A General Model for Minimizing Age of Information at Network Edge

Chengzhang Li, Shaoran Li, and Y. Thomas Hou
Virginia Polytechnic Institute and State University, Blacksburg, VA, USA

***Abstract*—Recently, a new metric, called Age of Information (AoI), has become popular to quantify the freshness of information collected at network edge. AoI research is still in its infancy and most prior efforts assume overly simplified models in their investigation. In this paper, we consider a more general model for AoI research that is closer to what happens in the real world. Specifically, we consider general and heterogeneous sampling behaviors among source nodes, varying sample size, and multiple data transmission units in each time slot. Under this much general setting, we develop new theoretical results (in terms of properties and performance bounds) and a new near-optimal low-complexity scheduling algorithm. Our results make a major advance of AoI research in terms of more realistic models.**

I. Introduction

Data collection is a critical component in modern network systems. In a typical scenario, at the network edge, multiple source nodes for different applications collect samples of information from the physical environment and forward the sampled information to the edge server. The collected information can then be either processed and stored locally (edge computing) and/or forwarded to the cloud. Since many applications on the upper layer depend on the timeliness of the sampled information, it is critical to forward the freshest sample to the edge as soon as possible.

Recently, the so-called Age of Information (AoI) has been introduced as a new metric to quantify the freshness of information [1, 2]. AoI is defined as the elapsed time for a sample (stored at a particular location, e.g., edge or cloud) between current time (now) and the time when the sample was first generated at its source. Due to potentially large number of source nodes and limited transmission capacity between source nodes and the base station (BS), not all newly generated samples can be forwarded to the BS at the same time. As a result, a scheduling algorithm is needed to allocate the limited channel resource to the source nodes to minimize AoI for all source nodes.

There has been active research on minimizing AoI (see, e.g., [3–17] and a bibliography on this research in [18]). Most of the efforts consider very simple models where one sample is taken at each source node in each time slot (i.e., per time slot sampling), with one unit of data in each sample, and channel transmission capacity is one unit of data in each time slot. Specifically, papers most relevant to this paper include [3–7]. In [3] the authors considered cache updating under the above simple model. In [4–7] the authors considered the simple model with unreliable channel (i.e. each transmission has a probability to fail). In addition, in [5], the authors considered a throughput constraint. In [6], the authors considered the influence of CSI knowledge. In [7], the authors considered a general channel constraint and some heterogeneous sampling behaviors. Although results from such simple models (in terms of sampling behavior, sample size, transmission rate) offer some initial understanding on AoI, the practical applications from these results are limited.

In this paper, we study AoI in a general setting that is more relevant to applications in the real world. Specifically, we consider the following general models. (i) We consider various sampling behaviors at each source, such as arbitrary sampling, periodic sampling, and per time slot sampling. Note that per time slot sampling, the simplest sampling behavior, is what has been mostly studied in the literature. However, sampling behaviors such as arbitrary sampling and periodic sampling (with each source having different sampling period) are most common in real-world applications and deserve further investigation. (ii) We allow sample size collected at each source node to vary, depending on the underlying application. This is a major generalization of state-of-the-art where sample sizes from all sources are identical (one unit). Again, this generalization is absolutely necessary if we want to have our AoI research to be relevant to applications in the real world. (iii) We generalize the transmission capacity with multiple transmission units in each time slot. This is also a generalization of state-of-the-art where there is only one transmission unit.

The main contributions of this paper are the following:

- We study AoI with a much more general model than those used in the state-of-the-art in terms of sampling behavior at the source nodes, sample size collected from each source, and transmission capacity. Such generalizations offer much better characterization of source heterogeneity and transmission behavior in the real world. As a result, findings based on this general model not only have more significance from theoretical perspective, but also have greater impacts on applications in the real world.
- Under this general model, it is much more challenging to design an AoI minimization scheduler, due to a much larger search space than those considered in the literature. As a first step, we develop two properties for an optimal scheduling solution. We show these properties help reduce the search space for the optimal and near-optimal solution. They also serve as a guideline for developing optimal AoI scheduling algorithms for other AoI problems in the general setting.

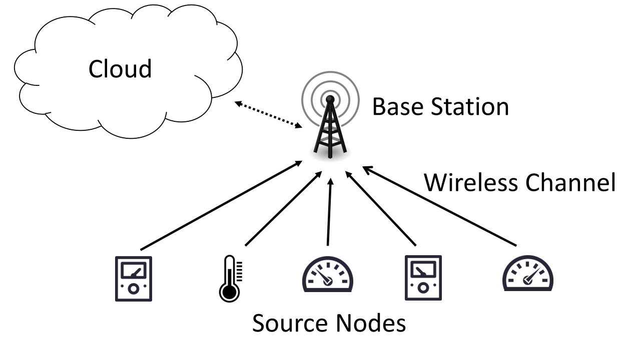


Fig. 1. System Model: N nodes collect information and forward it to a BS.

- In the reduced search space, we further develop theoretical lower bounds for AoI under different sampling behaviors (arbitrary, periodic and per time slot). These lower bounds serve as performance benchmarks to assess the quality of a scheduling algorithm under different sampling behaviors.
- We design a low-complexity scheduling algorithm that is applicable to all three sampling behaviors. Through theoretical analysis, we find that our algorithm can guarantee a factor of 3 from the objective value. Through simulation study, we find that our scheduling algorithm is near-optimal when there is no synchronization in sampling among the sources.

II. System Model

Consider a network consisting of N source nodes and one BS as shown in Fig. 1. Each source node samples information from its environment and attempts to transmit it to the BS. For uplink data transmission, time is equally divided into time slots and each time slot can accommodate a number of data transmission units. Denote M as the number of data transmission units per time slot over the uplink bandwidth. Then, in each time slot, the BS is capable of allocating the M transmission units to a single or multiple source nodes for uplink data transmission. For simplicity, with respect to each source node, we assume each transmission unit carries the same amount of information over all time slots.[1]

At each source node, information is collected (or generated) at a specific sampling rate at this source. Denote L_i (in number of transmission units) as the amount of information in each sample for source node i. A wide range of sampling behavior are possible among the N sources, such as:

- **Arbitrary Sampling.** Each node perform its own sampling (either following a random or deterministic pattern) and is independent of other sources. This sampling behavior is the most general among all sampling behaviors.
- **Periodic Sampling.** Source node i performs sampling at every T_i time slots. The sampling intervals (T_i's) are generally different among different source nodes. This sampling behavior is likely the most prevailing sampling behavior among real-world applications.
- **Per Time Slot Sampling.** Each source node samples information in every time slot. This is the special case for periodic sampling with $T_i = 1$ for every node i. It is a model used by most works in AoI research (see, e.g., [3, 5, 6]). Although simple, this model may not be an accurate characterization of real world sampling behavior as each source node usually samples at different rate, due to difference in applications.

[1]The general case considering channel diversity in time and frequency domains will be explored in our future research.

When the BS allocates transmission units to a source node, the source node will always transmit its freshest sample (the most recently generated sample). Recall each sample from each node i consists of L_i units of information. Due to the size of L_i, it may take multiple time slots to complete the transmission of this sample. Only after all L_i units of the sample from source i are transmitted to the BS, we say the BS has received this sample. Once the transmission of a sample begins, the remaining unfinished units from this sample must be transmitted (over multiple time slots if needed) before any new sample is considered (even if the new sample is fresher than the one currently under transmission).

The BS maintains the sample that it has most recently received from each source node and considers it the freshest information that it possesses from that source. Again, a sample from a source is not considered received until the sample (consisting of multiple units) is received in its entirety (possibly requiring multiple time slots). Upon receiving a sample from source i completely, the BS replaces the previous sample from source i with this newly received sample.

III. AoI Modeling and Problem Statement

At each source node i, denote $U_i^{\mathrm{s}}(t)$ as the generation time of the most recent sample at time slot t. Denote $A_i^{\mathrm{s}}(t)$ as the AoI at source node i at time slot t. We have

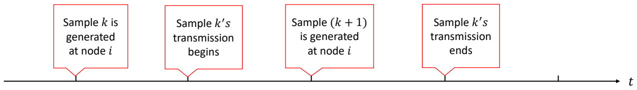
$$A_i^{\mathrm{s}}(t) = t - U_i^{\mathrm{s}}(t). \tag{1}$$

Note that $A_i^{\mathrm{s}}(t)$ is a zigzag-like function with a slope of 1 between sampling intervals and is reset to 0 in each time slot when a new sample is generated. Clearly, $0 \leq A_i^{\mathrm{s}}(t) < T_i$ in the periodic sampling case.

At the BS, it maintains the most recent (complete) sample that it has received from each of the N source nodes. Note that this sample maintained at the BS from source node i may be different from (older than) the freshest sample currently at source node i. Denote $U_i^{\mathrm{B}}(t)$ as the generation time of the sample from source node i that is currently maintained by the BS at time slot t. Denote $A_i^{\mathrm{B}}(t)$ as the AoI for this sample at the BS at time slot t. Then we have

$$A_i^{\mathrm{B}}(t) = t - U_i^{\mathrm{B}}(t). \tag{2}$$

Since $U_i^{\mathrm{B}}(t) \leq U_i^{\mathrm{s}}(t)$, we have $A_i^{\mathrm{B}}(t) \geq A_i^{\mathrm{s}}(t)$, i.e., the AoI for source node i as perceived (maintained) by the BS is older (larger) than or equal to that at the source node, which is intuitive. Note that $A_i^{\mathrm{B}}(t)$ is also a zigzag-like function with a



t	$t_i(k)$	$b_i(k)$	$t_i(k+1)$	$e_i(k)$	$e_i(k)+1$
$U_i^S(t)$	$t_i(k)$	$t_i(k)$	$t_i(k+1)$	$t_i(k+1)$	$t_i(k+1)$
$A_i^S(t)$	0	$b_i(k)-t(k)$	0	$e_i(k)-t_i(k+1)$	$e_i(k)-t_i(k+1)+1$
$U_i^B(t)$	$t_i(k-1)$	$t_i(k-1)$	$t_i(k-1)$	$t_i(k-1)$	$t_i(k)$
$A_i^B(t)$	$t_i(k)-t_i(k-1)$	$b_i(k)-t_i(k-1)$	$t_i(k+1)-t_i(k-1)$	$e_i(k)-t_i(k-1)$	$e_i(k)-t_i(k)+1$

Fig. 2. An example showing the evolution of $U_i^{\text{s}}(t)$ and $A_i^{\text{s}}(t)$ at source node i versus $U_i^{\text{B}}(t)$ and $A_i^{\text{B}}(t)$ at the BS during different time instances.

slope of 1 between time instances when a sample is received and is reset at the end of each time slot when a new sample is completely received at the BS.

We now make a connection between $A_i^{\text{B}}(t)$ and $A_i^{\text{s}}(t)$. From source node i, for the k-th sample that is actually selected for transmission,[2] denote its beginning (starting) transmission time slot as $b_i(k)$, and ending (finishing) transmission time slot as $e_i(k)$, where $e_i(k) \geq b_i(k)$. Since this k-th sample is selected for transmission at time $b_i(k)$, it must be the freshest sample at source node i at that time, with a generation time of $U_i^{\text{s}}\big(b_i(k)\big)$. After this k-th sample is completely sent to the BS at the end of time slot $e_i(k)$, in the beginning of the next time slot $(e_i(k)+1)$, we have

$$U_i^{\text{B}}(e_i(k)+1) = U_i^{\text{s}}\big(b_i(k)\big).$$

From (2) and (1), we have

$$\begin{aligned} A_i^{\text{B}}(e_i(k)+1) &= e_i(k)+1-U_i^{\text{B}}(e_i(k)+1) \\ &= e_i(k)+1-U_i^{\text{s}}(b_i(k)) \\ &= e_i(k)+1-\Big(b_i(k)-A_i^{\text{s}}\big(b_i(k)\big)\Big) \\ &= A_i^{\text{s}}\big(b_i(k)\big)+e_i(k)-b_i(k)+1. \end{aligned}$$

Therefore, over all t, we have

$$A_i^{\text{B}}(t+1) = \begin{cases} A_i^{\text{s}}\big(b_i(k)\big)+e_i(k)-b_i(k)+1, & \text{if } t=e_i(k), \\ A_i^{\text{B}}(t)+1, & \text{otherwise.} \end{cases} \tag{3}$$

An example of AoI evolution is given in Fig. 2.

Based on (3), the long-term average of source node i's AoI at the BS can be written as:

$$\bar{A}_i^{\text{B}} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} A_i^{\text{B}}(t). \tag{4}$$

Denote w_i as the weight of source node i's information, which can be used to reflect the priority of node i. Then the AoI over all source nodes at the BS can be written as

$$\bar{A}^{\text{B}} = \sum_{i=1}^{N} w_i \bar{A}_i^{\text{B}}. \tag{5}$$

Since there are only M data units available for transmission in each time slot, a scheduling algorithm is needed to decide how to allocate the M data units to a subset of source nodes in each time slot. Denote $\boldsymbol{X}(t)$ as the scheduling decision for time slot t, where $\boldsymbol{X}(t)$ is an $N\times 1$ vector with its i-th element $x_i(t)$ being the number of data units that is allocated to user i at time slot t. Since each transmission data unit can be allocated to at most one source node, we have

$$\sum_{i=1}^{N} x_i(t) \leq M. \tag{6}$$

Clearly, each different scheduling algorithm will yield a very different performance of $\bar{A}^{\text{B}}$ in (5). Our goal is to find an optimal scheduling algorithm so that $\bar{A}^{\text{B}}$ is minimized.

Based on (4), minimizing $\bar{A}_i^{\text{B}}$ requires the design of a scheduling algorithm over an infinite number of time slots, which makes the search space for $\boldsymbol{X}(t)$ infinite. To address this problem, we show, in the next section, how to reduce the search space for an optimal scheduling solution.

IV. Properties for An Optimal Scheduling Algorithm

Given that an optimal solution to our scheduling problem may not be unique, an efficient approach to reduce the search space is to find some properties associated with a particular optimal scheduling solution. Based on these properties, it becomes more tractable to find an optimal solution or design a near optimal solution.

A. An Order-based Scheduling

At each time slot t, it's intuitive to perform an order-based scheduling, that is, to assign an order for all source nodes and allocate data transmission units to the source node currently with the highest order before allocating to the source node with the second highest order and so forth. The order can be designed based on $A_i^{\text{s}}(t)$, $A_i^{\text{B}}(t)$, w_i and L_i for each source node.

The following lemma states that for each time slot t, there exists an optimal order-based scheduling algorithm that minimizes $\bar{A}^{\text{B}}$.

Lemma 1 *Under arbitrary sampling, there exists an order-based scheduling algorithm that achieves the optimal objective.*

A Proof Sketch Suppose $\boldsymbol{X}^*(t)$ is an optimal scheduling algorithm that minimizes $\bar{A}^{\text{B}}$. For any time slot t, suppose the scheduled transmission samples are from source nodes $i_1, i_2, \cdots, i_P$ with ending time slots $e_1, e_2, \cdots, e_P$ such that $t \leq e_1 \leq e_2 \leq \cdots \leq e_P$. Denote $\mathcal{S}$ as the set of transmission units that are allocated to source nodes $i_1, i_2, \cdots, i_P$ to complete their current samples from time t (inclusive) under $\boldsymbol{X}^*(t)$. We define the order of those source nodes as $i_1 > i_2 > \cdots > i_P$. Based on this order, we can re-allocate the transmission units in $\mathcal{S}$ as follows. We first allocate transmission units to finish i_1's sample in its entirety and then move to i_2, and so on. By doing this, the ending time slot of each node will not increase and thus the new $\bar{A}^{\text{B}}$ is either equal or smaller than the previous objective. Since the previous

[2]Recall that not every sample generated at source node i will be transmitted to the BS.

objective is optimal, then only equality is possible and such order-based scheduling is optimal. ∎

Using this property, we only need to find or design an order-based scheduling algorithm. This property allows us to work in a much smaller search space. Also note that since this property is for arbitrary sampling, it applies to periodic and per time slot sampling policies as well.

B. Cyclic Transmission

Another property that we want to explore is whether an optimal scheduling algorithm exhibits a cyclic (periodic) transmission pattern, i.e., with the same scheduling decision for every, say T_c, time slots. It appears that such property is hard to establish under arbitrary sampling policy. So we will focus on periodic sampling policy, which also includes per-time slot sampling policy.

More formally, we say a scheduling algorithm is *cyclic* if it repeats its scheduling decision for a fixed number of time slots. Denote $\boldsymbol{X}_c(t)$ as a cyclic scheduling algorithm and T_c as its cycle (in number of time slots). Then there exists a t_0 such that for any $t > t_0$, we have

$$\boldsymbol{X}_c(t) = \boldsymbol{X}_c(t + T_c).$$

The following lemma states the existence of such an optimal cyclic scheduler under periodic sampling policy.

Lemma 2 *When each source is sampled periodically (even with different periods), there exists a cyclic scheduling algorithm that achieves optimal objective.*

A Proof Sketch We first define the *state* of the network in a time slot as the complete information of current AoI at the source nodes, current AoI at the BS, and number of remaining transmission units that are still needed for each source node. Each state has a corresponding weighted-sum AoI at the BS. We can prove that there exists an optimal scheduling algorithm $X^*(t)$ (with objective $\bar{A}^*$) that only visits a subset of states, and the corresponding weighted-sum AoI at the BS for each state in the subset is smaller than a finite upper bound. Therefore, there are only a finite number of possible AoI values at the BS for states in the subset. Since the number of possible combinations of other components in a state is also finite, the number of states in the subset is finite. As time goes to infinity, there must be a state that appears infinite times. We then divide the time domain into infinite number of segments based on the appearance of this state, with this state appearing in the first time slot of each segment. Obviously, there must be a segment with average AoI at the BS smaller than or equal to $\bar{A}^*$. Since $X^*(t)$ is optimal, only equality is possible. Then we can construct an optimal cyclic scheduling algorithm by repeating the states within this segment. ∎

Since Lemma 2 applies to periodic sampling policy, it also applies to per time slot sampling policy.

C. Complexity Analysis

Under arbitrary sampling, Lemma 1 helps reduce the search space. If sampling is periodic, the search space can be further reduced by Lemma 2. However, even under periodic sampling, the reduced search space for an optimal scheduling algorithm is still infinite since T_c can be any number. Therefore, we have to pursue an efficient heuristic algorithm to achieve near-optimal performance.

V. Performance Bounds

Before we design a scheduling algorithm, we first develop some lower bounds for our objective function under different cases. These results are not only important to serve as a performance benchmark to assess the scheduling algorithm that we will develop (in Section VI), they are also of significant theoretical value on their own as they generalize a number of results (developed for special or simple cases) in the literature.

We will develop lower bounds of our objective function, denoted as $\alpha_{(*,*)}$ for the following four cases: (i) per time slot sampling under finite link capacity: $\alpha_{(\text{PTS},M)}$, (ii) arbitrary sampling under infinite link capacity: $\alpha_{(\text{ARB},\infty)}$, (iii) arbitrary sampling under finite link capacity: $\alpha_{(\text{ARB},M)}$, and (iv) periodic sampling under finite link capacity: $\alpha_{(\text{PRD},M)}$.

A. The Case of Per Time Slot Sampling Under Finite Link Capacity

In this case, each source node takes a sample at every time slot, i.e., $T_i = 1$ and $A_i^{\text{s}}(t) = 0$ for all i. Here the AoI at the BS is purely limited by the link capacity, M. In the literature (see, e.g., [3, 5]), lower bounds for the same objective function have been developed for the simple case where $M = 1$ and $L_i = 1$ for each source node i. Our development here is for a general value of $M \geq 1$ and different values of L_i for each different user, which is what happens in practice.

Since per time slot sampling is a special case of periodic sampling, by Lemma 2, there exists an optimal cyclic algorithm $\boldsymbol{X}_c^*(t)$ with a cycle T_c. Denote N_i as the number of fully transmitted samples from node i over a cycle of T_c time slots. In a cycle, the number of transmission units allocated among the source nodes cannot be more than the total number of available transmission units. We have

$$\sum_{i=1}^{N} N_i L_i \leq M \cdot T_c. \tag{7}$$

Define r_i as the transmission rate for source node i, i.e.,

$$r_i = \frac{N_i}{T_c}. \tag{8}$$

Under per time slot sampling, since at most one sample from each source node can be sent to the BS, we have

$$0 < r_i \leq 1. \tag{9}$$

Intuitively, r_i shows the percentage of a sample from source node i that can be transmitted in a time slot.

Dividing (7) by T_c and using (8), we have

$$\sum_{i=1}^{N} r_i L_i \leq M. \tag{10}$$

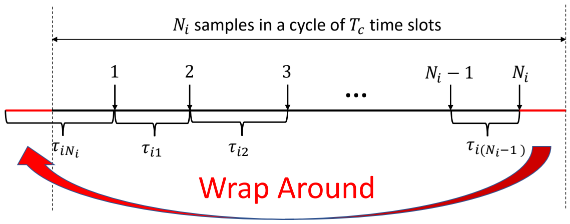


Fig. 3. A cycle with N_i samples. Each sample has its time interval since the last sample. The first interval is formed by connecting two partial intervals in the beginning and end of this cycle.

For source node i, it has N_i samples transmitted in a cycle. Consider the time interval between two successive transmitted samples. Then we have (N_i-1) intervals. By wrapping around a transmission cycle and viewing it cyclically (see Fig. 3), the time from the beginning of the cycle until the first transmitted sample and the time from the N_i-th transmitted sample to the end of the cycle together can be considered as the interval time for the first transmitted sample. We have, therefore, a total of N_i intervals for source node i in T_c. Denote these N_i intervals as $\tau_{i1}, \tau_{i2}, ...\tau_{iN_i}$ (see Fig. 3), we have

$$\sum_{j=1}^{N_i} \tau_{ij} = T_c.$$

Since it takes at least one time slot to transmit a sample from source node i to the BS, $A_i^{\text{B}}(t) \geq 1$. Then, during time interval τ_{ij}, the sum of AoI at the BS (for source node i) is at least

$$1 + 2 + ... + \tau_{ij} = \frac{\tau_{ij}^2 + \tau_{ij}}{2}.$$

So in a cycle with T_c time slots, a lower bound for $\bar{A}_i^{\text{B}}$ can be found by taking time average (over T_c time slots) of $\bar{A}_i^{\text{B}}(t)$ for N_i time intervals. We have

$$\bar{A}_i^{\text{B}} \geq \frac{1}{T_c} \sum_{j=1}^{N_i} \frac{\tau_{ij}^2 + \tau_{ij}}{2} = \frac{1}{T_c} \sum_{j=1}^{N_i} \frac{\tau_{ij}^2}{2} + \frac{1}{2}. \tag{11}$$

Applying the Cauchy-Schwarz inequality, we have

$$N_i \sum_{j=1}^{N_i} \tau_{ij}^2 \geq (\sum_{j=1}^{N_i} \tau_{ij})^2 = T_c^2. \tag{12}$$

Applying (12) to (11) we have

$$\bar{A}_i^{\text{B}} \geq \frac{T_c}{2N_i} + \frac{1}{2} \quad \text{or} \quad \bar{A}_i^{\text{B}} \geq \frac{1}{2r_i} + \frac{1}{2}.$$

Based on (5), we have

$$\bar{A}^{\text{B}} \geq \sum_{i=1}^{N} w_i (\frac{1}{r_i} + \frac{1}{2}). \tag{13}$$

To find a lower bound for $\bar{A}^{\text{B}}$, we can use a lower bound for $\sum_{i=1}^{N} w_i(\frac{1}{2r_i} + \frac{1}{2})$, which means we need to find the minimum of $\sum_{i=1}^{N} \frac{w_i}{r_i}$. We have the following optimization problem:

$$\begin{aligned} \min_{r_i} \quad & \sum_{i=1}^{N} \frac{w_i}{r_i} \\ \text{s.t.} \quad & \text{Constraints (9) and (10)} \end{aligned} \tag{14}$$

The above optimization problem is convex and can be easily solved. In the optimal solution, there are K nodes ($0 \leq K \leq N$) with their $r_i^* = 1$ and the remaining $N - K$ nodes with their $r_i^* < 1$. Without loss of generality, we assume $r_i^* = 1$ for $i \leq K$ and $r_i^* < 1$ for $i > K$. Define

$$M_K = M - \sum_{i=1}^{K} L_i. \tag{15}$$

In solving the convex optimization, the KKT conditions require, for $i \leq K$,

$$\sqrt{\frac{w_i}{L_i}} \geq \frac{\sum_{j=K+1}^{N} \sqrt{w_j L_j}}{M_K}, \tag{16}$$

and for $i > K$, r_i^* is given as:

$$r_i^* = \frac{M_K \sqrt{\frac{w_i}{L_i}}}{\sum_{j=K+1}^{N} \sqrt{w_j L_j}} < 1. \tag{17}$$

With the optimal solution to (14), a lower bound of $\bar{A}^{\text{B}}$, denoted by $\alpha_{(\text{PTS},M)}$, is given by

$$\begin{aligned} \alpha_{(\text{PTS},M)} &= \sum_{i=1}^{N} w_i (\frac{1}{2r_i^*} + \frac{1}{2}) \\ &= \frac{1}{2M_K} (\sum_{i=K+1}^{N} \sqrt{w_i L_i})^2 + \frac{1}{2} \sum_{i=K+1}^{N} w_i + \sum_{i=1}^{K} w_i. \end{aligned} \tag{18}$$

In the special case of $M = 1$ and $L_i = 1$, we have $K = 0$ and $M_K = 0$. Therefore,

$$r_i^* = \frac{\sqrt{w_i}}{\sum_{j=1}^{N} \sqrt{w_j}} \quad (1 \leq i \leq N), \tag{19}$$

and the lower bound

$$\alpha_{(\text{PTS},1)} = \frac{1}{2} (\sum_{i=1}^{N} \sqrt{w_i})^2 + \frac{1}{2} \sum_{i=1}^{N} w_i, \tag{20}$$

which are the main results reported in [3].

B. The Case of Arbitrary Sampling Under Infinite link Capacity

In this case the link capacity is infinite, i.e., $M \to \infty$. Here the BS can update information for all nodes in every time slot. $\bar{A}^{\text{B}}$ is purely limited by the source sampling, $A_i^{\text{s}}(t)$.

We define the average AoI at the source node i as

$$\bar{A}_i^{\text{s}} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} A_i^{\text{s}}(t). \tag{21}$$

From the evolution of AoI (3), under infinite link capacity, we have

$$A_i^{\mathrm{B}}(t) = A_i^{\mathrm{s}}(t-1) + 1, \quad \text{for } t > 0.$$

So in this case $\bar{A}^{\mathrm{B}}$ equals to

$$\alpha_{(\mathrm{ARB},\infty)} = \sum_{i=1}^{N} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} w_i(A_i^{\mathrm{s}}(t)+1) = \sum_{i=1}^{N} w_i(\bar{A}_i^{\mathrm{s}}+1). \tag{22}$$

This means the average AoI at the BS side is the average AoI at the source side plus 1. Here "1" is the transport delay of the network, meaning that the fresh information at the source needs one time slot to be sent to the BS. Note that under infinite link capacity, $\alpha_{(\mathrm{ARB},\infty)}$ is actually a constant rather than a theoretical bound.

It appears that none of the existing works considered the limitation of source sampling. $\alpha_{(\mathrm{ARB},\infty)}$ reveals an important fact that infinitely increasing link capacity cannot decrease $\bar{A}^{\mathrm{B}}$ to 0. Instead, $\bar{A}^{\mathrm{B}}$ will converge to $\alpha_{(\mathrm{ARB},\infty)}$.

C. The Case of Arbitrary Sampling Under Finite Link Capacity

In Section V-A and V-B, we have already derived two lower bounds, $\alpha_{(\mathrm{PTS},M)}$ and $\alpha_{(\mathrm{ARB},\infty)}$, respectively from the limitation of link capacity and source sampling. In the general case of arbitrary sampling and finite link capacity, both $\alpha_{(\mathrm{PTS},M)}$ and $\alpha_{(\mathrm{ARB},\infty)}$ will apply and we can choose the tighter of the two as the lower bound. That is,

$$\alpha_{(\mathrm{ARB},M)} = \max(\alpha_{(\mathrm{PTS},M)}, \alpha_{(\mathrm{ARB},\infty)}). \tag{23}$$

In the previous works (see, e.g., [3, 5]), the authors did not consider the impact of source sampling when developing a lower bound. Thus, under arbitrary sampling and finite link capacity where source sampling is the major limiting for $\bar{A}^{\mathrm{B}}$ (e.g., when M is relatively large), $\alpha_{(\mathrm{PTS},M)}$ (a generalization of the lower bounds in [3, 5]) can be much looser than $\alpha_{(\mathrm{ARB},M)}$ developed in this paper.

D. The Case of Periodic Sampling Under Finite Link Capacity

Under the periodic sampling case (under finite link capacity), we use a new relaxation technique to develop a tighter lower bound, $\alpha_{(\mathrm{PRD},M)}$.

By Lemma 2, there exists an optimal cyclic algorithm $\boldsymbol{X}_c^*(t)$ with a cycle T_c. It's easy to see T_c should be a multiple number of each node's sampling cycle, T_i. Just as in Section V-A, denote N_i as the number of fully transmitted samples from source node i over a cycle. Eq. (7) still holds. Other than the transmission rate r_i, for the periodic sampling case, we define p_i as the transmission percentage for source node i, i.e.,

$$p_i = \frac{N_i T_i}{T_c}. \tag{24}$$

Intuitively, p_i represents the percentage of fully transmitted samples over all generated samples in a cycle of T_c time slots. Clearly, we have

$$0 < p_i \le 1. \tag{25}$$

Dividing (7) by T_c and using (24), we have

$$\sum_{i=1}^{N} \frac{p_i L_i}{T_i} \le M. \tag{26}$$

Under periodic sampling we can also find N_i time intervals for each source node i in one cycle, $\tau_{i1}, \tau_{i2}, ...\tau_{iN_i}$, as we did in Section V-A. To obtain a lower bound of $\bar{A}^{\mathrm{B}}$, we assume that transmission of each sample can be finished in one time slot. Consider the following problem: If N_i is given, when should these N_i transmissions occur in order to minimize $\bar{A}^{\mathrm{B}}$ in a cycle? Under the optimal strategy (to achieve the smallest $\bar{A}^{\mathrm{B}}$), transmission of a sample should occur in the time slot immediately following the time instance when the sample is taken. Further, under the optimal transmission strategy, the lengths of transmission intervals should be similar. That is, the difference between any two transmission intervals is at most one T_i. Otherwise, we can use the average of their intervals for transmission and obtain a smaller $\bar{A}^{\mathrm{B}}$.

Therefore, if we define $H_i = \lfloor \frac{1}{p_i} \rfloor$ (where $\lfloor\cdot\rfloor$ is the floor function), to minimize $\bar{A}^{\mathrm{B}}$ in one cycle, each transmission interval τ_{ij} should be equal to either $H_i T_i$ or $(H_i+1)T_i$. Suppose in one cycle, there are n_1 intervals with length $H_i T_i$ and n_2 intervals with length $(H_i+1)T_i$. We have

$$\begin{cases} n_1 + n_2 = N_i \\ n_1 H_i T_i + n_2 (H_i+1) T_i = T_c. \end{cases} \tag{27}$$

Solving n_1 and n_2, we have

$$\begin{cases} n_1 = (H_i+1)N_i - \frac{T_c}{T_i} \\ n_2 = \frac{T_c}{T_i} - H_i N_i. \end{cases} \tag{28}$$

Since (11) still holds in this case, we can substitute (28) into (11) and we have

$$\begin{aligned} \bar{A}_i^B &\ge \Big(\big((H_i+1)N_i - \frac{T_c}{T_i}\big)(H_i T_i)^2 \\ &\quad + \big(\frac{T_c}{T_i} - u_i N_i\big)\big((H_i+1)T_i\big)^2 \Big) \frac{1}{2T_c} + \frac{1}{2} \\ &= \frac{T_i}{2}(2H_i + 1 - (H_i^2 + H_i)p_i) + \frac{1}{2} \\ &= \frac{T_i}{2} f(p_i) + \frac{1}{2}, \end{aligned}$$

where $f(p_i)$ is defined by

$$f(p_i) = 2[\frac{1}{p_i}] + 1 - \big([\frac{1}{p_i}]^2 + [\frac{1}{p_i}]\big)p_i. \tag{29}$$

To find a lower bound for $\bar{A}^{\mathrm{B}}$, we can use a lower bound for $\sum_{i=1}^{N} w_i(\frac{T_i}{2} f(p_i) + \frac{1}{2})$, which means we need to find the minimum of $\sum_{i=1}^{N} w_i T_i f(p_i)$. We have the following optimization problem:

$$\begin{aligned} \min_{p_i} \quad & \sum_{i=1}^{N} w_i T_i f(p_i) \\ \text{s.t.} \quad & \text{Constraints (25) and (26)} \end{aligned} \tag{30}$$

Note that $f(p_i)$ can be considered as a piece-wise linear function. So the optimization problem (30) can be reformulated to a linear programming problem, which can be easily solved. We omit the details for solving (30) due to paper length limitation.

With the optimal solution to (30) (denoted by p_i^* for all i), a lower bound of $\bar{A}^{\text{B}}$, denoted by $\alpha_{(\text{PRD},M)}$, is given by

$$\alpha_{(\text{PRD},M)} = \sum_{i=1}^{N} w_i(\frac{T_i}{2} f(p_i^*) + \frac{1}{2}). \quad (31)$$

In the above derivation for $\alpha_{(\text{PRD},M)}$, we consider the impacts of both link capacity and source sampling. With consideration of the fact that $f(p_i) \geq 1/p_i$ and $f(p_i) \geq 1$, we can find $\alpha_{(\text{PRD},M)}$ is always tighter than both $\alpha_{(\text{PTS},M)}$ and $\alpha_{(\text{ARB},\infty)}$. Therefore, $\alpha_{(\text{PRD},M)}$ is always tighter than the lower bound for arbitrary sampling, $\alpha_{(\text{ARB},M)}$.

VI. Juventas: A Near-Optimal Scheduling Algorithm

In this section, we propose a low-complexity scheduling algorithm, code named Juventas[3], in the reduced search space derived in Section IV.

For source node i, suppose transmission of a sample begins at t_1 and ends at t_2 ($t_2 \geq t_1$). Then, at time slot (t_2+1), based on (3), we have

$$A_i^{\text{B}}(t_2+1) = A_i^{\text{s}}(t_1) + t_2 - t_1 + 1. \quad (32)$$

On the other hand, if during the same time interval $[t_1, t_2]$, source node i is not scheduled for any transmission, then based on (3), we have

$$A_i^{\text{B}}(t_2+1) = A_i^{\text{B}}(t_1) + t_2 - t_1 + 1. \quad (33)$$

Note that $A_i^{\text{B}}(t_2+1)$ in (33) is greater than $A_i^{\text{B}}(t_2+1)$ in (32) if the sample does not complete its transmission by time t_2. So the benefit of completing transmission of this sample by t_2 (in terms of decrease of $A_i^{\text{B}}(t_2+1)$) is the difference on the RHS in (33) and (32), i.e.,

$$A_i^{\text{B}}(t_1) - A_i^{\text{s}}(t_1).$$

Note that this decrease of $A_i^{\text{B}}(t)$ after t_2 is dependent on AoI difference between the BS and source node i at t_1. So the amount of age decrease at the BS w.r.t. source node i when a sample completes its transmission has already been determined by AoI status at an *earlier* time slot, i.e., the time slot when the sample starts its transmission.

Suppose the transmission of a sample at source node i starts at time slot t. Denote $\Delta_i(t)$ as the AoI *outage* which is given by

$$\Delta_i(t) = A_i^{\text{B}}(t) - A_i^{\text{s}}(t). \quad (34)$$

At each time slot t, we will use $\Delta_i(t)$ to make a scheduling decision to transmit new samples.[4] The motivation is intuitive: serving the node with largest $\Delta_i(t)$ will offer the greatest relieve in reducing its AoI at the BS. However, $\Delta_i(t)$ alone is not sufficient to be the scheduling metric. Both the weight w_i and packet size L_i must also be taken into considerations, as shown in (5). Therefore, we propose to use $\sqrt{w_i/L_i} \cdot \Delta_i(t)$ as the scheduling metric. The source node with the largest value of $\sqrt{w_i/L_i}\Delta_i(t)$ will be selected for transmission and the BS will allocate as many transmission units as available to transmit this sample before considering others.[5] The key steps of Juventas are shown in Fig. 4.

Juventas Algorithm
1: For each time slot t, do the following:
2: Complete transmission of the un-completed sample from the previous time slot (if there is any).
3: Among all other source nodes, find node i with the largest $\sqrt{w_i/L_i} \cdot \Delta_i(t)$.
4: If L_i is less than or equal to the number of remaining transmission units, complete transmission of this sample. Go to Step 3.
5: If L_i is greater than the number of remaining transmission units, transmit this sample incompletely with all remaining transmission units.

Fig. 4. Key steps of Juventas algorithm.

Note that by design, Juventas is an order-based scheduling algorithm. It can also be shown that Juventas is a cyclic algorithm when sampling is periodic. We omit its proof due to paper length limitation.

The following theorem offers a performance guarantee of Juventas (with a factor 3) when $L_i \leq M$.[6]

Theorem 1 *Under arbitrary sampling, if $L_i \leq M$ for each source node i, $\bar{A}^B$ under Juventas scheduling algorithm is upper bounded by*

$$\bar{A}^B \leq 3\bar{A}^* + \sum_{i=1}^{N} w_i \quad (35)$$

where $\bar{A}^*$ is the optimal objective at the BS.

A Proof Sketch First, we show an interesting property about $\Delta_i(t)$. Denote $y_i(t)$ as a binary indicator on whether or not source node i starts to transmit a sample at time slot t. For each source node i, when $T \to \infty$, the sum of AoI increase and AoI decrease at the BS balances out, and we can prove

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} y_i(t)\Delta_i(t) = 1. \quad (36)$$

For $L_i \leq M$, under Juventas, each sample will finish its transmission within no more than 2 time slots. We define

[3]Juventas is the ancient Roman goddess for youth and rejuvenation.

[4]For a sample that is not finishedin the previous time slot, Juventas will use as many transmission units as needed in the current time slot to complete it (before allocating transmission units to start new samples), as shown in Fig. 4.

[5]Our idea is corroborated by the scheduling algorithm in [3] under per time slot scheduling with L_i and $M = 1$. The authors developed a near-optimal scheduling algorithm that allocates transmission rate in proportional to $\sqrt{w_i}$. Incidentally, Juventas performs better than this scheduling algorithm even in the same simple case with $M = 1$, $L_i = 1$ and per time slot sampling. This is because we make scheduling decision in each time slot while the one in [3] makes global scheduling decision at $t = 0$.

[6]The condition $L_i \leq M$ can be easily justified in the real world where the sample taken from a source node (e.g., sensor) is almost always smaller than the cellular transmission rate in a TTI (M).

$d^{\rm c}_{il}(t)$, $d^{\rm p}_{il}(t)$ and $d^{\rm n}_{il}(t)$ as binary variables indicating whether the l-th unit from node i starts its transmission, finishes its previous transmission or isn't transmitting in time slot t. From (15), we have

$$\sum_{i=K+1}^{N}\sum_{l=1}^{L_i}(d^{\rm c}_{il}(t)+d^{\rm p}_{il}(t)) \geq M_K, \quad \text{for } t>0. \tag{37}$$

For each node i, considering (36), we can prove

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{l=1}^{L_i}\left(d^{\rm c}_{il}(t)\Delta_i(t)+d^{\rm p}_{il}(t)\Delta_i(t-1)\right)=L_i. \tag{38}$$

For each node $1\leq i\leq N$, we define

$$s_i(t)=A^{\rm B}_i(t+1)-A^{\rm s}_i(t-2)-3. \tag{39}$$

By the design of Juventas, we can prove that for any i,j,l,t,

$$d^{\rm c}_{jl}(t)\sqrt{\frac{w_j}{L_j}}\Delta_j(t)\geq d^{\rm c}_{jl}(t)\sqrt{\frac{w_i}{L_i}}s_i(t), \tag{40}$$

and

$$d^{\rm p}_{jl}(t)\sqrt{\frac{w_j}{L_j}}\Delta_j(t-1)\geq d^{\rm p}_{jl}(t)\sqrt{\frac{w_i}{L_i}}s_i(t). \tag{41}$$

Combining (40) and (41) at each time slot t for each node i, we can prove

$$s_i(t)\leq\frac{\sum\limits_{j=K+1}^{N}\sqrt{\frac{w_j}{L_j}}\sum\limits_{l=1}^{L_j}\left(d^{\rm c}_{jl}(t)\Delta_j(t)+d^{\rm p}_{jl}(t)\Delta_j(t-1)\right)}{\sqrt{\frac{w_i}{L_i}}\sum\limits_{j=K+1}^{N}\sum\limits_{l=1}^{L_j}\left(d^{\rm c}_{jl}(t)+d^{\rm p}_{jl}(t)\right)}. \tag{42}$$

Combining (42) with (37) and (38), we can prove

$$\bar{A}^{\rm B}_i\leq\bar{A}^{\rm s}_i+3+\sqrt{\frac{L_i}{w_i}}\frac{\sum_{j=K+1}^{N}\sqrt{w_jL_j}}{M_K}. \tag{43}$$

Considering (5), (16) and (43), we can prove

$$\begin{aligned}\bar{A}^{\rm B}&\leq\frac{1}{M_K}(\sum_{i=K+1}^{N}\sqrt{w_iL_i})^2+\sum_{i=K+1}^{N}w_i(\bar{A}^{\rm s}_i+3)\\&+\sum_{i=1}^{K}w_i(\bar{A}^{\rm s}_i+4)=2\alpha_{(\text{PTS},M)}+\alpha_{(\text{ARB},\infty)}+\sum_{i=1}^{N}w_i\\&\leq3\alpha_{(\text{ARB},M)}+\sum_{i=1}^{N}w_i\leq3\bar{A}^*+\sum_{i=1}^{N}w_i.\end{aligned}$$

VII. Simulation Results

In this section, we evaluate the performance of Juventas. In our simulation, we assume periodic sampling (including per time slot sampling) at each node. For the source nodes, we assume they can be classified into 10 groups, with each group containing the same number of source nodes. The weight, sampling rate, and sample size for the source nodes within the same group are identical but differ from those in a different group. The weight of each source node is normalized w.r..t $\sum_{i=1}^{N}w_i$ before each simulation.[7] For each simulation, we terminate when the BS has received at least 100 samples from each source node. Then we calculate $\bar{A}^{\rm B}$.

TABLE I
Simulation Parameters

Type	1	2	3	4	5	6	7	8	9	10
w_i	6	33	25	39	36	46	35	17	35	10
L_i	1	15	11	10	19	13	13	18	17	12
T_i	10	12	45	2	25	9	49	36	26	24
M	50									
N	100									

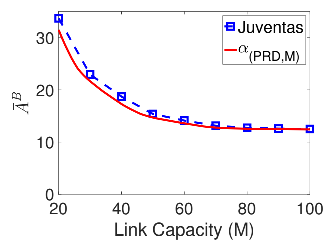


Fig. 5. $\bar{A}^{\rm B}$ for varying M

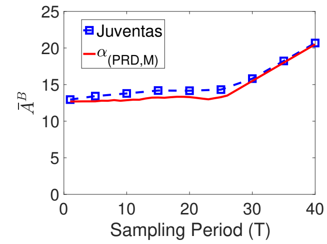


Fig. 6. $\bar{A}^{\rm B}$ for varying T

(i) With the parameter settings w_i, L_i, T_i, N given in Table I (all units normalized), Fig. 5 shows the objective value, $\bar{A}^{\rm B}$ by Juventas, as a function of increasing link capacity M. We see that $\bar{A}^{\rm B}$ decreases monotonically as M increase. Also shown in this figure is the lower bound for periodic sampling $\alpha_{(\text{PRD},M)}$ that we derived in (31). Clearly, we see that Juventas can achieve near-optimal performance.

(ii) With the parameter settings w_i, L_i, M, N given in Table I, Fig. 6 shows the objective value, $\bar{A}^{\rm B}$, as a function of increasing sampling cycle T, which is the same for all source nodes. The lower bound $\alpha_{(\text{PRD},M)}$ is also shown in this figure. We find that Juventas can achieve near-optimal performance. Note that $f(p_i)$ in (29) is a piecewise function so $\alpha_{(\text{PRD},M)}$ doesn't increase monotonically as T increases.

(iii) With the parameter settings w_i, T_i, M, N given in Table I, Fig. 7 shows the objective value, $\bar{A}^{\rm B}$, as a function of increasing sample size L. Here, all source nodes have the same sample size L. We see that $\bar{A}^{\rm B}$ by Juventas increases monotonically as L increases. The lower bound $\alpha_{(\text{PRD},M)}$ is also shown in this figure, and we see that Juventas can achieve near-optimal performance.

(iv) With the parameter settings w_i, L_i, T_i, M given in Table I, Fig. 8 shows the objective value, $\bar{A}^{\rm B}$, as a function of increasing number of source nodes N. We see that $\bar{A}^{\rm B}$ for Juventas increases monotonically as N increases. The lower bound $\alpha_{(\text{PRD},M)}$ is also shown in this figure, and we see that Juventas can achieve near-optimal performance.

Finally, we explore the impact of synchronization in sampling on the performance of Juventas. If two source nodes have the same sampling cycle T_i and the same initial state, $A^{\rm s}_i(0)$, we say they are synchronized. In all the simulations in (i) to (iv), the source nodes are not synchronized, either with different sampling rates or different initial states. We now

[7]Note that we vary the value N in some of the experiments.

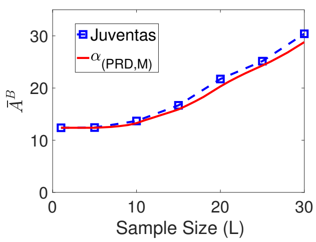


Fig. 7. $\bar{A}^{\mathrm{B}}$ for varying L

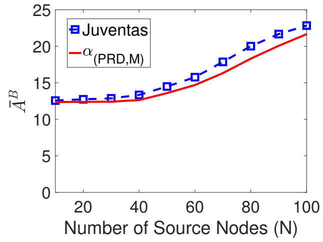


Fig. 8. $\bar{A}^{\mathrm{B}}$ for varying N

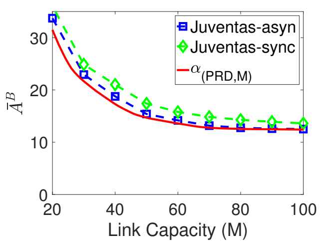


Fig. 9. Weak Synchronization

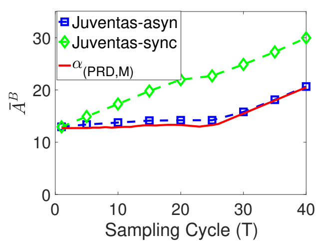


Fig. 10. Strong Synchronization

study the impact of synchronization. In the first scenario, we consider synchronization only within each type of nodes (weak synchronization). In the second scenario, we consider synchronization among all source nodes (strong synchronization).

Fig. 9 shows the results under weak synchronization (with the same parameter settings as in Fig. 5). We see that the objective $\bar{A}^{\mathrm{B}}$ under weak synchronization is slightly larger than that under no synchronization. Fig. 10 shows the results under strong synchronization (with the same parameter settings as in Fig. 6). We see that the objective $\bar{A}^{\mathrm{B}}$ under strong synchronization is considerably larger than that under no synchronization. Based on the results in Fig. 9 and Fig. 10, we conclude that synchronization is harmful to AoI performance and should be avoided or minimized when we initialize the source nodes.

VIII. Conclusions

Minimizing AoI is an important objective in data collection. However, most of existing research on minimizing the AoI is based on overly simplified models that are quite far from real world applications. In this paper, we addressed this important issue by generalizing three key aspects in AoI research: sampling behavior, sample size, and transmission capacity. Under the general setting, we developed two interesting properties to reduce the search space and derived tight lower bounds for an optimal solution. Further, we developed Juventas, a low-complexity scheduling algorithm that was shown to offer near-optimal performance when there is no synchronization among the source nodes and have a guaranteed performance (within a factor of 3).

Acknowledgment

This research was supported in part by NSF Grant CNS-1617634.

References

[1] S. Kaul, M. Gruteser, V. Rai and J. Kenney, "Minimizing Age of Information in Vehicular Networks," in *Proc. IEEE SECON,* pp. 350–358, Salt Lake City, UT, USA, June 27–30, 2011.

[2] S. Kaul, R. Yates and M. Gruteser, "Real-Time Status: How Often Should One Update?" in *Proc. IEEE INFOCOM,* pp. 2731–2735, Orlando, FL, USA, Mar. 25–30, 2012.

[3] R. Yates, P. Ciblat, A. Yener and M. Wigger, "Age-Optimal Constrained Cache Updating," in *Proc. IEEE ISIT,* pp. 141–145, Archen, Germany, June 25–30, 2017.

[4] I. Kadota, E. Uysal-Biyikoglu, R. Singh and E. Modiano, "Minimizing the Age of Information in Broadcast Wireless Networks," in *Proc. Allerton Conference,* pp. 844–851, Monticello, IL, USA, Sept. 27–30, 2016.

[5] I. Kadota, A. Sinha and E. Modiano, "Optimizing Age of Information in Wireless Networks with Throughput Constraints," in *Proc. IEEE INFOCOM,* pp. 1844–1852, Honolulu, HI, USA, Apr. 16–18, 2018.

[6] R. Talak, S. Karaman and E. Modiano, "Optimizing Age of Information in Wireless Networks with Perfect Channel State Information," in *Proc. WiOpt,* pp. 1–8, Shanghai, China, May 7–11, 2018.

[7] R. Talak, S. Karaman and E. Modiano, "Optimizing Information Freshness in Wireless Networks under General Interference Constraints," in *Proc. ACM MobiHoc,* pp. 61–70, Los Angeles, CA, USA, June 26–29, 2018.

[8] Y. Hsu, E. Modiano, and L. Duan, "Age of Information: Design and Analysis of Optimal Scheduling Algorithms," in *Proc. IEEE ISIT,* pp. 561–565, Archen, Germany, June 25–30, 2017.

[9] S.K. Kaul and R.D. Yates, "Status Updates over Unreliable Multiaccess Channels," in *Proc. IEEE ISIT,* pp. 331–335, Archen, Germany, June 25–30, 2017.

[10] R.D. Yates, "Lazy Is Timely: Status Updates by an Energy Harvesting Source," in *Proc. IEEE ISIT,* pp. 3008–3012, Hong Kong, China, June 14–19, 2015.

[11] B.T. Bacinoglu, E.T. Ceran and E. Elif Uysal-Biyikoglu, "Age of Information under Energy Replenishment Constraints," in *Proc. Information Theory and Applications Workshop,* pp. 25–31, San Diago, CA, USA, Feb. 1–6, 2015.

[12] A. Kosta, N. Pappas, A. Ephremides and V. Angelakis, "Age and Value of Information: Non-Linear Age Case," in *Proc. IEEE ISIT,* pp. 326–330, Archen, Germany, June 25–30, 2017.

[13] A.M. Bedewy, Y. Sun and N.B. Shroff, "Age-Optimal Information Updates in Multihop Networks," in *Proc. IEEE ISIT,* pp. 576–580, Archen, Germany, June 25–30, 2017.

[14] Y. Sun, E. Uysal-Biyikoglu, R.D. Yates, C.E. Koksal and N.B. Shroff, "Update or Wait: How to Keep Your Data Fresh," *IEEE Trans. on Information Theory,* vol. 63, issue 11, pp. 7492–7508, Nov. 2017.

[15] A.M. Bedewy, Y. Sun and N.B. Shroff, "Optimizing Data Freshness, Throughput, and Delay in Multi-Server Information-Update Systems," in *Proc. IEEE ISIT,* pp. 2569–2573, Barcelona, Spain, July 10–15, 2016.

[16] C. Joo and A. Eryilmaz, "Wireless Scheduling for Information Freshness and Synchrony: Drift-based Design and Heavy-Traffic Analysis," in *Proc. WiOpt,* pp. 1–8, Paris, France, May 15–19, 2017.

[17] N. Lu, B. Ji and B. Li, "Age-based Scheduling: Improving Data Freshness for Wireless Real-Time Traffic," in *Proc. ACM MobiHoc,* pp. 191–200, Los Angeles, CA, USA, June 26–29, 2018.

[18] Y. Sun, "A Collection of Recent Papers on the Age of Information," http://www.auburn.edu/\%7eyzs0078/ [Online; accessed on 2018-10-20].