Stable Age of Information Scheduling with NOMA in Edge Networks

Yifei Zou, *Member, IEEE,* Peng Wang, Shikun Shen, Dongxiao Yu, *Senior Member, IEEE,* Jorge Torres Gómez, *Senior Member, IEEE,* Falko Dressler, *Fellow, IEEE,* Xiuzhen Cheng, *Fellow, IEEE*

Abstract—In this paper, we study a stable Age-of-Information (AoI) scheduling problem to handle the massive packet aggregation from arbitrary total of n end devices to an edge device via a single hop wireless channel when information constantly arrives at the end side. Specifically, we consider the arrivals of the information with an *online* injection mode, i.e., information is injected to end devices with an unknown injection rate in each time slot. After the information is injected, the AoI with respect to those end devices constantly increases until their fresh messages are received by the edge device, which characterizes the freshness of the information at destination. Based on the online injection mode, we propose the first distributed stable AoI scheduling algorithm combining NOMA (Non-Orthogonal Multiple-Access) technique in this paper, to minimize the expected average peak AoI (EAP-AoI) of our end-to-edge information system. Adopting NOMA technique enables k messages decoded from a mixed signal by the edge device in our algorithm, with parameter $k > 1$. We prove that our algorithm is stable even under the asymptotically maximum injection rate of $O(k/n)$ that any stable AoI scheduling algorithm may handle, and the EAP-AoI of our information system is bounded by $O(\sqrt[3]{nk})$ time slots under the injection rate of $O(k/n)$. Comparing with two existing results, the EAP-AoI in our algorithm is $O(\frac{n^{2/3}}{k^{4/3}})$ and $O(\frac{n^{2/3}}{k^{1/3}})$ times smaller. Numerical results also verify the stability and efficiency of our algorithm.

Index Terms—Distributed Algorithm, Stable Age-of-Information Scheduling, NOMA Technology, SINR Model.

1 Introduction

In recent years, the wide deployment of mobile devices and rapid development of wireless techniques have strongly supported a great amount of applications in edge networks, which heavily rely on time-sensitive information collected by sensors from the environment. To depict and quantify the freshness of the collected information in those applications, the concept of Age-of-Information (AoI) was introduced in [1], [2], [3]. Specifically, the AoI accounts for the elapsing time of received packets at destination since its generation at the source.

- *Y. Zou, P. Wang, S. Shen, D. Yu (Corresponding Author) and X. Cheng are with Institute of Intelligent Computing, School of Computer Science and Technology, Shandong University, Qingdao, 266237, P.R. China. E-mail: yfzou@sdu.edu.cn, wangpeng1102@163.com, shikunshen@mail.sdu.edu.cn, dxyu@sdu.edu.cn, xzcheng@sdu.edu.cn*
- *J. T. Gómez and F. Dressler are with the School of Electrical Engineering and Computer Science, TU Berlin, Berlin, 10587, Germany. E-mail: {torres-gomez, dressler}@ccs-labs.org .*

Manuscript received MM DD, YYYY; revised MM DD, YYYY.

Since AoI gives priority to the updates that can greatly reduce the information time lag at the destination, the scheduling on the basis of AoI can significantly improve the freshness of information in many applications and information systems [4], [5], [6].

A common approach to minimize the AoI in edge networks operates under a *generate at will* mode [7] and a centralized manner [8]. In this setup, end devices—also referred to as sensor nodes or nodes throughout this paper—periodically gather information from their environment using their sensors and report the freshest data to the designated edge device via centralized scheduling algorithms, such as the widely adopted Time-Division Multiple Access (TDMA) scheme. Then, the edge device estimates the content of the aggregated information by a filter and picks out the useful information for high-level applications. In this scheduling mode, its advantage is that the total packets to be scheduled in each period is predictable and even fixed, since all sensors will report their collected information regularly. Thus, it is feasible to design elegant and efficient AoI scheduling algorithms based on the multi-access channel [9], [10], [11]. Meanwhile, since all collected information are aggregated to the edge device regardless of their importance, a fraction of network resources, such as the bandwidth and the energy resource, will be misused to transmit useless information. Besides, a centralized filter results in a high requirement on the energy and computing resource.

To overcome these disadvantages, this paper considers an alternative framework in which the filters are deployed on each of the sensor nodes. Once some information is collected by a node, it will firstly be estimated by the distributed filter. Only the important information will be scheduled to the edge device. Considering the reported performance on the use of distributed filter [12], in our work we focus on how the filtered information arrives at the edge side as fresh as possible. Figure 1 is an illustration for our alternative framework, in which only the important/useful information will be transmitted through the wireless channel. Compared with the previous framework, a wider bandwidth can be provided to schedule the useful information, which makes it possible to design more efficient scheduling algorithms.

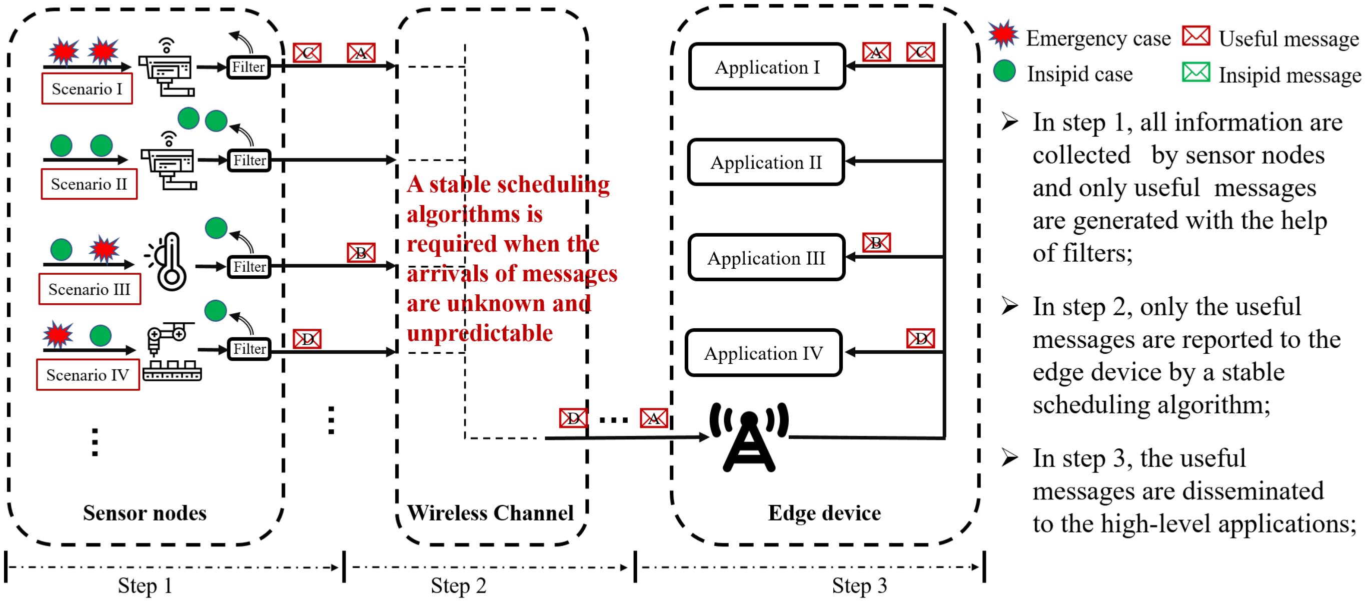


Fig. 1: Illustration of our alternative framework on information aggregation

Even though the new framework reduces the workload on message aggregation by only scheduling the useful information, a new challenge arises for designing stable and efficient scheduling algorithm: *when will the useful information arrive at the end side becomes unknown for both the edge and the end devices, and is totally determined by the environment.* Even though some methods can predict the arrival of useful information, it often requires additionally knowledge and is not comprehensive enough for general scenarios [13]. To depict the uncertain arrivals of the useful information in an information scheduling process, we formulate the AoI scheduling problem in an *online* mode and study a distributed stable AoI scheduling algorithm in this paper. Specifically, when the information collected by a sensor in a time slot t is useful, we say a packet with useful information is injected to the sensor node in t.[1] Then, we define the injection rate of a node in time slot t as the total of packets with useful information only in expectation in t, thus indicating how likely the useful information arrives at the node in a time slot. Also, in an interval of multiple time slots, each node may have multiple pieces of information injected while the distribution of injection is unknown. We consider a harsh assumption that all the injection of information, injection rate of nodes in each time slot, and the distribution of injections in any interval are unknown, unpredictable, independent and may vary for different nodes. The only knowledge can be used by nodes for the information scheduling is an upper bound of the injection rates among all nodes.

For our newly proposed AoI scheduling problem under the *online* injection mode, the previous scheduling algorithms no longer guarantee efficient performances, since most of them require packets to be scheduled periodically should be predictable and even fixed. To solve the AoI scheduling problem with the online information injections, a *stable* AoI scheduling algorithm should be designed with the following property: *for each sensor node, its Age-of-Information at the edge device and the piece of useful information not received by the edge device should always be kept at some low-level values, when information is constantly injected.*

The unpredictable injections and unknown injection rate prevent keeping the designed scheduling algorithm stable. For example, too much information injected in a short period may temporarily overwhelm a scheduling algorithm. Thus, the piece of information not received by the edge device and the AoI w.r.t (with respect to) the end nodes increases if no targeted operation is given. Also, the OMA (Orthogonal Multiple-Access) technique adopted in previous works has become a bottleneck on improving the efficiency and stabilization of AoI scheduling since in OMA, a device can at most receive one message in each time slot. To overcome those challenges, a two-stage scheduling combining with NOMA (Non-Orthogonal Multiple-Access) technique is proposed in this paper, the first stage of which is to schedule the recently injected information, while the second stage of which is targeted for the information with a relative large age but still not received by the edge device. Both of the two stages are facilitated by NOMA technique that actualizes multiple messages decoded from a mixed signal, which dramatically make the AoI scheduling process fast and stable even under a high injection rate. By doing this, it ensures that our scheduling algorithm is stable and efficient when facing the unpredictable injections of the useful information.

Contributions. Our paper considers a realistic and harsh information aggregation scenario in an end-to-edge information system, in which the useful information is constantly injected at the end side with an un-

1. Since only useful information will be reported by sensors, in the following, when the word *information* is mentioned in our work, it indicates the useful information.

known and unpredictable pattern. We firstly formulate such scenario by proposing a *stable* property for online AoI scheduling problem. Then, a NOMA-facilitated stable AoI scheduling algorithm is presented with a competitive performance in terms of the freshness of information and the stability of scheduling process. Our main contributions are summarized as follows:

- We present a stable AoI scheduling pattern, in which packets with useful information only are constantly injected to the end nodes with unpredictable injections and unknown injection rates. Compared with previous works that regularly aggregate all information from end nodes regardless of the content, our stable AoI scheduling prevents network resource from transmitting useless information.
- A distributed stable AoI scheduling algorithm with NOMA is proposed, to schedule the constantly injected information from n end nodes to an edge device via a single hop wireless channel. Even when the injection rate gets close to the maximum injection rate of $O(k/n)$ that any stable AoI scheduling algorithm may handle with a constant factor, our algorithm can still keep stable and ensure that the edge nodes have their expected average peak AoI bounded within $O(\sqrt[3]{nk})$ time slots. The parameter k in NOMA technique means the edge device can at most decode k messages from a mixed signal. When comparing with two existing results: $O(n/k)$ time slots for a centralized greedy scheme without considering the information injection and $O(n)$ time slots for the scheduling works without NOMA in [14], [15], our algorithm is $O(\frac{n^{2/3}}{k^{4/3}})$ and $O(\frac{n^{2/3}}{k^{1/3}})$ times faster. Extensive simulation results confirm the stability and efficiency of our algorithm.

Roadmap. The remaining parts of this paper are organized as follows. Section 2 introduces the related work. Section 3 formulates our network model and the relative definitions in our online information scheduling problem. The stable algorithm and theoretical analysis are presented in Section 4 and Section 5, respectively. Section 6 shows the simulation results, and Section 7 concludes our paper.

2 Related Work

To optimize the AoI minimization problem, a series of scheduling schemes have been proposed in the past decades, including [16], [17], [18] in single hop wireless networks, [19], [20], [21] in multi-hop wireless networks, and [6], [8], [22] directly in edge networks to schedule the information between the sensors on the end side, the edge servers and the cloud servers. However, a large fraction of the previous works on the AoI optimization problem, as far as we know, are designed in a centralized framework and rely on the reliable communication assumption. Some known results for distributed AoI scheduling are [7], [23] on graph interference model, [11] on physical interference model, and [24] on a channel failure model caused by the dynamic channel access attack. The most relevant works considering the similar online AoI optimization problem include [6], [8], [22], [25], [26]. In [6], [8], [22], [25], the packets are constantly generated by a single sensor/user and delivered to an edge server. The work in [26] extends the online AoI optimization problem to a broad scenario in which the data packets are scheduled from multiple sensors to an edge server. Based on the Lyapunov optimization theory, a centralized max-weight policy is proposed to minimize the expected weighted sum AoI of the information system. Compared with the existing works, especially with the newest [25], [26], our work outperforms in the distributed framework, the combination of NOMA technique, and the realistic NOMA-SINR communication model, respectively.

TABLE 1: Related works on AoI scheduling

Reference	Environment	Dis.	NOMA	Online
[16], [17], [18]	single hop	X	X	X
[19], [20], [21]	multi-hop	X	X	X
[7], [11], [23], [24]	edge network	√	X	X
[10], [27], [28], [29]	single hop	X	√	X
[6], [8], [22], [25], [26]	edge network	X	X	√
this work	edge network	√	√	√

As a significant tool to improve communication performance, the NOMA technique is not typically reported to minimize the AoI of downlinks [10], [27], [28] and uplinks [29] in edge networks and relative scenarios. Specifically, to minimize the average AoI of multiple users, an adaptive buffer-aided scheme and a heuristic adaptation of the driftplus-penalty approach from the Lyapunov framework are used in [10] and [27], respectively. Besides, to minimize the expected weighted sum AoI, a low-complexity power allocation policy with NOMA is presented in [28]. Finally, in [29], the NOMA technique is used to minimize the uplinks from users to the base station and the result is compared with the AoI scheduling with OMA. However, all of the works mentioned above are the centralized ones with a two-layer NOMA technique, in which a receiver can at most decode 2 messages from a mixed signal. Thus, the efficiency on AoI scheduling by adopting NOMA in the above works at most has a 2-times improvement, compared with that without NOMA.

Table 1 is a summary for the above related works about AoI scheduling without/with NOMA technique. Compared with the existing works on AoI scheduling in edge networks, our work is the first one investigating the online AoI scheduling problem under distributed framework with NOMA technique.

3 Model and Problem Definitions

We consider our stable AoI scheduling problem with online information injection in a 2-dimensional Euclidean space, in which an edge device and n sensor nodes are arbitrarily deployed. By normalizing the minimum and maximum distance from sensor nodes to edge device as 1 and d separately, we assume that the edge device and sensor nodes have their transmission range $R > d$, i.e., all the edge device and sensor nodes are in a single hop wireless network. To evaluate time stamps, we assume nodes are synchronized, and define a time duration as a round, in which the nodes can collect information from the environment once through their sensors. Meanwhile, the duration in a round is enough for the edge device and sensor nodes to transmit or receive a message. Note that in our scheduling framework, the filter is distributed on each node. When a node v collects an information from the environment in a round t, the filter estimates the importance of the information according to its content. If the information is important or useful, a message $\mathcal{M}_v$ containing the useful information will be generated by v, and the scheduling on $\mathcal{M}_v$ starts from round $t+1$ until the edge device receives it. If the filter believes that the collected information does not contain any useful information, no message will be generated for it. By doing this, our model makes sure that the wireless channel only serves for the useful information.

When a message is generated by a node because of an important information collected, we also say the corresponding message is injected to the node. Note that for each node, the importance of the information to be collected in the next round is unknown and unpredictable. Thus, whether a message will be generated by a node in the coming round is also unknown and unpredictable. To capture the uncertainty in information/message injections, one intuitive way is to assume an *injection rate* λ on each node. Specifically, the injection rate of a node in round t is defined as the expected number of messages injected in the node in round t. The distribution of injection rate is unknown, unpredictable and independent among different nodes and rounds.

3.1 Stable Age-of-Information Scheduling Problem

To formulate the freshness of an important information when it arrives at the edge device, the concept *Age-of-Information* (AoI) has been defined in the following.

Definition 1 [30]: Let $\mathcal{A}_v(t)$ be the AoI with respect to the end device v at round t, which is the time that elapsed since the generation of the freshest message transmitted from v to the edge device, i.e.

$$\mathcal{A}_v(t) = t - \tau_v(t), \tag{1}$$

where $\tau_v(t)$ is the generation time of the freshest message received by the edge device from the node v at round t.

According to the above definition, $A_v(t)$ drops if a fresher message from v successfully arrives at the edge side at round $(t-1)$, and increases linearly otherwise. The fresher message from v means that the message has a fresher generation time than all the messages that was generated and delivered to the edge device by v in the previous rounds. Such a definition works well in the previous works in which the fresher messages are always prepared by the end devices in each round. However, if such a definition is directly adopted in our online injection mode, the AoI also increases when the most fresh message generated by v has been received by the edge device, but v no longer has any new important information injected and fresher message to deliver in the following rounds. Such an increasing of AoI is insignificant and will mislead the AoI scheduling strategy. A more detailed explanation is given in the Appendix. To avoid this case, we additionally assume that for each end device, it is in *active* state for AoI update if it has some fresher messages to deliver in current round. Otherwise, it is in *inactive* state. Boolean variable $s_v(t) = 0$ or 1 is used to indicate that an end device v is in inactive or active state in round t. Initially, all end devices are in inactive state because the important message has not been injected. We assume that $A_v(t) = 0$ when v is in inactive state. Let $m_v(t) = 1$ denote the event that a fresher message from v is received by the edge device in round t, and $m_v(t) = 0$ otherwise. Then, $A_v(t)$ evolves as follow:

$$\mathcal{A}_v(t) = \begin{cases} 0, & \text{if } s_v(t) = 0, \\ t - \tau_v(t), & \text{if } s_v(t) = 1 \text{ and } m_v(t-1) = 1, \\ \mathcal{A}_v(t-1) + 1, & \text{if } s_v(t) = 1 \text{ but } m_v(t-1) = 0. \end{cases} \tag{2}$$

Figure 2 illustrates an example to show the difference between the conventional AoI and our new defined AoI by Equation 2 when the important messages arrive in an online mode. Specifically, in the given example, the important information arrives at the node v at the beginning of rounds 3, 7, 9, and the corresponding messages are generated by v immediately. At the end of rounds 4, 8, 10, those messages are successfully received by the edge device. According to our definition for the active and inactive states w.r.t. AoI update, the node v is in active state within rounds $[3-4]$ and $[7-10]$. Then, we have the curves in Figure 2 (a) and (b) to illustrate the conventional AoI and our new defined AoI w.r.t. node v, respectively. In Figure 2 (a), the AoI of node v keeps increasing even though v is in inactive state and $\mathcal{A}_v(8) = 8 - 3 = 5$. Meanwhile, in Figure 2 (b), by setting $\mathcal{A}_v(t) = 0$ when node v is in inactive state, we make sure that our new defined AoI accurately depicts the elapsed time since the generation of the freshest message when v is in active state.

Minimize the Expected Average Peak AoI. From Figure 2 (b), we can see that the peak age w.r.t. to an end node v is achieved just before a fresher message from v successfully arriving at the edge side. Besides, in the last round of an interval I, the last peak age w.r.t. the end node v in I is achieved if v is active at that moment. Thus, the average peak age w.r.t the end node v in an interval I that consists of $|I|$ rounds is defined as

$$\hat{\mathcal{A}}_v(I) = \frac{\sum_{t=1}^{|I|-1} \mathcal{A}_v(t) m_v(t) + \mathcal{A}_v(|I|) s_v(|I|)}{\sum_{t=1}^{|I|-1} m_v(t) + s_v(|I|)}. \tag{3}$$

In the above equation, $m_v(t) = 1$ indicates the event that a fresher message from v is received by the edge device in round t and the age of v reaches to a peak

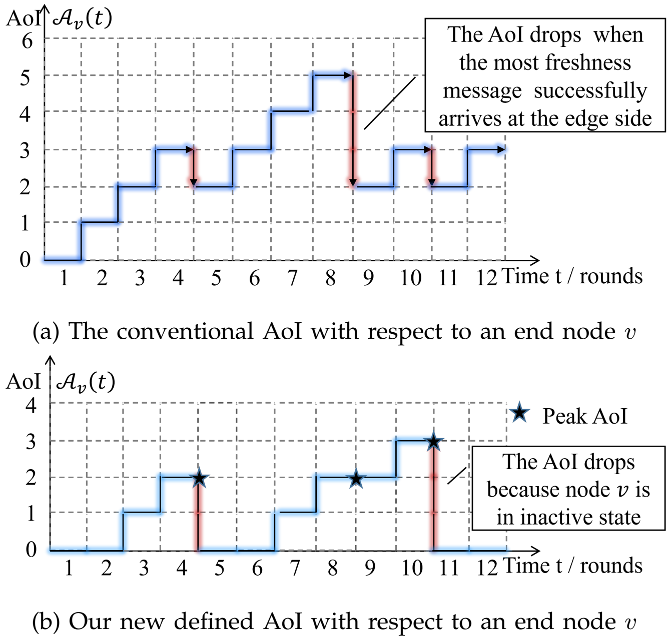


Fig. 2: The conventional AoI and our new defined AoI in an online injection mode when the important packets are generated by v at the beginning of rounds $\{3, 7, 9\}$, and received by the edge node at the end rounds $\{4, 8, 10\}$.

meanwhile. $s_v(|I|) = 1$ means the end node v is active at the last round of the interval I, in which the peak age of v is also reached. Thus, $\sum_{t=1}^{|I|-1} \mathcal{A}_v(t)m_v(t)+\mathcal{A}_v(|I|)s_v(|I|)$ is the sum of the peak ages w.r.t. the end node v in the interval I and $\sum_{t=1}^{|I|-1} m_v(t) + s_v(|I|)$ is the number of times that v reaches to its peak age in the interval I.

Therefore, the average peak age of the whole end-to-edge information system in an interval I is defined as

$$\hat{\mathcal{A}}(I) = \frac{\sum_{v\in V}\left(\sum_{t=1}^{|I|-1} \mathcal{A}_v(t)m_v(t) + \mathcal{A}_v(|I|)s_v(|I|)\right)}{\sum_{v\in V}\left(\sum_{t=1}^{|I|-1} m_v(t) + s_v(|I|)\right)}, \tag{4}$$

in which V is the set of n end devices. Let $\mathbb{E}[\hat{\mathcal{A}}(I)]$ be the expected value of $\hat{\mathcal{A}}(I)$, i.e. the expected average peak AoI (EAP-AoI) of the whole information system in I.

The objective of our AoI scheduling problem is a distributed scheduling policy to *Minimize the Expected Average Peak Age-of-Information* (MEAPA) of our stable information system in an interval I when the important information constantly arrives. In each round of the algorithm, each of the node can transmit its message with a specific transmission power or switch to listen for the messages in the wireless channel. We use the SINR rate $SINR(\mathcal{M}, t) \geq \beta$ to formulate the fact that a message $\mathcal{M}$ can be decoded by its receiver in a round t, the detail of which is given in the following NOMA-SINR model. The final outcome of our policy is that the fresh messages from the end nodes can be timely received by the edge node, so that the EAP-AoI of the whole end-to-edge information system can be minimized. Besides, since all the messages generated by the end nodes are important, an additional constraint is that all the messages from the end nodes should be finally received by the edge node if the interval I contains sufficient rounds.

Aiming at the MEAPA makes sense in our AoI scheduling problem, since the age of end devices highly reflects the *freshness* of information. Minimizing EAP-AoI provides a worst case guarantee on the AoI scheduling from a statistical view. When the EAP-AoI gets minimized, we can strongly believe that the information aggregated to the edge device is fresh. Note that the study in [31] has shown that optimizing the scheduling strategy on minimizing AoI in wireless systems is NP-hard, without considering NOMA and online arrivals of important information. Compared with the problem in [31], the NOMA technique and online arrivals of important information result in additional challenges for the scheduling. Thus, our problem in this paper is at least as hard as the problem in [31], which is also NP-hard.

3.2 NOMA-SINR Model

The NOMA technique is adopted in our scheduling algorithm to facilitate the transmissions in wireless channel. Specifically, it allows multiple messages decoded in one round from a mixed signal, as long as a specific interference cancellation sequence of the messages can be satisfied. A detailed description is given in the following.

For convenience, the edge and end devices who *transmit* or *listen* are denoted as *transmitters* and *receivers*, respectively. $S(u, v)$ is defined as the signal from the transmitter u to the receiver v. In traditional SINR model without NOMA, the receiver can at most decode one signal in a round with a requirement that the strength of the decoded signal must be larger than the sum of the strengths of other signals (i.e., interference) and the ambient noise. However, the NOMA technology open a new door for the receiver to recover multiple signals from the mixed signal in one round, as long as an interference cancellation sequence of the signals can be satisfied. Specifically, for a receiver v, if there is a signal $S(u, v)$ whose strength is larger than the sum of the interference and the ambient noise, v can decode the signal $S(u, v)$ from the the mixed signal. Then, the interference from $S(u, v)$ can be cancelled when v tries to decode other signals. The following NOMA-SINR equation

$$SINR(u, v) = \frac{P_u/d(u, v)^\alpha}{\sum_{w\in W\setminus(W'\cup\{u\})} P_w/d(w, v)^\alpha + N} \tag{5}$$

shows how the interference is accumulated and cancelled in transmissions, and in which situation will the transmissions succeed. In the above NOMA-SINR equation, W is the set of transmitters in current round, and W' is the set of transmitters whose signals have already been decoded by v. For each transmitter w, P_w is the transmission power, and $d(w, v)$ is the Euclidean distance between w and v. α is the path-loss exponent, and N is the ambient noise, both of which are constants determined by the environment. When $SINR(u, v) \geq \beta$,

v can decode the signal from u, where β is a threshold determined by the hardware of receivers. In usual, $\alpha \in (2, 6]$ and $\beta > 1$. In our work, we assume that all communications in the single hop wireless network have the same SINR parameters α and β. For convenience, the SINR rate of the message $\mathcal{M}$ from transmitter u to receiver v in round t is also written as $SINR(\mathcal{M}, t)$.

The communication parameters such as distance, transmit power, and fading characteristics in the above NOMA-SINR model determines the performance of NOMA on the following two aspects. The first one is whether a signal can be decoded by the receiver. The second one is the amount of data delivered by the signal if it can be decoded. In our paper, since the sensor nodes behave individually, their communications may collide and interfere with each other, i.e., the communication is no longer reliable. Thus, our paper focus on the first aspect: whether the signal can be decoded under the communication parameters. As for the bandwidth of communication, we assume that it is enough for sensor nodes to deliver its packet within a time slot.

Additional Assumptions. We assume that all the end devices have the uniform maximum and minimum transmission power: P_{max} and P_{min}. The edge device has the stronger hardware and energy supply, so that its maximum transmission power can be larger than P_{max}. In each transmission, the edge device and nodes can choose a transmission power within the scope of their maximum and minimum transmission power to transmit a message. Defining R as the transmission range of the edge device and nodes, and since all devices are in a single hop wireless network, we have $R \leq (P_{min}/\beta N)^{1/\alpha}$. Otherwise, the transmissions with power P_{min} may fail and makes no sense. Also, in NOMA technique, the ratio of maximum/minimum transmission power and the hardware are two of the bottlenecks for multiple-message decoding from a mixed signal. In this paper, we assume that the ratio of maximum/minimum transmission power is sufficient large to support k message-decoding, together with the hardware on edge device. A detailed introduction for the k-layer NOMA technique can be found in [32]. Three queues are used by each node to store its messages. We assume that all messages generated by n sensor nodes have the same size, and the length of queues on each node is sufficient enough in our algorithm. Also, each sensor node v has its unique ID. The NOMA-SINR parameters α, β, k, the number of sensor nodes n, and the upper bound on the injection rate λ is assumed to be known by all nodes. A table of notations presented in model, algorithm design and analysis is given in Table 2.

TABLE 2: Notations in model, algorithm and analysis

Notations	Definitions
V, n	Set/number of nodes in network, n is sufficient large
$\mathcal{A}_v(t)$	AoI w.r.t. the end node v at round t
$\hat{\mathcal{A}}_v(I)$	average peak AoI w.r.t. the end node v in interval I
$\hat{\mathcal{A}}(I)$	average peak AoI w.r.t the information system in interval I
$\mathbb{E}(X)$	the expected value of a variable X
P_{max}, P_{min}	Maximum/Minimum transmission power of end devices
ID_w	Unique ID of the node w
P_w	Transmission power of the node w, $P_w \in [P_{min}, P_{max}]$
k	Maximum number of decoded signals in NOMA
α, β	SINR parameters, $\alpha \in (2, 6]$ and $\beta > 1$
N	Ambient noise, determined by the environment
R	Transmission range, $R \leq (P_{min}/\beta N)^{1/\alpha}$
$S(u, v)$	Signal form the transmitter u to the receiver v
$SINR(u, v)$	SINR rate of the signal $S(u, v)$ at receiver v
$\mathcal{M}_v$	Message $\mathcal{M}$ generated by node v
$SINR(\mathcal{M}, t)$	SINR rate of the message $\mathcal{M}$ at round t
$signal(v)$	Strength of the signal sensed by the node v
λ	Injection rate of the important information on a node
S_v, Q_v, F_v	Three queues in the node v
ϵ, μ, c, γ	Constants in algorithm and analysis; $\epsilon \in (0, 1)$, $\mu = 6$, $c > \frac{\log R}{\log n}$, and $\gamma \geq \max\{1, \frac{c\alpha+1+\log\beta}{\mu}\}$
T, H, J	Parameters in algorithm and analysis; $T = \lfloor \sqrt[3]{nk} \rfloor$, $H = \lfloor (1-\epsilon^2)T \rfloor$, and $J = T - H$

4 Stable AoI Scheduling Algorithm

4.1 Challenges and Solutions in Algorithm Design

In previous works about AoI scheduling in edge networks and relative wireless areas, n nodes always try to deliver their freshest messages to the edge device in a centralized pattern without considering the importance of their collected information in each round. While in our stable AoI scheduling problem, each node firstly estimates the importance of its collected information and will only generate a message to schedule when the collected information is important. Thus, here comes a question about the significance of our algorithm design: *Is it necessary to design a new algorithm for the stable AoI scheduling problem with information online injection, rather than choosing an alternative one from the previous works?* Our answer is *Yes*, because the stable AoI scheduling is harder than the previous ones without considering the information injection. Besides, the solutions in previous works no longer suit because of the following challenges.

The biggest challenge is taken by the unknown injection rate and the unpredictable message injection when the important information arrives at the end side in an online pattern. In previous works, the number of messages to be scheduled in each round is predictable, known or even fixed. However, in our stable AoI scheduling problem, since the injection rate is unknown, the message injection and number of messages to be disseminated in each round is unpredictable. This results in difficulties from reported solutions to handle the contention and interference in wireless channel, and directly results in inefficient transmissions in a wireless channel between the edge device and sensor nodes. In general, the previous works are more likely to consider the AoI scheduling in an offline mode. While our problem is how to keep a stable scheduling in an online mode.

The second challenge is balancing the newly injected messages with the accumulated ones not received by the edge device. Unlike existing works that focus solely on scheduling the freshest messages and dropping stale ones to minimize AoI, our work aims to optimize the expected average peak AoI while ensuring that all important information selected by the local filter is delivered to the edge server, even if it is accumulated on the end side and outdated. Too many messages injected in a short period can temporarily overwhelm a scheduling algorithm. As a result, messages not received by the edge device accumulate at the end nodes, losing their freshness. Balancing the trade-off between the immediate transmission of newly injected messages and the efficient scheduling of accumulated messages is crucial, which ensures that no message is delayed excessively before being received by the edge device.

The third problem is how to tune the transmission power of each node in a distributed framework, to adapt the specific interference cancellation sequence in NOMA technique when signals arrive at the edge device. In a distributed framework, the lacking of a global cooperation makes it hard for each node to know the transmission powers chosen by other nodes in the coming round, and the distances from the other nodes to the edge device. So, it is nearly impossible for nodes to directly cooperate with each other to make sure the signals reach a geometric cancellation sequence with a tight ratio when they arrive at the edge device.

Solutions to Challenges. The Challenges one and two have arisen the additional requirements on our algorithm design. In our algorithm, we use three first-in-first-out queues in each node as the buffers for the messages injected in successive stages that consist of multiple rounds and start from stage 1. The queue S is used to store the messages injected in current stage i; the queue Q is used to schedule the messages injected in the last stage $i-1$; the queue F is used to schedule the messages accumulated in the stages 1, 2, $\cdots i-1$. By doing this, the message scheduling in our algorithm becomes independent with the injection rate and message injections in each round, but relates to the expectation of the injected messages in each stage, which may follow some distributions when the stage contains sufficient rounds. In a busy edge networks scenario with a high injection rate, the queues Q and F always contain enough messages injected in stages 1, 2, $\cdots i-1$. Thus, our scheduling algorithm on queues Q and F will obtain the high efficiency since they do not need to wait for the new injected messages. However, there are two trade-offs on each stage i that should be balanced.

- Trade-off I: if the stage i is too long, messages need to wait for a long time in S before it is scheduled by nodes in queues Q and F, which consequently increases the age of messages when they arrives at the destination. If the stage i is too short, when nodes start to schedule the messages in queues Q and F, it is likely that there are not enough messages in queues Q and F to be scheduled, which reduces the efficiency of our scheduling algorithm. The Equation 6 in Section 5 directly shows how we reach a balance in this trade-off.
- Trade-off II: in each stage, both of the fresh messages in the queue Q and the accumulated messages in the queue F should be scheduled. On one hand, if too many rounds are used to deliver fresh messages, the AoI is minimized but more outdated messages accumulate. On the other hand, if too many rounds are allocated to outdated messages, the AoI cannot be effectively optimized. In our algorithm, constant fraction of rounds in each stage are assigned to schedule the fresh messages in queue F and the remaining rounds are used to deliver the accumulated messages in queue S, which is proved to be an efficient approach.

As for implementing the NOMA technique to facilitate the transmissions in scheduling, nodes first simulate a coin tossing game to reduce the contention in the incoming transmissions. Then, a sparse mapping scheme between the nodes and their transmission powers are given. Thus, when the signals arrive at the edge device, an interference cancellation sequence is satisfied for the first k messages with the largest transmission power, which will be decoded from the mixed signal.

4.2 Algorithm Description

Our **S**table **A**ge-of-**I**nformation **S**cheduling **A**lgorithm, termed as *SAISA* for short, consists of successive stages, each of which contains T rounds. In each stage, the first H rounds belongs to the phase 1, and the remaining J rounds belongs to the phase 2. Specifically, $T = \lfloor \sqrt[3]{nk} \rfloor$, $H = \lfloor (1-\epsilon^2)T \rfloor$, $J = T - H$, with constant $\epsilon \in (0, 1)$. In phase 1, the fresh data packets are delivered to minimize the AoI, and the phase 2 is left to deliver those outdated data packets. Thus, all the uesful information can be collected by the edge-based server. Different from the traditional AoI optimization works that drop the staled packets [33], we schedule the freshest packets to minimize the AoI, but also transmit those staled packets, because the staled packet also contains useful information and should not be discarded in many practical scenarios.

In each phase, three queues S_v, Q_v, and F_v are used by each node v to store injected messages. Specifically, S_v is used by v to store the messages injected in current stage. The messages in S_v do not participate in the scheduling in current stage and will be transferred to queue Q_v at the beginning of the next stage; the queue Q_v stores the messages injected in last stage. In phase 1 of each stage, v will select some messages from Q_v, and try to deliver them to the edge device. A message will be removed from Q_v when it has been successfully received by the edge device. When phase 1 ends, the remaining messages in Q_v will be transferred into the queue F_v. In phase 2 of each stage, v will try to disseminate the messages in F_v to the edge device. Note that the queues used in our algorithm follow the first-in-first-out rule, which means that each time when v decides to select a message from a queue to transmit, the message earliest arriving at the queue will be selected. When a message from v was successfully received by the edge device, the edge device will return an acknowledgement message. Then, v will remove the message from its queues.

The pseudocode of the sensor nodes and edge device in each stage is given in Algorithm 1 and 2, respectively.

Algorithm 1: SAISA for sensor nodes

In each stage, a sensor node v does:

$Q_v \leftarrow S_v$; $S_v \leftarrow \varnothing$; // Phase 1 starts
for H *rounds* **do**
  if *in odd round* **then**
    if *Queue Q_v is not empty* **then**
      $P_v \leftarrow$Trans-Power-Selection ( );
      Select the first message $\mathcal{M}_v$ from Q_v;
      Transmit $\{ID_v, \mathcal{M}_v\}$ with power P_v;
  if *in even round* **then**
    listen;
    if *Received an acknowledgement message from the edge device to v* **then**
      $Q_v \leftarrow Q_v \setminus \{\mathcal{M}_v\}$;
$F_v \leftarrow Q_v \cup F_v$; $Q_v \leftarrow \varnothing$; // Phase 2 starts
for J *rounds* **do**
  if *in odd round* **then**
    if *Queue F_v is not empty* **then**
      $P_v \leftarrow$Trans-Power-Selection ( );
      Select the first message $\mathcal{M}_v$ from F_v;
      Transmit $\{ID_v, \mathcal{M}_v\}$ with power P_v;
  if *in even round* **then**
    listen;
    if *Received an acknowledgement message from the edge device to v* **then**
      $F_v \leftarrow F_v \setminus \{\mathcal{M}_v\}$;

Trans-Power-Selection ( ):
$z = 0$; $p = 1/2$;
do
  $a \leftarrow 0$ or 1 with probabilities p and $1-p$;
  $z \leftarrow z + 1$;
while $a = 0$;
Randomly and uniformly select an integer x from the interval $[2^{\mu z} z^4, 2^{\mu z+1} z^4]$;
Output $P_v \leftarrow P_{min} x^{\gamma x}$;

Algorithm 2: SAISA for the edge device

In each stage, the edge device u does:

for T *rounds* **do**
  if *in odd round* **then**
    Listen, and try to decode the messages $\{ID_v, \mathcal{M}_v\}$s one by one from the received signal via NOMA technique;
  if *in even round* **then**
    Broadcast all the ID_vs decoded from the signal in last rounds as the acknowledgement message with transmission power P_{max};

A detailed description for phases 1 and 2 in our algorithm execution on node v and edge device u in stage i is given in the following.

In stage 1, as the beginning stage, the queue S_v is empty when stage 1 starts, and Q_v, F_v are always empty in stage 1. Thus, v only stores the injected messages in S_v when they arrive in stage 1. No operation is given for queue Q_v and F_v.

In other stages i with $i > 1$. Note that in stage $i-1$, there have been some messages injected and stored in S_v. At the beginning of stage i, all messages in S_v are transferred to Q_v, i.e., $Q_v \leftarrow S_v$ and $S_v \leftarrow \varnothing$. Then, in each odd round of phase 1, v will select a message $\mathcal{M}_v$ from the queue Q_v according to the first-in-first-out rule, and transmit $\mathcal{M}_v$ with transmission power P_v. P_v is determined by the function Trans-Power-Selection, which will be described later in this Section. Meanwhile, in each odd round of phase 1, the edge device keeps listening to receive the messages from the nodes. After receiving some messages, the edge device will broadcast an acknowledgement message in the following even round. If a message $\mathcal{M}$ is received by the edge device in round t, and the edge device broadcasts a corresponding acknowledgement message in round $t+1$, we say $\mathcal{M}$ is acknowledged. For any node v having its message $\mathcal{M}_v$ acknowledged by the edge device in phase 1, v removes the message $\mathcal{M}_v$ from its queue Q_v.

When the first phase in stage i ends, the messages remaining in Q_v is transferred to F_v, i.e., $F_v \leftarrow Q_v \cup F_v$ and $Q_v \leftarrow \varnothing$. Then, in each odd round of phase 2, v will select a message $\mathcal{M}_v$ from the queue F_v, and broadcast $\mathcal{M}_v$ with transmission power P_v. Similarly, if v receives the acknowledgement message from the edge device, which confirms the receiving of message $\mathcal{M}_v$, it removes the message $\mathcal{M}_v$ from queue F_v. All of the new messages injected in node v during the stage i will be stored in the queue S_v.

Considering that stage 1 is simple, in the following, we mainly introduce the algorithm execution in the stage i with $i > 1$, termed as the stage $i > 1$ for short. The main difference between the phases 1 and 2 in stage $i > 1$ relies on two points: one is that the nodes select messages from Q_v in the phase 1 and from F_v in the phase 2. The messages in Q_v are all injected in stage $i-1$, and will be transferred to F_v at the end of the stage i if they are not received by the edge device in the phase 1 of current stage. While the messages in F_v not received by the edge device in the phase 2 of stage i will be scheduled by node v again in phase 2 of the next stage, until it is acknowledged by the edge device. Another one is that phase 1 and 2 have the different running time. Phase 1 contains $\lfloor (1-\epsilon^2)\sqrt[3]{nk} \rfloor$ rounds, and phase 2 contains $\lfloor \sqrt[3]{nk} \rfloor - \lfloor (1-\epsilon^2)\sqrt[3]{nk} \rfloor$ rounds, with constant $\epsilon \in (0, 1)$.

In phase 1 or 2, each time when a node v selects a message from the queues Q_v or F_v, v chooses a transmission power P_v by executing the function Trans-Power-Selection, which is detailed in our Algorithm 1. In this function, each node v firstly simulates the following coin tossing game to get a parameter z. Then, an integer x is randomly and uniformly selected from the interval $[2^{\mu z} z^4, 2^{\mu z+1} z^4]$. Finally, node v chooses $P_{min} x^{\gamma x}$ as the value of its transmission power P_v.[2] Constant μ is set to be 6 in our analysis for the final high probability guarantee even in the worst case, and may be smaller in reality. Constant γ is no less than $\max\{1, \frac{c\alpha+1+\log\beta}{\mu}\}$ with c as a constant larger than $\frac{logn}{logR}$ in analysis. Such a power selection scheme was firstly proposed in [35] for a single transmission without NOMA. Here, we extend this scheme with NOMA for multiple transmissions, which is more complex.

Coin Tossing Game: node v repeatedly flips a fair coin until the head of the coin is flipped. z is the number of times that the tail of the coin was flipped during this game. Line 24-26 of algorithm 1 is the corresponding pseudocode.

A Theorem is given here to conclude the performance of our algorithm about the stability on message scheduling, and the efficiency on EAP-AoI minimization. The corresponding proofs will be given in the next section.

Theorem 1: Our Age-of-Information scheduling algorithm is stable w.r.t. any message injection rate smaller than $\frac{k}{2n}$. The expected average peak AoI of the whole end-to-edge information system is $O(\sqrt[3]{nk})$ rounds.

Discussion for the FIFO Approach. Without the FIFO rule, it is hard to guarantee that an accumulated message will be eventually received by the server. The Last-In-First-Out (LIFO) rule can serve as a counterexample. When Phase 1 ends, the remaining messages in Q_v are transferred to F_v. In the following Phase 2, the messages in F_v are scheduled using the LIFO rule. If the number of messages transferred from Q_v to F_v in each stage exceeds or equal to the number of messages that can be delivered from queue F_v in Phase 2 (which is very likely in an interval with a high injection rate), the messages that are recently transferred into Q_v will be delivered, and the message with the oldest age will never be received by the server in the interval.

Additionally, even though the FIFO principle is deployed in each queue, the trace of messages from the sensors to the edge server is not FIFO, as some messages are transferred from one queue to another. For example, suppose node v generates its first message $\mathcal{M}_v^0$ at stage i and its second message $\mathcal{M}_v^1$ at stage $i+1$. In stage $i+1$, message $\mathcal{M}_v^0$ is in queue Q_v and has some probability to be scheduled. Message $\mathcal{M}_v^1$ is in queue S_v. If the scheduling of $\mathcal{M}_v^0$ in Phase 1 of stage $i+1$ fails, message $\mathcal{M}_v^0$ is transferred to queue F_v, and message $\mathcal{M}_v^1$ is transferred to queue Q_v. Then, in stage $i+2$, $\mathcal{M}_v^1$ is scheduled in Phase 1 as a fresh message, and $\mathcal{M}_v^0$ is scheduled in Phase 2 as an accumulated message. If both messages are successfully delivered in stage $i+2$, the server first receives $\mathcal{M}_v^1$ in Phase 1 and then $\mathcal{M}_v^0$ in Phase 2. With this example, we can see that even with FIFO rules, the order of messages cannot be guaranteed.

2. We assume that the gap between maximum transmission power and minimum transmission power is sufficient large for this setting, i.e. $P_{min} x^{\gamma x} \leq P_{max}$. Otherwise, a trade-off scheme from [35] can be used to reduce the energy requirement but with some communication efficiency sacrificed.

5 Algorithm Analysis

Our analysis section consists of three parts. In part one, we show the efficiency of our algorithm on the MEAPA problem, by proving that the EAP-AoI of the information system is $O(\sqrt[3]{nk})$ rounds. In the second part, we show that our scheduling algorithm is stable when the information online arrives at the end side. For each important message, the time gap between its generation by the end node and its reception by the edge node is $O(\sqrt[3]{nk})$ rounds in expectation. In the last part, some basic and technical proofs are given, to supports our claims in the first and second parts. Due to page limitation and also for a better organization, we put the last part into the Appendix.

The additional assumptions in our analysis are given in the following. Since it takes 2 rounds for the edge device to receive and acknowledge k messages, the injection rate $\lambda > k/2n$ for each node in each round is an inherently impossible case for our stable scheduling algorithm, in which the number of messages not received by the edge device and the AoI always increase. In [14], a similar upper bound has been proved in the case without NOMA. In the following, we will show that our algorithm works well on minimizing the EAP-AoI in all the cases of $\lambda \in (0, \frac{(1-\epsilon)k}{2n}]$, in which ϵ is an arbitrarily constant in the interval $(0,1)$. Also, this paper considers the edge scenarios with massive end devices, and has the $R \leq n^c$ assumption for the following analysis, in which c is a positive constant.

5.1 Efficiency of Our Algorithm on MEAPA

The peak AoI w.r.t. a node v in the whole interval I is rather complex to analyze, because v may become inactive after all its freshest messages have been received by the edge node. To make the analysis brief, we can divide the whole interval I into multiple sub-intervals I', in each of which the node v always keeps active.[3] Each of the sub-interval is also termed as the active interval for node v. Without loss of generality, we consider the AoI w.r.t. node v in an active interval I' starting from round t_0 and ending at round t'. During the sub-interval I', the AoI of node v reaches to its peak for h times at rounds $t_1, t_2, ..., t_h$, respectively, with $t_0 \leq t_1 < t_2 < ... < t_h \leq t'$. For $j = 1, 2, ..., h$, let $\mathcal{M}_j$ be the fresh message received by the edge node at the round t_j. In other words, the AoI of v reaches to a peak at the round t_j, and drops because the fresh message $\mathcal{M}_j$ is received by the edge node at the end of round t_j. Let a_j and b_j be the generation time of message $\mathcal{M}_j$ by node v and the reception time by the edge node, respectively. Obviously, $b_j = t_j$ and a_j is the

3. In other words, v always has some fresh messages not received by the edge node during those sub-intervals.

moment when an important information is injected on the node v.

In the following, we consider the peak AoI achieved at rounds t_j with cases: $j = 1$, $j = 2, 3, ..., h-1$, and $j = h$. In the first case with $j = 1$, the analysis on peak AoI $\mathcal{A}_v(t_1)$ branches on whether $\mathcal{M}_1$ is the first message generated by the node v in the active interval I'. In the second case with $j = 2, 3, ..., h-1$, the generation time and reception time of message $\mathcal{M}_{j-1}$ can help to bound the peak age $\mathcal{A}_v(t_j)$. In the final case with $j = h$, the analyses slightly differ on whether sub-interval I' is the last active interval in the whole interval I. In general, we use Lemmas 1, 2, and 3 to show that $\mathbb{E}(\mathcal{A}_v(t_j)) = O(\sqrt[3]{nk})$ on all of the cases mentioned above. In other words, all the peak AoI w.r.t node v in a sub-interval I' have their expected values bounded by $O(\sqrt[3]{nk})$. Extending this result to all the active sub-intervals in the whole interval I and all the end nodes, the EAP-AoI of our stable information system will also be $O(\sqrt[3]{nk})$ rounds, which proves the EAP-AoI part in Theorem 1.

The Lemmas 1, 2, and 3 are presented in the following.

Lemma 1: For any given sub-interval I' in which v is active, $\mathbb{E}(\mathcal{A}_v(t_1)) = O(\sqrt[3]{nk})$.

Proof: As is assumed above, the AoI w.r.t. node v reaches to the first peak at the round t_1, and drops when the message $\mathcal{M}_1$ from v is received by the edge node. a_1 and b_1 are the generation time of $\mathcal{M}_1$ by node v and the reception time by the edge node, respectively. Then, the analysis of $\mathbb{E}(\mathcal{A}_v(t_1))$ branches in the following two cases. In case 1, when $\mathcal{M}_1$ is the first message generated by the node v in the interval I', we have $a_1 = t_0$, $b_1 = t_1$, and $\mathcal{A}_v(t_1) = t_1 - t_0 = b_1 - a_1$. In case 2, when $\mathcal{M}_1$ is not the first message generated by the node v in I', let's assume that $\mathcal{M}'$ is the first generated message with a' and b' be its generation time by v and reception time by the edge node. The message $\mathcal{M}'$ was generated earlier than $\mathcal{M}_1$, but received by the edge node later, due to some communication failures. Then, we have $a' = t_0$, $b_1 = t_1$, and $a' < a_1 < b_1 < b'$, the last of which holds because $\mathcal{M}'$ is generated earlier by node v but received later by the edge node compared than $\mathcal{M}_1$. Thus, $\mathcal{A}_v(t_1) = t_1 - t_0 = b_1 - t_0 \leq b' - a'$. In Lemma 4, we give an expected bound for the time gap of a message between its generation time to its reception time. According to the time bound in Lemma 4, we have $\mathbb{E}(\mathcal{A}_v(t_1)) = O(\sqrt[3]{nk})$. □

Lemma 2: For any given sub-interval I' in which v is active, $\mathbb{E}(\mathcal{A}_v(t_j)) = O(\sqrt[3]{nk})$ with $j = 2, 3, ..., h-1$.

Proof: For two messages $\mathcal{M}_{j-1}$ and $\mathcal{M}_j$ with $j = 2, 3, ..., h-1$, we have $a_{j-1} < b_{j-1}$, $a_j < b_j$, $a_j \leq b_{j-1}+1$. Otherwise, the node v will be inactive from round $b_{j-1}+1$ to round a_j, which contradicts the definition of active sub-interval I'. Then, we have $\mathcal{A}_v(t_j) = t_j - a_{j-1} = t_j - b_{j-1} + b_{j-1} - a_{j-1} \leq b_j - (a_j - 1) + b_{j-1} - a_{j-1}$. According to the time bound $\mathbb{E}(b-a) = O(\sqrt[3]{nk})$ for arbitrary message $\mathcal{M}$ from lemma 4, we can get $\mathbb{E}\left(\mathcal{A}_v(t_j)\right) = O(\sqrt[3]{nk})$. □

Lemma 3: For any given sub-interval I' in which v is active, $\mathbb{E}(\mathcal{A}_v(t_h)) = O(\sqrt[3]{nk})$.

Proof: As is mentioned above, $\mathcal{M}_v(t_h)$ is the last peak AoI obtained in the sub-interval I'. Its analyses differs in the following two cases. Case 1: the last peak AoI is obtained at round t_h when $\mathcal{M}_h$ is received by the edge node at the end of round t_h and I' ends because v becomes inactive from the round $t_h + 1$. In this case, we can conduct an analysis on the messages $\mathcal{M}_{h-1}$ and $\mathcal{M}_h$ to obtain the result that $\mathbb{E}(\mathcal{A}_v(t_h)) = O(\sqrt[3]{nk})$, which is similar with that in Lemma 2. Case 2 is a very special case, in which the peak age is obtained at round t_h not due to a fresh message received by the edge node but because t_h is the last round of interval I. In this case, $\mathbb{E}(\mathcal{A}_v(t_h)) = \mathbb{E}(t_h - a_{h-1}) \leq \mathbb{E}(b_h - a_{h-1}) = O(\sqrt[3]{nk})$. □

5.2 Stability of Our Algorithm on MEAPA

In this part, we show the stability of our algorithm when important information online arrives at the end side. From the last section, our algorithm execution in stage 1 is simple. In the following, we choose an arbitrary stage $i > 1$ to start our analysis, and consider a *busy* scenario in which $\lambda = \frac{(1-\epsilon)k}{2n}$, and the queues Q_v, F_v on nodes v always contain enough messages to schedule. Once we proved the Theorem 1 in this harsh case, the performance of our algorithm can also be guaranteed in some other weaker cases with a similar proof.

Let m_i be the number of messages injected in the end nodes at stage $i-1$. Then, at the beginning of phase 1 in stage i, m_i messages will be transferred to queues Q_vs and scheduled by the nodes. Once a message was received and acknowledged by the edge device, it is removed from the queues Q_vs. When phase 1 ends, the messages that still remain in queue Q_v will be transferred to queue F_v by the node v. In each round of phase 2, each node v selects a message from its queue F_v to broadcast with the transmission power P_v.

From the view of a message $\mathcal{M}_v$ that is injected in node v in stage $i-1$, it firstly stays in queue S_v until the stage i starts. Then, it is transferred to queue Q_v by node v. In phase 1 of stage i, $\mathcal{M}_v$ has some probability to be transmitted by v, and removed from the Q_v if the edge device receives $\mathcal{M}_v$. If $\mathcal{M}_v$ still remains in the queue Q_v when phase 1 ends, it is transferred to the queue F_v and scheduled in phase 2 of the following stages until it is acknowledged by the edge device. The following Properties for the phase 1 and the phase 2 are given first as a sufficient condition for proving the stability of our algorithm in Theorem 1, the technical proof of which are put into the Appendix for better organization.

Property 1: In phase 1 of an arbitrary stage $i > 1$:

- there are totally $\frac{kH}{2}$ messages in Q_vs acknowledged by the edge device at least with the probability of $1 - \frac{1}{n^5}$;
- $Pr[E[\sum_{v \in V} |Q_v|] \geq rkH/2 + 1] \leq H^{-4-r}$ for $r \in \{0, 1, \cdots\}$, when the phase 1 of stage i ends. In other words, the probability that the expected number of messages remaining in queue Q_vs is at least $\frac{rkH}{2} +$

1 is at most H^{-4-r} for any non-negative integer r when the phase 1 of stage i ends;

Property 2: In phase 2 of an arbitrary stage $i > 1$:

- in each pair of odd and even rounds, each node v has its messages in F_v acknowledged at least with probability $\frac{k}{n}(1-\frac{1}{n^5})$;
- $\forall v \in V$, $E[|F_v|] \le k^2H^2/4$. In other words, when the phase 2 of stage i ends, for each node v, the number of messages in queue F_v is no larger than $k^2H^2/4$ in expectation.

Lemma 4: For any message $\mathcal{M}$ generated by node v at round a, and received by the edge node at round b, we have $\mathbb{E}(b-a) = O(\sqrt[3]{nk})$.

Proof: Without loss of generality, we assume that the message $\mathcal{M}$ is injected to node v in stage $i-1$. In general, there are two cases for $\mathcal{M}$ acknowledged by the edge device. Case 1: $\mathcal{M}$ is acknowledged in phase 1 of stage i. Case 2: $\mathcal{M}$ is acknowledged in phase 2 of stage j, $j \ge i$. Let Z be 1 if case 1 occurs, and be 2 if case 2 occurs. Obviously, $E[b-a|Z=1] \le 2T$ and $Pr[Z=1] \le 1$. By setting $r=0$ in result 2 of Property 1, $Pr[Z=2] \le H^{-4}$. Combining the first-in-first-out rule and Property 2, $E[b-a|Z=2] \le \frac{k^2H^2/4}{\frac{k}{n}(1-\frac{1}{n^5})} \cdot \frac{2T}{J}$. Thus, we get

$$\begin{aligned} E[b-a] &= Pr[Z=1]E[b-a|Z=1] + \\ &\qquad Pr[Z=2]E[b-a|Z=2] \\ &\le 2T + H^{-4} \cdot \frac{k^2H^2/4}{\frac{k}{n}(1-\frac{1}{n^5})} \cdot \frac{2T}{J} \\ &\le 2T + \frac{nkT}{2JH^2} = O(\sqrt[3]{nk}). \end{aligned} \tag{6}$$

□

Our Algorithm is Stable. Note that for each node v in an arbitrary stage $i > 1$, the injected messages at most stay in the queue S_v for one stage before they are transferred into the queue Q_v. Messages in Q_v are either acknowledged by the edge device, or transferred to queues F_v at the end of phase 1. So, the stability of F_v directly reflects that whether our algorithm is stable under the injection rate λ. From the Properties 1 and 2, we can see that there are $kT/2$ messages scheduled in each T rounds with high probability,[4] if there are enough messages injected in the information system. Also, at the end of each stage, the expectation number of messages in queue F_v of each node v is bounded as $k^2H^2/4$. Besides, as proved in the last subsection, the EAP-AoI of our stable information system is $O(\sqrt[3]{nk})$; Thus, the stability and efficiency of our algorithm are strongly supported.

6 Simulation Results

We illustrate the empirical performances of our SAISA with NOMA in this Section. Specifically, the efficiency and stabilization of SAISA are investigated by observing the average peak age, number of messages on the end side, and comparing with the previous algorithms, when NOMA parameter k, number of end devices n, and injection rate λ vary. The average peak age observed in simulations directly shows the efficiency of SAISA and reflects the stabilization. Also, when a message is injected but not arrived at the edge side, we say it is still on the end side. Thus, the number of messages on the end devices also verifies the stabilization of our algorithm. Finally, a comparison between our algorithm and the existing works are given to show the advantage of our work on online scheduling and the NOAM technology.

6.1 Settings in Simulation

Initially, there are n end devices randomly and uniformly deployed in a circle area with radius of 300 m. The edge device is in the centre of the circle area. The transmission range of the edge/end devices are also 300 m.[5] By setting P_{max} sufficient large and $k \in \{5, 10\}$, the edge device in our simulation can at most decode k messages from a mixed signal in each round by applying the NOMA technique. We use a *random injection mode* to depict the arrivals of the useful information on the end side, in which λ_1 is defined as the sum of the injection rate of all the end devices, also termed as total injection rate for short. Note that with any given k in NOMA, the edge device can at most decode k messages in each round. An inherent bound for λ_1 is k, the details of which will be discussed later. In simulation, we test the performance of our algorithm by choosing $n \in \{2.5, 5.0, 7.5, 10.0\} \times 10^3$, $k \in \{5, 10\}$, and $\lambda_1 = \{0.1, 0.2, 0.3, 0.4, 0.5\} \times k$. A detailed parameter setting is given in Table 3. Note that the random and uniform deployment of nodes in our simulation is a common setting that can be found in the works [4], [11], [27], [34], and our random injection mode is inspired by the statistical injection mode in [14], in which the variety of nodes on message injection is ignored. For each reported result, our algorithm has been executed by at least 1.0×10^4 rounds to show the stability of our algorithm, and are verified by over 20 runs. All experiments are conducted on a Linux machine with Intel Xeon CPU E5-2670@2.60GHz and 128 GB main memory, implemented in C++ programming language.

Random Injection Mode. In our information system with an edge device and n end devices, λ_1 has been defined as the sum of the injection rate of all the end devices in one round, i.e., larger the value of λ_1, more useful information will arrive at the end side in each round in expectation, and harder will it be for our algorithm to keep efficient and stable in terms of AoI and number of messages on the end side, respectively. An inherent bound for λ_1 is k for any algorithm with NOMA because the edge side can at most receive k messages in each round. And for our algorithm, the upper bound is $0.5k$ since the edge device requests an extra round to give an acknowledgement after receiving

4. With a probability $1 - n^{-c_0}$ for some positive constant $c_0 > 1$.

5. In reality, a large fraction of end/edge devices in 5G have their transmission ranges in this magnitude.

TABLE 3: Parameter settings in simulation

Parameter settings	Definitions	Parameter settings	Definitions
$n \in \{2.5, 5.0, 7.5, 10.0\} \times 10^3$	Number of the end devices	$R = 300m$	Transmission range
$k \in \{5, 10\}$	Parameter in NOMA technology	$\epsilon = 0.2$	Parameters in SAISA
$\lambda_1 \in \{0.2, 0.4, 0.6, 0.8, 1.0\} \times \lambda_{max}$	The sum of the injection rates	$T = \lfloor \sqrt[3]{nk} \rfloor$	Parameters in SAISA
$\lambda_{max} = k/2$	An upper bound of λ_1	$H = \lfloor (1-\epsilon^2) \times T \rfloor$	Parameters in SAISA
$\alpha = 3.0$, $\beta = 1.5$, $N = 1.0$	SINR parameters	$J = T - H$	Parameters in SAISA

TABLE 4: Stable Value of the Average Peak AoI.

Stable Values		$n = 2500$	$n = 5000$	$n = 7500$	$n = 10000$
$k = 5$	$\lambda_1 = 0.5$	122.87	172.74	210.65	243.47
	$\lambda_1 = 1.0$	157.75	222.33	272.36	311.25
	$\lambda_1 = 1.5$	192.11	270.69	332.26	381.20
	$\lambda_1 = 2.0$	227.40	321.51	392.65	452.19
	$\lambda_1 = 2.5$	No longer stable			
$k = 10$	$\lambda_1 = 1.0$	143.24	201.15	245.49	283.11
	$\lambda_1 = 2.0$	202.54	285.56	349.65	399.84
	$\lambda_1 = 3.0$	247.84	348.26	427.59	493.89
	$\lambda_1 = 4.0$	291.78	413.45	503.33	577.16
	$\lambda_1 = 5.0$	No longer stable			

the messages from the end devices. Thus, in simulation, we test the performance of our algorithm with $\lambda_1 \in \{0.1, 0.2, 0.3, 0.4, 0.5\} \times k$. After the total injection rate λ_1 defined and fixed, an intuitive way to simulate the arrival of the useful information on each node is to uniformly assume that each end device has one information arrived with the probability of λ_1/n in each round. However, this uniform assumption neglects the fact that the injection rates of the end devices may differ because of their varieties in reality. To make our injection mode more realistic, we assume that in each round, each end device has its injection rate randomly chosen from the interval $[\frac{\lambda_1}{2n}, \frac{3\lambda_1}{2n}]$. With such an uniform and random assumption in our injection mode, we make sure that the total injection rate λ_1 can reflect the total arrivals of the useful information on end side and the varieties of the end devices in reality.

Concrete Example in Practice. Our distributed information system is applicable to various edge-based and Internet-of-Things (IoT) scenarios. For instance, consider an edge-based smart agriculture system. Multiple sensor nodes, each equipped with cameras and basic computing and communication units, are deployed in farmlands. When the pictures of pests are caught by the camera and detected as significant by the local filter, the pest information is sent to the edge server, and an unmanned aerial vehicle is dispatched to address the issue. The local filter in this context can be a pre-trained, lightweight AI model capable of running on resource-constrained sensor nodes. In each farmland, the appearance of pests may follow a certain distribution. In our simulations, we assume a uniform distribution for simplicity, where important information is injected into an end device with the Random Injection Mode. The uniform distribution adopted in our simulation is one of the most common distributions in the field of statistic.

6.2 Evaluation of Simulation

Average Peak Age-of-Information. Firstly, we investigate the average peak AoI of the whole information system in Figure 3 when our algorithm is executed, in which the x-axes and y-axes represent the number of rounds and the average peak AoI observed in our simulation, respectively. Specifically, in Figure 3, we show the average peak AoI of all the end devices by setting $k \in \{5, 10\}$, $n \in \{2.5, 5.0, 7.5, 10.0\} \times 10^3$, and $\lambda_1 \in \{0.1, 0.2, 0.3, 0.4, 0.5\} \times k$. For a clear observation, the stable values of their average peak AoI are also liseted in the Table 4. From the curves in Figure 3 and the data in Table 4, we can get the results that our algorithm keeps stable and is efficient when $\lambda_1 \leq 0.4k$, and the maximum total injection rate which can be handled by our algorithm is within $[0.4k, 0.5k]$. The detailed analyses for the curves in Figure 3 are given in the following:

- for each of the curves in Figure 3 with $\lambda_1 \leq 0.4k$, the average peak AoI firstly increases and then keeps stable at no more than 700 rounds. This is because our SAISA in the simulation has a short initialization period in which the message injection just starts. Later, when messages are gradually injected, the average peak AoI keeps stable at a relatively low level, with respect to the massive end devices;
- by comparing all the curves with the same k, n and various $\lambda_1 \leq 0.4k$ in the Figure 3, we can see that the average peak AoI slightly increases when λ_1 gets larger. In other words, when there are more messages injected in each round, it takes longer time for SAISA to schedule those messages;
- by comparing all the curves with the same k, $\lambda_1 \leq 0.4$ and various n in the Figure 3, we can see that the average peak AoI keeps stable at a larger value when n gets larger. This result verifies our theoretical time complexity $O(\sqrt[3]{nk})$ in Lemma 4. Note that even in the worst case with $n = 10000$, $\lambda_1 = 4.0$ and $k = 10$, the average peak AoI keeps stable at no more than 700 rounds, which is a competitive result in terms of the total number of nodes;
- from the curves in Figure 3 with $\lambda_1 = 0.5k$, we observe that the number of messages injected in each round in expectation is larger than the number of messages that can be scheduled by our algorithm. Thus, more and more messaged are injected but stay on the end side, wait for scheduling. And, the average peak AoI keeps increasing when $\lambda_1 = 0.5k$. From the observations that our algorithm is stable and efficient when $\lambda_1 \leq 0.4k$, but the average AoI keeps increasing when $\lambda_1 = 0.5k$, we get a result that the maximum total injection rate that can be handled by our algorithm is within $[0.4k, 0.5k]$.
- by comparing the curves with the same n, $\lambda_1 \leq 0.4$

and various k in the Figure 3 and the data in Table 4, we can see that when k doubles from 5 to 10, the maximum total injection rate that can be handled by our algorithm increases approximately by 2 times. With the same n and λ_1, the average peak AoI with $k = 10$ is smaller than that with $k = 5$. In other words, more simultaneous transmissions is helpful for our algorithm to tolerate a higher injection rate and reduce the average peak AoI.

Average and Maximum Number of Messages on the End Side. Figure 4 depicts the average and maximum number of messages on each of the end devices, which have been injected but still not arrived at the edge side. With a detailed analysis and comparison on the curves in Figures 4, we can get the following results.

- when $\lambda_1 \leq 0.4k$, the number of messages on the end devices keeps stable at a small value. Specifically, when $n \in \{2.5, 5.0, 7.5, 10.0\} \times 10^3$, $\lambda_1 \in \{0.1, 0.2, 0.3, 0.4\} \times k$ and $k = 5$ in Figures 4, the average/maximum number of messages on the end side keeps stable at a larger value when $1/n$ and λ_1 gets larger. In the worst case when $n = 2500$, $\lambda_1 = 2.0$, $k = 5$, we have the average and maximum number of messages no larger than 0.1 and 6, respectively.
- when $\lambda_1 = 0.5k$, the average and maximum number of messages on the end side always increase, and our algorithm no longer keeps stable.

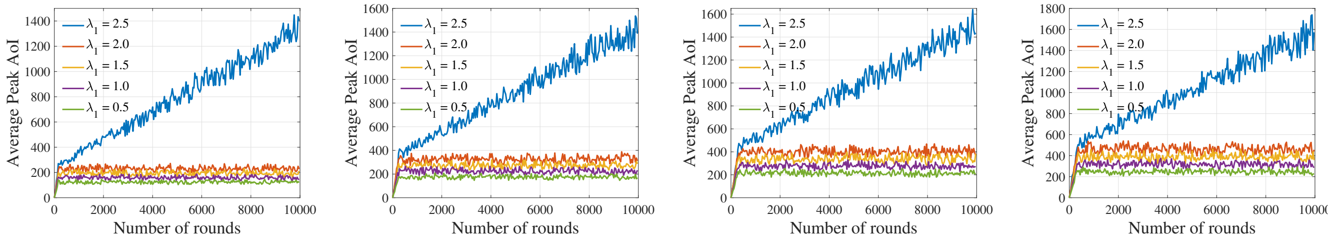
Comparison with the Previous Works. As is mentioned in the motivation of our research, scheduling the useful information in an online mode is more efficient than periodically scheduling all of the information in an offline mode, regardless of the importance of the information. And combining with NOMA technology dramatically improves the performance of the scheduling algorithm on efficiency and stabilization. To verify these standpoints, in this part, we compare our algorithm with the following ones in terms of efficiency and stabilization on information scheduling:

- OPT_{TDMA}: an optimal and centralized TDMA scheme with NOMA in an offline mode, in which the messages from n end devices are periodically scheduled to the edge device. Specifically, in each round, the messages from the k end nodes with the largest ages are scheduled to the edge side, which is a centralized greedy scheme. Thus, in each of the first i round, there are always k nodes having their peak AoI obtained at the value of i when $i \leq \lfloor n/k \rfloor$. In the remaining rounds of the interval I, there are always k nodes having their peak AoI obtained at the value of $\lfloor n/k \rfloor + 1$. Thus, we get the average peak AoI of this TDMA scheme
$$\frac{\sum_{i=1}^{\lfloor n/k \rfloor} ki + \sum_{i=\lfloor n/k \rfloor+1}^{|I|} k(\lfloor n/k \rfloor + 1)}{k|I|}, \qquad (7)$$
which gets close to n/k when I is sufficient large.
- SLB: a stable local broadcast algorithm without NOMA in [14], in which the messages are randomly injected on each of the end node, but at most one message can be decoded by the edge node from a mixed signal in each round. When an end node has its fresh message received by the edge node, its AoI reaches to a peak in current round and drops in the next round.
- $DAoI$: a distributed AoI minimization algorithm without NOMA in [15], in which the sensor nodes with data packets join in or leave the networks. Without NOMA technique, the edge node can at most receive one message in each round.

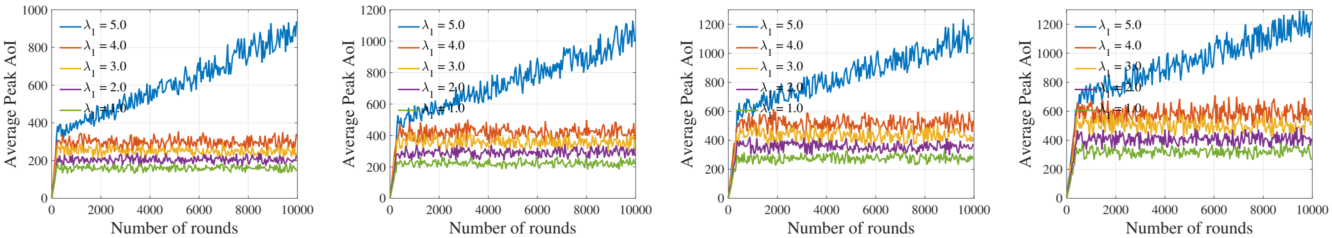
In Figure 5 (a)-(d), we compare the performance of our algorithm with OPT_{TDMA}, with $n \in \{2.5, 5.0, 7.5, 10.0\} \times 10^3$, $\lambda_1 \in \{0.5, 1.0, 1.5, 2.0\}$ and $k = 5$. Note that all of the end devices participate in the information scheduling in OPT_{TDMA}. Thus, its average peak AoI has no relationship with the online injection of the useful information, but directly determined by n and k, as is formulated in Equation 7. As for our SAISA in Figure 5 (a)-(d), we can see that its average AoI always keeps stable when algorithm is executed and slightly gets larger when λ_1 or n increases. By comparing the performance of our algorithm with that of the OPT_{TDMA} in Figure 5 (a)-(d), we can get the results that the average peak AoI of our algorithm is at least 2 times smaller than that of the OPT_{TDMA} and the gap in the comparison becomes larger when the number of nodes increases. In other words, our algorithm is more efficient, especially in a large scale network with massive end devices.

In Figure 5 (e)-(h), we simulate the stable local broadcast (SLB) algorithm without NOMA in [14], the distributed AoI (DAoI) optimization algorithm without NOMA in [15] by setting $n \in \{2.5, 5.0, 7.5, 10.0\} \times 10^3$, $\lambda_1 \in \{0.1, 0.2\}$, and our SAISA with $k = 5$. Specifically, in Figure 5 (e)-(h), the average peak AoI of our SAISA with $\lambda_1 = 0.1$ and 0.2 keeps stable at the low levels. Whereas, the average peak AoI of SLB algorithm is about 4 times larger than of our SAISA when $\lambda_1 = 0.1$. Additionally, its average AoI no longer keeps stable when $\lambda_1 = 0.2$. As for the $DAoI$, its average peak AoI is about $2 \sim 3$ times larger than that of our SAISA when n increases from 2500 to 10000. From the above comparison, we get the result that combining with NOMA in our SAISA does help on reducing the average peak AoI.

Conclusion on Simulated Results. In simulation section, we test the efficiency and stabilization of our SAISA by investigating the average peak Age-of-Information, the number of messages on the end side, and the comparisons with previous works under various settings of parameters. From the above investigations, we know that 1) our SAISA is stable when the total injection rate is no larger than $0.4k$. Noting that k is an inherent upper bound for the total injection rate, the maximum injection rate which can be handled by our algorithm is asymptotically optimal; 2) when our algorithm is stable, the average peak AoI and the number of messages on the end side keeps stable at the low levels, which shows the efficiency of our algorithm; 3) compared with an

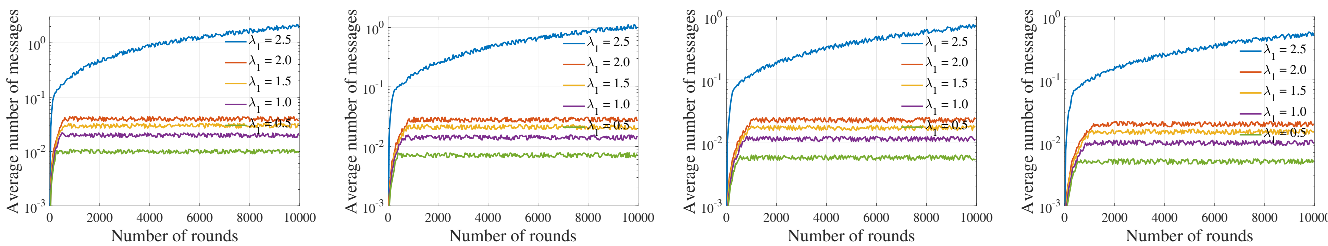


(a) $n = 2500$, $k = 5$ (b) $n = 5000$, $k = 5$ (c) $n = 7500$, $k = 5$ (d) $n = 10000$, $k = 5$

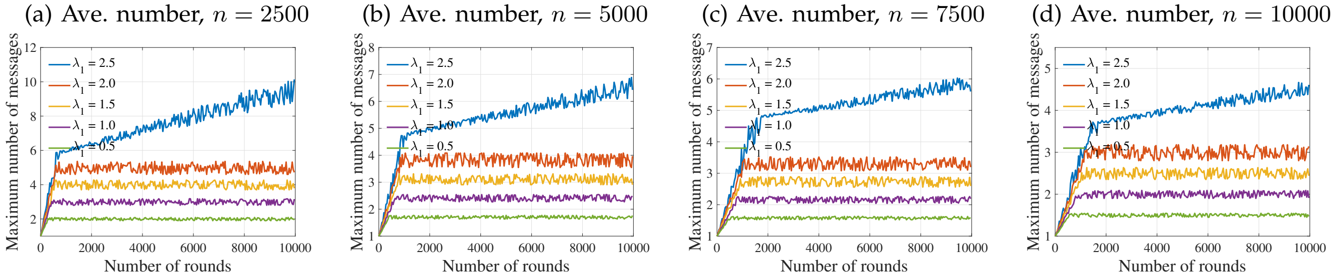


(e) $n = 2500$, $k = 10$ (f) $n = 5000$, $k = 10$ (g) $n = 7500$, $k = 10$ (h) $n = 10000$, $k = 10$

Fig. 3: Average Peak Age-of-Information when k, n and λ_1 vary.

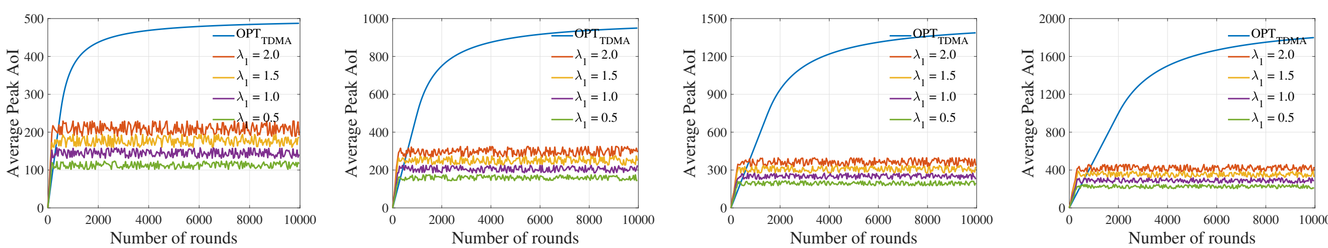


(a) Ave. number, $n = 2500$ (b) Ave. number, $n = 5000$ (c) Ave. number, $n = 7500$ (d) Ave. number, $n = 10000$

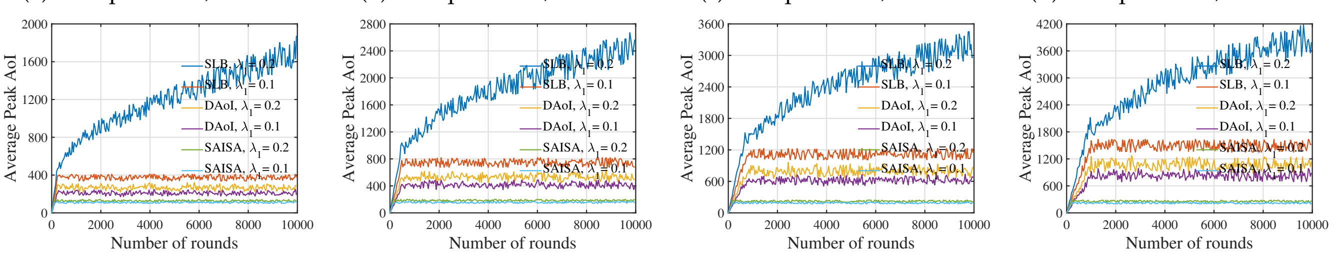


(e) Max. number, $n = 2500$ (f) Max. number, $n = 5000$ (g) Max. number, $n = 7500$ (h) Max. number, $n = 10000$

Fig. 4: Average and maximum number of messages on the end side when $k = 5$, n and λ_1 vary.



(a) Comparison I, $n = 2500$ (b) Comparison I, $n = 5000$ (c) Comparison I, $n = 7500$ (d) Comparison I, $n = 10000$



(e) Comparison II, $n = 2500$ (f) Comparison II, $n = 5000$ (g) Comparison II, $n = 7500$ (h) Comparison II, $n = 10000$

Fig. 5: Comparisons I and II on Average Peak Age-of-Information when $k = 5$, n and λ_1 vary.

optimal TDMA scheme in offline mode and a stable local broadcast algorithm in [14] without NOMA, we verify our standpoint that the online scheduling mode with NOMA does help on improving the efficiency and strengthening the stabilization on AoI scheduling in an end-to-edge information system.

7 CONCLUSION

We initiated the study on stable AoI scheduling problem with online message injections under the NOMA-SINR model. To handle the unknown injection rate and the unpredictable injections when messages are constantly injected at each end node, a distributed stable algorithm with NOMA is presented. Even under the asymptotically maximum injection rate of $O(k/n)$ that any stable AoI scheduling algorithm may handle, our proposed algorithm keeps stable with the expected average peak Age-of-Information bounded by $O(\sqrt[3]{nk})$ rounds. Comparing with two existing schemes, the expected average peak Age-of-Information in our algorithm is $O(\frac{n^{2/3}}{k^{4/3}})$ and $O(\frac{n^{2/3}}{k^{1/3}})$ times smaller. Our results shed a light on stable protocol studies for realistic AoI scheduling problems in edge networks. Extending our research to the mobile edge networks will be our work in the future.

ACKNOWLEDGEMENT

This work was in part supported by the Federal Ministry of Education and Research (BMBF, Germany) within the 6G Research and Innovation Cluster 6G-RIC under Grant 16KISK020K, the Joint Key Funds of National Natural Science Foundation of China under Grant U24A20244, and Shandong Science Fund for Excellent Young Scholars (No.2023HWYQ-007).

REFERENCES

[1] E. Altman, R. EI-Azouzi, D. Menasche, and Y. Xu. Forever Young: Aging Control For Smartphones In Hybrid Networks. Available: http://arxiv.org/abs/1009.4733.

[2] S. Kaul, R. Yates, and M. Gruteser. Real-Time Status: How Often Should One Update? in *INFOCOM*, 2012.

[3] R. D. Yates, Y. Sun, D. R. Brown III, S.K. Kaul, E. Modiano, S. Ulukus. Age of information: An introduction and survey. in arXiv preprint arXiv:2007.08564, 2020.

[4] J. Wang, X. Cao, B. Yin, Y. Cheng. Sleep-Wake Sensor Scheduling for Minimizing AoI-Penalty in Industrial Internet of Things, in *IEEE Internet Things J.*, 9(9): 6404-6417, 2022.

[5] J. Wang, D. Qiao. How to Minimize the Weighted Sum AoI in Multi-Source Status Update Systems: OMA or NOMA? CoRR abs/2205.03143, 2022.

[6] J. Zhu, J. Gong. Online Scheduling of Transmission and Processing for AoI Minimization with Edge Computing, CoRR abs/2202.06193, 2022.

[7] Y. Deshpande, O. Ayan, W. Kellerer. Improving AoI via Learning-based Distributed MAC in Wireless Networks, CoRR abs/2202.09222, 2022.

[8] P. Zou, O. Ozel, S. Subramaniam. Optimizing Information Freshness Through Computation-Transmission Tradeoff and Queue Management in Edge Computing, in *IEEE/ACM Trans. Netw.*, 29(2): 949-963, 2021.

[9] W. Ahsan, W. Yi, Z. Qin, Y. Liu, and A. Nallanathan. Resource Allocation in Uplink NOMA-IoT Networks: A Reinforcement-Learning Approach, in *IEEE Trans. Wirel. Commun*, 20(8): 5083–5098, 2021.

[10] J. Chen, Y. Liu, Q. Chen, X. Lan, G. Cheng, Y. Fu, Z. Zhang. On the Adaptive AoI-aware Buffer-aided Transmission Scheme for NOMA Networks, in *WCNC*, 1-6, 2021.

[11] D. Yu, Y. Zou, M. Xu, Y. Xu, Y. Zhang, B. Gong, X. Xing. Competitive Age of Information in Dynamic IoT Networks, in *IEEE Internet Things J.*, 8(20): 15160-15169, 2021.

[12] M. Yuan, L. Zhang, F. He, X.Tong, X. Li. InFi: End-to-End Learnable Input Filter for Resource-efficient Mobilecentric Interference, in *MobiCom*, 2022.

[13] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury and A. K. Qin. A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges (Extended Abstract), in *ICDE*, 2023,

[14] D. Yu, Y. Zou, J. Yu, X. Cheng, Q. Hua, H. Jin, F. C. M. Lau. Stable Local Broadcast in Multihop Wireless Networks Under SINR, in *IEEE/ACM Trans. Netw.*, 26(3): 1278-1291, 2018.

[15] L. Yang, Y. Zou, D. Yu, J. Yu. Distributed Age-of-Information optimization in edge computing for Internet of Vehicles, in *J. Syst. Archit.*, 144: 103000, 2023.

[16] C. Li, S. Li, Y. Chen, Y.T. Hou, W. Lou. AoI Scheduling with Maximum Thresholds, in *INFOCOM*, 2020.

[17] C. Li, Q. Liu, S. Li, Y. Chen, Y. T. Hou, W. Lou. On Scheduling with AoI Violation Tolerance, in *INFOCOM*, 2021.

[18] C. Tsai, C. Wang. Unifying AoI Minimization and Remote Estimation - Optimal Sensor/Controller Coordination with Random Two-way Delay, in *INFOCOM*, 2020.

[19] Z. Fang, J. Wang, C. Jiang, Q. Zhang, Y. Ren. AoI-Inspired Collaborative Information Collection for AUV-Assisted Internet of Underwater Things, in *IEEE Internet Things J.*, 8(19): 14559-14571, 2021.

[20] J. Lou, X. Yuan, S. Kompella, N. Tzeng. AoI and Throughput Tradeoffs in Routing-aware Multi-hop Wireless Networks, in *INFOCOM*, 2020.

[21] R. Talak, S. Karaman, E. Modiano. Optimizing Information Freshness in Wireless Networks Under General Interference Constraints, in *IEEE/ACM Trans. Netw.*, 28(1): 15-28, 2020.

[22] H. Lv, Z. Zheng, F. Wu, G. Chen. Strategy-Proof Online Mechanisms for Weighted AoI Minimization in Edge Computing, in *IEEE J. Sel. Areas Commun.*, 39(5): 1277-1292, 2021.

[23] R. Talak, S. Karaman, E. Modiano. Distributed Scheduling Algorithms for Optimizing Information Freshness in Wireless Networks, in *SPAWC*, 2018.

[24] Y. Yang, X. Wei, R. Xu, L. Peng, L. Liu. Game-Based Channel Access for AoI-Oriented Data Transmission Under Dynamic Attack, in *IEEE Internet Things J.*, 9(11): 8820-8837, 2022.

[25] S. Deshmukh and B. B. Lozano. Online Rate Allocation for AoI Minimization in an Energy Constrained D2D Communication, in *COMSNETS*, 2024.

[26] H. Hu, K. Xiong, H. Yang, Q. Ni, B. Gao, P. Fan, K. B. Letaief. AoI-Minimal Online Scheduling for Wireless-Powered IoT: A Lyapunov Optimization-Based Approach, in *IEEE Trans. Green Commun. Netw.*, 7(4): 2081-2092, 2023.

[27] B. A. G. R. Sharan, S. Deshmukh, S. R. B. Pillai, B. Beferull-Lozano. Energy Efficient AoI Minimization in Opportunistic NOMA/OMA Broadcast Wireless Networks, in *IEEE Trans. Green Commun. Netw.*, 6(2): 1009-1022, 2022.

[28] C. Guo, S. Wu, Z. Deng, J. Jiao, N. Zhang, Q. Zhang. Age-optimal Power Allocation Policies for NOMA and Hybrid NOMA/OMA Systems, in *ICC*, 2021.

[29] A. Maatouk, M. Assaad, A. Ephremides. Minimizing The Age of Information: NOMA or OMA? in *INFOCOM Workshops*, 2019.

[30] T. Zhu, J. Li, H. Gao, Y. Li, and Z. Cai. AoI Minimization Data Collection Scheduling for Battery-Free Wireless Sensor Networks, in *IEEE Transactions on Mobile Computing*, DOI: 10.1109/TMC.2021.3106013.

[31] Q. He, D. Yuan, A. Ephremides. Optimizing freshness of information: On minimum age link scheduling in wireless systems, in *WiOpt*, 2016.

[32] P. N. Thakre and S. B. Pokle, A survey on Power Allocation in PD-NOMA for 5G Wireless Communication Systems, in ICETET-SIP, 2022.

[33] C. Maice, M. Codreanu, and A. Ephremides. On the age of information in status update systems with packet management, in *IEEE Transactions on Information Theory*, 62(4): 1897-1910, 2016.

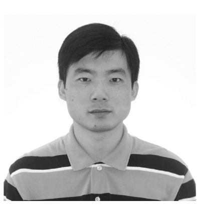
[34] D. Yu, Y. Zhang, Y. Huang, H. Jin, J. Yu, Q. Hua. Exact Implementation of Abstract MAC Layer via Carrier Sensing, in *INFOCOM*, 2018.

[35] M. M. Halldórsson, S. Holzer, E. A. Markatou, N. A. Lynch. Leader election in SINR model with arbitrary power control, in *Theor. Comput. Sci.*, 811: 21-28, 2020.
[36] R. Motwani and P. Raghavan. Randomized algorithms. In Cambridge University Press, 1995.
[37] T. Kesselheim. Dynamic message scheduling in wireless networks, in *PODC*, 2012.

Yifei Zou received the B.E. degree in 2016 from Computer School, Wuhan University, and the PhD degree in 2020 from the Department of Computer Science, The University of Hong Kong. He is currently an Assistant Professor with the school of computer science and technology, Shandong University. His research interests include wireless networks, ad hoc networks and distributed computing.

Peng Wang received his MS degree in engineering at the school of Control Science and Engineering of Shandong University in 2005. He is currently pursuing a Ph.D. degree in Computer Science from Shandong University. His research interests includes wireless network, edge computing, and distributed computing.

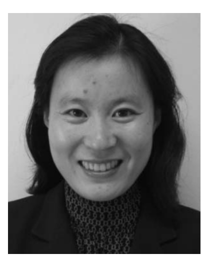
Shikun Shen received the B.E. degree from the Computer School, Wuhan University, in 2022. He is currently pursuing his MS degree with the School of Computer Science and technology, Shandong University. His research interests include wireless networks, distributed computing, and edge intelligence.



Dongxiao Yu received the BSc degree in 2006 from the School of Mathematics, Shandong University and the PhD degree in 2014 from the Department of Computer Science, The University of Hong Kong. He became an associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2016. He is currently a professor in the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing and graph algorithms.

Jorge Torres Gómez is with the School of Electrical Engineering and Computer Science, TU Berlin. From 2008 to 2018, he lectured at the School of Telecommunications and Electronics, Technological University José Antonio Echeverría in Cuba. He has been with the Department of Signal Theory and Communications, UC3M, Spain, as a guest lecturer and with the TU Chemnitz as a postdoc. He is on the Educational and Professional Activities committee in the IEEE German Chapter Executive Committee. His research interests include wireless communications, molecular communications, DSP, SDR, and Education.

Falko Dressler received his M.Sc. and Ph.D. degrees from the Dept. of Computer Science, University of Erlangen in 1998 and 2003, respectively. He is a full professor and Chair for Data Communications and Networking at the School of Electrical Engineering and Computer Science, TU Berlin. Dr. Dressler has been associate editor-in-chief for IEEE Trans. on Mobile Computing and Elsevier Computer Communications as well as an editor for journals such as IEEE/ACM Trans. on Networking, IEEE Trans. on Network Science and Engineering, Elsevier Ad Hoc Networks, and Elsevier Nano Communication Networks. He has been chairing conferences such as IEEE INFOCOM, ACM MobiSys, ACM MobiHoc, IEEE VNC, IEEE GLOBECOM. He authored the textbooks Self-Organization in Sensor and Actor Networks published by Wiley & Sons and Vehicular Networking published by Cambridge University Press. He has been an IEEE Distinguished Lecturer as well as an ACM Distinguished Speaker. Dr. Dressler is an IEEE Fellow as well as an ACM Distinguished Member. He is a member of the German National Academy of Science and Engineering (acatech). He has been serving on the IEEE COMSOC Conference Council and the ACM SIGMOBILE Executive Committee. His research objectives include adaptive wireless networking (radio, visible light, molecular communications) and embedded system design (from microcontroller to Linux kernel) with applications in ad hoc and sensor networks, the Internet of Things, and cooperative autonomous driving systems.



Xiuzhen Cheng received her M.S. and Ph.D. degrees in computer science from the University of Minnesota – Twin Cities in 2000 and 2002, respectively. She is a professor in the School of Computer Science and Technology, Shandong University. Her current research interests include cyber physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time), and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is Fellow of IEEE and a member of ACM.