simple relay network,” in 2018 IEEE Information Theory Workshop (ITW). IEEE, 2018, pp. 1–5.

[15] X. Guo, Z. Niu, S. Zhou, and P. Kumar, “Delay-constrained energy-optimal base station sleeping control,” IEEE Journal on Selected Areas in Communications, vol. 34, no. 5, pp. 1073–1085, 2016.

[16] J. Wu, Y. Bao, G. Miao, S. Zhou, and Z. Niu, “Base-station sleeping control and power matching for energy–delay tradeoffs with bursty traffic,” IEEE Transactions on Vehicular Technology, vol. 65, no. 5, pp. 3657–3675, 2015.

[17] O. Onireti, A. Mohamed, H. Pervazia, and M. Imran, “Analytical approach to base station sleep mode power consumption and sleep depth,” in 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE, 2017, pp. 1–7.

[18] M. Feng, S. Mao, and T. Jiang, “Boost: Base station on-off switching strategy for green massive mimo heatnets,” IEEE Transactions on Wireless Communications, vol. 16, no. 11, pp. 7319–7332, 2017.

[19] ———, “Base station on-off switching in 5g wireless networks: Approaches and challenges,” IEEE Wireless Communications, vol. 24, no. 4, pp. 46–54, 2017.

[20] J.-C. Ke, “The optimal control of an m/gl queueing system with server vacations, startup and breakdowns,” Computers & Industrial Engineering, vol. 44, no. 4, pp. 567–579, 2003.

[21] W. Liu, Y. Ma, and J. Li, “Equilibrium threshold strategies in observable queueing systems under single vacation policy,” Applied Mathematical Modelling, vol. 36, no. 12, pp. 6186–6202, 2012.

[22] J. Wu and Z. Lian, “A single-server retrial g-queue with priority and unreliable server under bernoulli vacation schedule,” Computers & Industrial Engineering, vol. 64, no. 1, pp. 84–93, 2013.
[23] F. A. Maraghi, K. C. Madan, and K. Darby-Dowman, “Batch arrival queueing system with random breakdowns and bernoulli schedule server vacations having general vacation time distribution,” International Journal of Information and Management Sciences, vol. 20, no. 1, pp. 55–70, 2009.

[24] A. M. Bedewy, Y. Sun, R. Singh, and N. B. Shroff, “Optimizing information freshness using low-power status updates via sleep-wake scheduling,” in Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, 2020, pp. 51–60.

[25] H. Huang, D. Qiao, and M. C. Gursoy, “Age-energy tradeoff optimization for packet delivery in fading channels,” IEEE Transactions on Wireless Communications, 2021.

[26] J. Xu and Y. Chen, “Information freshness in sleep-wake server systems,” IEEE Communications Letters, vol. 25, no. 7, pp. 2186–2190, 2021.

[27] V. G. Kulkarni, Modeling and analysis of stochastic systems. Chapman and Hall/CRC, 2016.

[28] J. Xu, I.-H. Hou, and N. Gautam, “Age of information for single buffer systems with vacation server,” IEEE Transactions on Network Science and Engineering, 2021, early access.

[29] S. Hao and L. Duan, “Regulating competition in age of information under network externalities,” IEEE Journal on Selected Areas in Communications, vol. 38, no. 4, pp. 697–710, 2020.

[30] E. L. Lehmann, “Some concepts of dependence,” The Annals of Mathematical Statistics, pp. 1137–1153, 1966.

[31] J. Sherman and W. J. Morrison, “Adjustment of an inverse matrix corresponding to a change in one element of a given matrix,” The Annals of Mathematical Statistics, vol. 21, no. 1, pp. 124–127, 1950.
Supplementary Material for the Paper "Balancing Age-Energy Tradeoff in Sleep-Wake Server Systems"

APPENDIX A

PERFORMANCE METRICS FOR SINGLE-SLEEP AND MULTIPLE-SLEEP STRATEGIES

Since Theorem 1 shows that CS has better performance in minimizing AoI than HT and BS, we here only introduce the performance metrics for single-sleep and multiple-sleep strategies under CS.

A. Single-sleep

Under the single-sleep scheme, the server sleeps for a random period $W$, and then sets up. If no arrival occurs during the sleeping period nor the setup period, the server will stay idling after setting up until a new arrival occurs. If there is a packet waiting in the buffer when the setup period is over, the server will stay idle after setting up until a new arrival occurs. If there is no arrival during the sleeping period and the setup period, the server will stay idling after setting up until a new arrival occurs.

The power consumption rate can be derived by considering the events within each regenerative cycle. Since the probability that no arrival occurs during the setup period is

$$E^1[P_{CS}] = \left\{ \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} \left[ P_B E[H_i] + \frac{1 - \theta_i}{\lambda} P_{ID} + \theta_i E[W] P_{SL} + \frac{W^*(\lambda)U^*(\lambda)}{\lambda} P_{ST} \right] \right\} \left\{ \sum_{i=1}^{k} \frac{\lambda_i}{\lambda} E[H_i] + \theta_i E[W] + E[U] + \frac{W^*(\lambda)U^*(\lambda)}{\lambda} \right\}.$$ 

Notice that $P_{SL}$ under single-sleep could be lower than that under N-policy, since under single-sleep, the server does not incur a cost for detecting and counting the number of arrivals during the sleeping period (see [15]).

To derive the AoI and PAoI, we only need to characterize the first time a packet arrives. Since Theorem 1 shows that CS has better performance in minimizing AoI than HT and BS, we here only introduce the derivation for single-sleep scheme in [26], we have

$$E[G_i] = \sum_{j=1}^{k} \frac{\lambda_j}{\lambda} \left( \frac{1 - W^*(\lambda)U^*(\lambda)}{\lambda} + W^*(\lambda)U^*(\lambda) + W^*(\lambda)U^*(\lambda) \right).$$

Then we can compute the PAoI and AoI using Equations (6) and (7). We do not present the closed-form expressions of PAoI and AoI here as they are involved.

B. Multiple-sleep

Under the multiple-sleep scheme, the server returns from a vacation period $W$ and finds the buffer non-empty, then the server wakes up; otherwise, another vacation period $W$ is taken. The energy consumption rate $P_{SL}$ under multiple-sleep scheme is also lower than that under N-policy, as the multiple-sleep scheme does not need to count the arrival packets. However, there may exist a detection cost $P_{DT}$ whenever the server returns from a sleeping period to detect whether packets are waiting (see [15]).

From [26], [28] we know that the averaged sleeping length under multiple-sleep scheme is $E[W]/(1 - W^*(\lambda))$, and the number of sleeping periods the server has before setting up is $1/(1 - W^*(\lambda))$.

We now introduce the way to calculate $E[G_i]$. Since an informative packet only has to wait if it arrives when the server is sleeping or setting up and the probability that the server experiences a sleeping/setup period is $\sum_{i=1}^{k} \frac{\lambda_i}{\lambda}$, using the derivation for single-sleep scheme in [26], we have

$$E[G_i] = \sum_{j=1}^{k} \frac{\lambda_j}{\lambda} \left( \frac{1 - W^*(\lambda)U^*(\lambda)}{\lambda} + W^*(\lambda)U^*(\lambda) + W^*(\lambda)U^*(\lambda) \right).$$
APPENDIX B
PROOF OF THEOREM 1
Proof: When θ_{1}^{HT} = θ_{1}^{BS} = θ_{1}^{CS}, then HT, BS, and CS have the same energy consumption and PAoI. We first show that E[Δ^{BS}] ≤ E[Δ^{HT}]. When θ^{BS} = θ^{HT}, then have the same E[I_{11}], E[H_{1}], H_{1}^{(2)}(0), and E[G] for BS and HT. Therefore

\[ E[Δ^{BS}] - E[Δ^{HT}] = \frac{2D_{1}^{(1)}(λ_{1}) - 2D_{1}^{(1)}(λ_{1})E[U]}{2E[I_{11}]} - \frac{D_{1}^{(1)}(λ_{1})}{E[I_{11}]}(1 - N) - E[U] \leq 0. \]

We now show E[Δ^{CS}] ≤ E[Δ^{BS}], Since both BS and CS have the same E[I_{11}], E[H_{1}], H_{1}^{(2)}(0), and E[G_{1}], we then have

\[ E[Δ^{CS}] - E[Δ^{BS}] \]

with the last inequality following from the fact that E[H_{1}]E[e^{-b_{1}H_{1}}] - E[H_{1}e^{-b_{1}H_{1}}] ≥ 0 (see [28], [30]).

APPENDIX C
PROOF OF COROLLARY 3
Proof: Since both the denominator and numerator of E[P^{CS}] are linear functions of N, E[P^{CS}] is either increasing or decreasing. We only need to compare the E[P^{CS}] for N = ∞ and N = 1. When N = ∞ and min_{i}θ_{i} > 0, we have E[P] = P_{SL}. When N = 1,

\[ E[P^{CS}] \geq \sum_{i=1}^{k} \frac{1}{λ} P_{SL} \left[ E[H_{i}] + \frac{1}{λ} + \theta_{i}(\frac{1}{λ} + E[U]) \right] \]

Similarly, the minimal E[P] is achieved at either \( \theta_{i} = 1 \) or \( \theta_{i} = 0 \) for all i because the denominator and numerator of E[P^{CS}] are linear on \( \sum_{i=1}^{k} λ_{i}θ_{i} \). So the minimal E[P^{CS}] is achieved at either \( \theta_{i} = 0 \) or \( \theta_{i} = 1 \) for all i.

We now show that if \( P_{ST} \leq P_{TD} \), then the minimal E[P^{CS}] is achieved at \( \theta_{i} = 1 \) for all i = 1, .., k. When \( P_{ST} \leq P_{TD} \), we have

\[ E[P^{CS}] \geq \sum_{i=1}^{k} \frac{1}{λ} \left[ P_{B}E[H_{i}] + \frac{N}{λ} P_{SL} + E[U]P_{ST} \right] \]

\[ \geq \sum_{i=1}^{k} \frac{1}{λ} \left[ P_{B}E[H_{i}] + \frac{N}{λ} P_{SL} + E[U]P_{ST} \right] \]

The equality holds only when \( \theta_{i} = 1 \) for all i. Hence proved.

APPENDIX D
PROOF OF THEOREM 2
Proof: With the Lagrangian multiplier \( η = (η_{1}, η_{2}, η_{3}, η_{4}) \), we write the Lagrangian function of Problem P3 as

\[ L(η, N, θ) = E[P^{CS}(N, θ)] + η_{1}(E[A^{CS}(N, θ)] - τ) + η_{2}(θ - 1) - η_{3}θ - η_{4}(N - 1). \]

By the KKT condition for Problem P3, the optimal solution must satisfy

\[ \partial E[P^{CS}(N, θ)] + η_{1} \partial E[A^{CS}(N, θ)] - η_{4} = 0 \]

\[ η_{1}(E[A^{CS}(N, θ)] - τ) = 0 \]

\[ η_{i} ≥ 0 \text{ for } i ∈ \{1, 2, 3, 4\}. \]

From Corollaries 2 and 3, we have that for fixed θ > 0, then \( \partial E[P^{CS}(N, θ)] \) and \( \partial E[A^{CS}(N, θ)] \) are strictly increasing or decreasing. Therefore, we must have \( η_{1} > 0 \), which means that \( E[A^{CS}(N, θ)] - τ = 0 \) always holds for the optimal solution. So that in the optimization problem, the constraint \( E[A^{CS}] ≤ τ \) is tight for the optimal solution when θ > 0. Therefore, we can have \( \theta = \frac{τ - \frac{1}{2} 2E[H]}{N + E[U] - \frac{1}{λ}} \), and the optimization problem P3 can be rewritten as

\[ \min_{N} \left\{ \left[ \frac{N}{λ} E[H] - \frac{U^{*}(λ)}{λ} \right] [P_{B}E[H] + \frac{1}{λ} P_{TD}] + τ - \frac{1}{λ} - 2E[H] \right\} \left\{ \frac{N}{λ} P_{SL} + E[U]P_{ST} - P_{TD} \right\} \]

s.t. \( ∞ > N ≥ \max \left\{ 1, λ(τ - \frac{1}{λ} - 2E[H]) - E[U] + \frac{U^{*}(λ)}{λ} \right\} \).

We can then find that E[P^{CS}] has both denominator and numerator as linear functions of N. Therefore E[P^{CS}] is either increasing or decreasing function of N. If it is decreasing, then it is optimal to let N become very large while keep \( λ = \frac{τ - \frac{1}{2} 2E[H]}{N + E[U] - \frac{1}{λ}} \) > 0 and we have Type 1 solution. If E[P^{CS}] is an increasing function of N, then the minimum E[P^{CS}] is achieved when N reaches its lower bound, where we have either Type 2 or Type 3 solution.

APPENDIX E
PROOF OF LEMMA 5
Proof: From Equation (24), we find that to show E[A_{i}(λ, N, θ)] is increasing on λ_{i}, one only needs to show that \( f(λ, θ) = \sum_{i=1}^{k} \lambda_{i} \frac{U^{*}(λ)}{λ} + \frac{1}{λ} \) is a decreasing function of λ_{i}. We now take the derivative of f(λ, θ) and have
\[ \frac{\partial f(\lambda, \theta)}{\partial \lambda_i} = -\frac{1}{\lambda^2} + \frac{1}{\lambda^3} \left\{ \left( \theta_i \lambda - \sum_{j=1}^{k} \theta_j \lambda_j \right) (1 - U^*(\lambda)) \right. \\
+ \left. \sum_{j=1}^{k} \theta_j \lambda_j \left[ -U^*(1)(\lambda) \lambda - 1 + U^*(\lambda) \right] \right\}, \]

We further take the derivatives of \( \frac{\partial f(\lambda, \theta)}{\partial \lambda_i} \) regarding each \( \theta_j \) with \( j \neq i \), we have

\[ \frac{\partial^2 f(\lambda, \theta)}{\partial \lambda_i \partial \theta_j} = \frac{1}{\lambda^3} \left\{ -\lambda_j (1 - U^*(\lambda)) \right. \\
+ \left. \lambda_j \left[ -U^*(1)(\lambda) \lambda - 1 + U^*(\lambda) \right] \right\} \] for \( j \neq i \).

From the fact that

\[ -U^*(1)(\lambda) \lambda - 1 + U^*(\lambda) = \lambda E[U e^{-\lambda U}] - 1 + E[e^{-\lambda U}] \leq 0, \]

we have \( \frac{\partial^2 f(\lambda, \theta)}{\partial \lambda_i \partial \theta_j} \leq 0 \). So that

\[ \frac{\partial f(\lambda, \theta)}{\partial \lambda_i} = -\frac{1}{\lambda^2} + \frac{1}{\lambda^3} \left\{ \left( \theta_i \lambda - \sum_{j=1}^{k} \theta_j \lambda_j \right) (1 - U^*(\lambda)) \\
+ \sum_{j=1}^{k} \theta_j \lambda_j \left[ -U^*(1)(\lambda) \lambda - 1 + U^*(\lambda) \right] \right\} \]

\[ \leq \frac{1}{\lambda^3} \left\{ \lambda \left( \theta_i - \sum_{j=1}^{k} \theta_j \lambda_j \right) (1 - U^*(\lambda)) \\
+ \lambda \lambda_j \left[ -U^*(1)(\lambda) \lambda - 1 + U^*(\lambda) \right] \right\} - \frac{1}{\lambda^2}. \]

If \( \lambda - \lambda_i \left( 1 - U^*(\lambda) + \lambda \right) \lambda - 1 - U^*(\lambda) \lambda - 1 + U^*(\lambda) \right\} \leq 0, \]

then \( \frac{\partial f(\lambda, \theta)}{\partial \lambda_i} \leq -\frac{1}{\lambda^2} < 0 \). Otherwise,

\[ \frac{\partial f(\lambda, \theta)}{\partial \lambda_i} \leq \frac{1}{\lambda^3} \left\{ -\lambda U^*(\lambda) - \lambda_i (1 - U^*(\lambda)) \\
+ \lambda \lambda_i \left[ -U^*(1)(\lambda) \lambda - 1 + U^*(\lambda) \right] \right\} \leq 0. \]

Hence proved.

\[ \frac{\partial E[AiCS(\lambda, N, \theta)]}{\partial \lambda_i} = \frac{E[AU e^{-\lambda U}] - 1 + U^*(\lambda)}{\lambda^2} + \frac{E[H_i]}{\lambda_i} + \frac{E[U]}{\lambda_i}. \]

Let \( y(u) = \lambda u e^{-\lambda u} - 1 + e^{-\lambda u} + \lambda u \), we have \( \frac{\partial y(u)}{\partial \lambda u} = -\lambda u e^{-\lambda u} - \lambda + \frac{\partial y(u)}{\partial \lambda u} = \lambda^2 u e^{-\lambda u} - \lambda^2 e^{-\lambda u} \). So that \( \frac{\partial E[AiCS(\lambda, N, \theta)]}{\partial \lambda_i} \geq -\lambda e^{-\lambda} + \lambda \geq 0 \), and we have \( y(u) \geq y(0) = 0 \). Hence \( \frac{\partial E[AiCS(\lambda, N, \theta)]}{\partial \lambda_i} \geq 0 \).

\[ \text{APPENDIX G} \]

\text{PROOF OF Lemma [7]}

\[ L(\lambda, \beta, \gamma) = \sum_{i=1}^{k} c_i(\lambda_i) + \sum_{i=1}^{k} \varphi_i(\lambda_i - \lambda_{\text{max}}) \]

\[ + \sum_{i=1}^{k} \phi_i(E[\lambda_i^{CS}(\lambda, N, \theta)] - \tau_i) \]

be the Lagrange function of Problem P4. Then by KKT conditions, the optimal solution to P4 must satisfy

\[ \frac{\partial C_i(\lambda_i)}{\partial \lambda_i} + \sum_{i=1}^{k} \varphi_i(\lambda_i - \lambda_{\text{max}}) + \phi_i(E[\lambda_i^{CS}(\lambda, N, \theta)] - \tau_i) = 0 \]

\[ \phi_i(E[\lambda_i^{CS}(\lambda, N, \theta)] - \tau_i) = 0 \]

\[ \varphi_i(\lambda_i - \lambda_{\text{max}}) = 0 \]

\[ \phi_i \geq 0 \]

\[ \varphi_i \geq 0 \]

for \( i \in \{1, \ldots, k\} \). Since \( \frac{\partial C_i(\lambda_i)}{\partial \lambda_i} \geq 0 \) and \( E[\lambda_i^{CS}(\lambda, N, \theta)] \geq 0 \), we must have \( \phi_i \geq 0 \) to make Equation (60) hold. So by Equation (61), we must have \( E[\lambda_i^{CS}(\lambda, N, \theta)] = \tau_i \) for each \( i \).

\[ \text{APPENDIX H} \]

\text{PROOF OF Proposition [8]}

\[ g_i(\lambda) = \frac{1 - U^*(\|\lambda\|_1)}{\|\lambda\|_1} \lambda_i + \lambda_i (E[H_i] - \tau_i) \]

\[ + \sum_{i=1}^{k} \lambda_i (E[H_i] + E[U]) + N. \]

From Lemma [7] we have that the equilibrium strategy \( \lambda^e = (\lambda_1^e, \ldots, \lambda_k^e)^T \) must satisfy \( g(\lambda^e) = \left( \begin{array}{c} g_1(\lambda^e) \\ \vdots \\ g_k(\lambda^e) \end{array} \right) = \left( \begin{array}{c} 0 \\ \vdots \\ 0 \end{array} \right) \). One can rewrite the system \( g(\lambda^e) = 0 \) as \( (Q(\lambda^e) + 1) \lambda^e = -N1, \) where \( Q \) is a diagonal matrix with \( Q_i = E[H_i] - \tau_i + \frac{1}{1 - U^*(\|\lambda^e\|_1)} \) and \( \kappa \) is a \( k \times 1 \) vector with the \( i \)th element being \( E[H_i] + E[U] \). Since

\[ \sum_{i=1}^{k} \frac{E[H_i] + E[U]}{\lambda_i} - \tau_i + \frac{1 - U^*(\|\lambda^e\|_1)}{\|\lambda\|_1} \]

\[ \sum_{i=1}^{k} \frac{E[H_i] + E[U]}{\lambda_i} + \frac{N}{\lambda_i} + \frac{1}{\lambda_i} E[U] \]
\[
\sum_{i=1}^{k} \lambda_i E[H_i] + \lambda E[U] > -1,
\]

we have \(1 + \kappa^T \hat{Q}^{-1} 1 \neq 0\). By the Sherman–Morrison formula [31] we have \(\hat{Q}(\lambda^r) + 1 \kappa^T\) to be invertible. Suppose \(\lambda_1 = -(\hat{Q}(\lambda_1) + 1 \kappa^T)^{-1} N 1\) and \(\lambda_2 = -(\hat{Q}(\lambda_2) + 1 \kappa^T)^{-1} N 1\) are two distinct solutions to \(g(\lambda) = 0\), then we must have either \(\lambda_1 \geq \lambda_2\) or \(\lambda_1 \leq \lambda_2\) element-wise. This is because \(\hat{Q}(\lambda)\) is a matrix whose elements are monotone on \(\|\lambda\|_1\). If \(\|\lambda_1\|_1 \geq \|\lambda_2\|_1\), then \(\hat{Q}(\lambda_1) \leq \hat{Q}(\lambda_2)\) element-wise, which means \(\lambda_1 \leq \lambda_2\). We thus must have \(\lambda_1 = \lambda_2\), which means \(g(\lambda) = 0\) has a unique solution. Hence proved.