International Journal of Innovative Computing, Information and Control
Volume **18**, Number **2**, April **2022**
ICIC International ©2022 ISSN 1349-4198
pp. **497–509**

DATA COLLECTION ALGORITHM FOR INTERNET OF THINGS BASED ON AGE OF INFORMATION AND SAMPLE EXTRUSION AWARENESS

Wei Hao, Hengzhou Ye* and Fengyi Huang

Guangxi Key Laboratory of Embedded Technology and Intelligent System
Guilin University of Technology
No. 319, Yanshan Street, Yanshan District, Guilin 541006, P. R. China
{ 1020190598; 2120190614 }@glut.edu.cn; *Corresponding author: 2002018@glut.edu.cn

Received October 2021; revised January 2022

Abstract. *As an emerging metric for information freshness, AoI (Age of Information) is widely used in time-sensitive systems. As a time-sensitive system, the IoT (Internet of Things) system allocates link resources to source nodes for data transmission so that users can obtain fresh data and quality services, which is particularly important. However, in the existing research work, for the IoT system composed of multiple sources and a common base station, the algorithms adopted only consider minimizing AoI and ignore the existence of sample extrusion phenomenon. Sample extrusion is a phenomenon that occurs when the collected sample stored in the buffer of the source node has not been completely transmitted to the base station, and the source node has collected another new sample. Based on a greedy strategy, we propose a link resource allocation algorithm that considers both AoI and sample extrusion to collect the data from various source nodes. Simulation experiments show that the proposed algorithm can achieve better comprehensive performance than the other existing algorithm.*

Keywords: Age of information, Time-sensitive, Link resource allocation, Sample extrusion, Data collection

1. **Introduction.** A large number of sensor source nodes in the IoT (Internet of Things) system collect surrounding information and send the information to the data processing center in a timely manner. The data processing center processes and stores the data or forwards it to the edge of the network or the cloud [1]. The outdated information will have a significant impact on control decisions. However, due to the limited link resources, not all information can be sent to the destination node in time. It is particularly important to design a resource allocation strategy and schedule each node to collect data. At present, a lot of research work has focused on scheduling nodes based on traditional metrics, such as latency and throughput, to ensure the collection of real-time state information [2-4], but they cannot fully reflect the freshness of data. For example, when the information is updated less frequently at the source node, i.e., the throughput is small, the latest information received at the destination node is still out of date due to the lack of updates, even though the delay during transmission is small. Similarly, when the transmission delay at the source node is large, even if the information at the source node is updated more frequently, the information received at the destination node is still outdated. Therefore, the choice of measurement method still needs further consideration.

AoI (Age of Information) can be effective in avoiding the above problems. AoI was first proposed in [5] and is defined as the elapsed time between the current time and the

DOI: 10.24507/ijicic.18.02.497

generation time of the latest received sample, which measures the freshness of the data from the perspective of the destination node. AoI is different from traditional delay and throughput [6,7]. It is a metric that can simultaneously capture delay and throughput. When calculating the AoI of each source node at the destination node, the generation time of the sample that arrived at the destination node before is covered with the generation time of the newly arrived sample. The AoI of each source node at the destination node includes not only the delay of the newly arrived sample but also the time difference until the next sample arrival. AoI has been widely used in many scenarios, such as UAV sensing services [8], industrial wireless networks [9], broadcast wireless networks [10], and cloud games [11].

At present, the research work on AoI is relatively extensive, such as the research on the application of information-aware scheduling in multi-source systems. Due to the limited communication capacity, not all sensors can send updates to the monitoring station at any time. Therefore, the scheduling algorithms are designed to minimize AoI [12,13]. In addition, some authors introduced queues and performed queue scheduling on source node information. They studied the AoI of different queues and selected the appropriate queuing strategies to minimize AoI [14,15]. Furthermore, several authors incorporated the effect of energy in their system model, and the best strategy for joint optimization of wireless energy transmission and update packet transmission scheduling of IoT devices is studied with the goal of minimizing long-term weighted AoI [16]. Shreedhar et al. [17] developed the AoI control protocol, Wang and Duan [18] considered the dynamic pricing for controlling AoI, and Xie et al. [19] studied the impact of coding schemes on AoI.

However, most of the work considers only the optimization objective of AoI and ignores the existence of the sample extrusion problem. Sample extrusion refers to the problem that when a sample is collected at the source node and sent to the destination node, the sample is not delivered to the destination node in time due to the limited link capacity, and a new sample is collected at the source node before the old sample in the buffer of the source node is completely transmitted to the destination node. Once the sample extrusion problem occurs, for the source node with only one buffer, it results in data loss. The choice between a new sample and an old sample must be made at each extrusion, and one of the samples must be discarded. In addition, the time taken to receive samples from this source node at the destination node becomes longer, and the freshness of the data is unguaranteed. As in [20] and [21], the Juventas scheduling algorithm and the Kronos scheduling algorithm proposed by the authors both finish transmitting the sample that have not been completely transmitted in the previous time slot before scheduling the other samples, resulting in the transmission of the large sample causing other samples to lose transmission opportunities and causing sample extrusion. In [22], Abbas et al. considered a system model in which all packet sizes are equal. They devised a discrete time Markov chain model to study the effects of the arrival rate of packets at the nodes, the number of nodes in the system, and queue length of each node on the system performance to reduce the overall AoI of the system. In [23], Feng and Yang designed an optimal online status updating policy to minimize the long-term average AoI at the destination while satisfying the energy causality constraint at the sensor, without considering the size of each transmitted update. In [24], centralized and distributed resource management schemes are proposed to allocate the limited communication resources to minimize AoI. In [9], the authors studied the scheduling problem of a multi-source system where sources report their time-varying information to a central monitoring station via multiple orthogonal channels and designed a scheduling strategy based on the weighted graphs to minimize AoI.

In time-sensitive IoT applications, if sample extrusion occurs, the period of information received at the destination node will increase and some samples may be lost, which seriously affects the quality of user experience. Therefore, a resource allocation algorithm is urgently needed to select and transmit the samples from heterogeneous source nodes in the IoT to achieve a balance between AoI and sample extrusion. In this paper, we design an algorithm based on a greedy strategy to solve it. The specific contributions are summarized as follows.

- Based on the AoI self-update process at the destination node, we give two independent mathematical representations of the optimization objectives of AoI and the number of sample extrusions for the analysis and design of the scheduling algorithm based on them.
- We describe and prove the relationship between AoI and sample extrusion, demonstrating the necessity to consider these two objectives separately in the design of the scheduling algorithm.
- In order to optimize AoI, we construct a representation of AoI gain. In order to minimize the number of sample extrusions, the possibility of the next sample arrival is described. We combine the above aspects to design a reasonable selection variable and propose a scheduling algorithm based on a greedy strategy to optimize AoI and sample extrusion.

The rest of the paper is organized as follows. The system model and two optimization objectives are introduced in Section 2. Section 3 presents the relationship between AoI and sample extrusion. Section 4 introduces the algorithm design process. The results of the performance evaluation are given in Section 5. Finally, conclusions are shown in Section 6.

2. Problem Formulation.

2.1. **System model.** In this paper, an IoT data collection system consisting of N heterogeneous source nodes and a BS (Base Station) is used. The source nodes collect samples periodically and send the data to the BS through a shared channel in a single-hop wireless network, as shown in Figure 1.

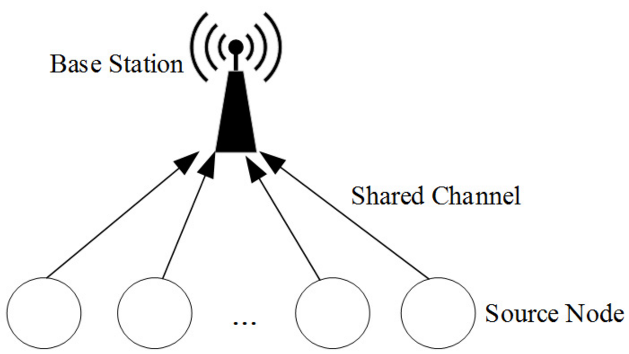


FIGURE 1. System model

As in [20] and [21], we use an advanced transmission technique that simultaneously divides the time domain and the frequency domain, dividing the time into multiple equal-sized time slots and dividing the bandwidth into multiple equal-sized bandwidth segments. Thus, the transmission resources are organized into a series of data transmission units spanning the time domain and the frequency domain, and these data transmission units can be allocated to multiple source nodes for sample transmission in each time slot. Due to the limited link capacity, M data transmission units in the uplink bandwidth must be reasonably allocated to N source nodes in each time slot. Each transmission data unit can be allocated to at most one source node. The ith source node is denoted as S_i, and $x_i(t)$ is used to represent the number of data transmission units allocated to S_i at time slot t. Since transmission takes a certain amount of time, each source node can transmit data only once in a time slot, i.e., only the samples of the source nodes with $x_i(t) = 0$ can continue to be selected for data transmission at time slot t. The total number of data transmission units allocated to all source nodes in a time slot must not exceed the link capacity. Hence, we have Equation (1).

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

2.2. **Optimization objectives.**

2.2.1. *AoI.* Only when all units of the sample from S_i arrive at the BS, we say that the BS has received this sample. Denote $U_i^{\mathrm{B}}(t)$ and $A_i^{\mathrm{B}}(t)$ as the generation time of the latest received sample from S_i at the BS at time slot t and the AoI from S_i at the BS at time slot t, respectively. According to the definition of AoI, $A_i^{\mathrm{B}}(t)$ is calculated as Equation (2).

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

Meanwhile, in order to reflect the difference in the importance of the information on each source node, a weight is assigned to each source node. Denote W_i as the weight of S_i. Using the weight normalization function, the normalized result w_i of S_i is obtained, which can be expressed as Equation (3).

$$w_i = W_i \Big/ \sum_{i=1}^{N} W_i \tag{3}$$

One of the target values considered in this paper, i.e., the long-term average weighted AoI, is represented by $\bar{A}^{\mathrm{B}}$ and can be determined by Equation (4).

$$\bar{A}^{\mathrm{B}} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} \left(w_i A_i^{\mathrm{B}}(t) \right) \tag{4}$$

2.2.2. *Number of sample extrusions.* In the process of collecting samples from the source nodes, when the link capacity is limited, if there are source nodes with small sampling periods or source nodes whose collected samples are too large to be completely transmitted, the sample extrusion is more likely to occur.

$x(t) = \{<1, x_1(t)>, <2, x_2(t)>, \ldots, <N, x_N(t)>\}$ is used to represent the scheduling sequence at time slot t. The sampling period of S_i is denoted as T_i, and the sampling size of S_i is denoted as L_i. $BL_i(t)$ represents the remaining sample size of S_i to be transmitted when the transmission of time slot t has not yet started. Assuming that at time slot t_0, each source node has collected a new sample to be transmitted, and there is a source node S_K that has a large sample waiting to be transmitted. The link capacity in each TTI is sufficient to transmit samples other than the large samples collected by S_K. If we use the non-preemptive scheduling algorithm JUVENTAS in [20], the following scheduling scheme exists: $x(t_0) = \{<1,0>, <2,0>, \ldots, <K,M>, \ldots, <N,0>\}$, $x(t_0+1) = \{<1,0>, <2,0>, \ldots, <K,M>, \ldots, <N,0>\}$, $x(t_0+2) = \{<1,0>, <2,0>, \ldots, <K,M>, \ldots, <N,0>\}$. At this time, once there are newly collected samples at time slots t_0+1 and t_0+2, sample extrusion will occur. That is, the samples collected at time slot t_0 have not yet been completely transmitted to the BS, and the source nodes collect new samples at time slot t_0+1. The samples at time slot t_0+1 have not been transmitted yet, and the latest samples have been collected at time slot t_0+2. If the following scheduling sequence is used, the generation of sample extrusion can be avoided: $x(t_0) = \{<1,0>, <2,0>, \ldots, <K,M>, \ldots, <N,0>\}$, $x(t_0+1) = \Big\{<1, BL_1(t_0+1)>, <2, BL_2(t_0+1)>, \ldots, <K, \max\Big(\min\Big(M - \sum_{i=1}^{N} BL_i(t_0+1) - BL_K(t_0+1), BL_K(t_0+1)\Big), 0\Big)>, \ldots, <N, BL_N(t_0+1)>\Big\}$, $x(t_0+2) = \Big\{<1, BL_1(t_0+2)>, <2, BL_2(t_0+2)>, \ldots, <K, \max\Big(\min\Big(M - \sum_{i=1}^{N} BL_i(t_0+2) - BL_K(t_0+2), BL_K(t_0+2)\Big), 0\Big)>, \ldots, <N, BL_N(t_0+2)>\Big\}$.

In this paper, we assume that the initial sampling time slot of all source nodes is 0 and use F_i to denote the initial sampling time slot of S_i. Denote $WT_i(t)$ as the number of S_i samples to be transmitted at the beginning of time slot t when S_i has not yet performed new sample collection. Sample extrusion occurs at the sampling time slot of the source node. Therefore, we observe sample extrusion at the sampling time slot. If there is a sample at the source node that has not been completely transmitted before, sample extrusion will occur. If there is no such sample, or if it is not currently a sampling time slot, sample extrusion will not occur. We use $D_i(t)$ to indicate whether sample extrusion occurs. Hence, we have Equation (5).

$$D_i(t) = \begin{cases} \min(1, WT_i(t)), & \text{if } (t - F_i) \bmod T_i = 0 \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

The average number of sample extrusions $\overline{E}$ is used to reflect the degree of sample extrusion at the source nodes, i.e., the other optimization objective of this paper, which is calculated as Equation (6).

$$\overline{E} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{i=1}^{N} D_i(t) \tag{6}$$

3. **Relationship between AoI and Sample Extrusion.** The process of sample extrusion and AoI changes is as follows. After the sample I_1 of S_i arrives at the BS, a new sample I_2 is collected and does not completely reach the BS in the current time slot. At this time, the AoI from S_i at the BS is T_i because I_1 of S_i is still retained at the BS. Before the next sample I_3 is collected, the current sample I_2 cannot be completely transmitted to the BS due to the limitation of the link capacity, and the AoI continues increasing. At this time, the value range of the AoI from S_i at the BS belongs to $(T_i, 2T_i - 1]$. When the new sample I_3 is collected in the next sampling time slot, I_2 has not completely reached the BS, and the sample extrusion occurs. At this time, if the latest sample I_3 completely reaches the BS, the AoI from S_i at the BS is 0, if the old sample I_2 completely reaches the BS, the AoI from S_i at the BS is T_i, and if neither of them reaches the BS, the AoI from S_i at the BS is $2T_i$. After that, the source nodes continue to sample and transmit, and the AoI from the source nodes on the BS is updated. A new sample must be collected in the sampling time slot, so the sample extrusion can be observed in the time slot before the sampling time slot. We assume that t is the sampling time slot, i.e., $(t - F_i) \bmod T_i = 0$. If $A_i^{\mathrm{B}}(t-1) \geq 2T_i - 1$, i.e., one or more samples have been collected before the sampling time slot and have not yet reached the BS, the sample extrusion occurs.

For a single source node, there is no simultaneous increase or decrease between AoI and the number of sample extrusions. For example, when the AoI from S_i on the BS is $2T_i - 2$, and the sample is completely transmitted to the BS at this time, no sample extrusion occurs, although the AoI is also large. For the whole IoT system, there are a large number of source nodes. Similarly, there is no homogeneous increasing and decreasing relationship between AoI and the number of sample extrusions in the whole IoT system. In summary, we must consider the two objectives simultaneously.

4. **Algorithm Design.** Since we must consider both the objectives of optimizing the average weighted AoI and the average number of sample extrusions, the AoI gain expression and the judgment variable of the possibility of the next sample arrival are given in 4.1 and 4.2, respectively. Based on the above, a selection variable is designed by combining 4.1 and 4.2, and a scheduling algorithm GSSA is obtained in 4.3.

4.1. **Minimization of AoI.** Since partial transmission of samples will not contribute to the reduction of AoI, in order to avoid the waste of transmission resources, when sample extrusion occurs: if the old sample has not yet started to be transmitted, the obsolete sample is directly discarded since the amount of resources still needed to transmit the new sample is the same as the old one; if it has been partially transmitted, this old sample is retained at the source node, and the transmission of the remaining part of this sample continues when it is next scheduled. Denote $AL_i(t)$ as the remaining sample size of S_i to be transmitted when the transmission of time slot t has been completed. $n_i(t)$ indicates whether S_i retains the new sample or the old sample at time slot t, and the choice of this is required only at the sampling time slot, which can be described as Equation (7).

$$n_i(t) = \begin{cases} 1, & \text{if } (t - F_i) \bmod T_i = 0 \wedge AL_i(t-1) = 0 \vee AL_i(t-1) = L_i \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$AL_i(t)$ is calculated as Equation (8).

$$AL_i(t) = BL_i(t) - x_i(t) \tag{8}$$

If S_i collects a new sample at time slot t and retains the new sample, $BL_i(t)$ is updated to the newly collected sample size, that is, L_i. Otherwise, the sample size of S_i to be transmitted at time slot t is the remaining sample size after transmission of the previous time slot. Hence, we have Equation (9).

$$BL_i(t) = \begin{cases} L_i, & \text{if } n_i(t) = 1 \\ AL_i(t-1), & \text{otherwise} \end{cases} \tag{9}$$

The generation time of the newly retained sample at S_i at time slot t is denoted as $U_i^{\mathrm{S}}(t)$. Correspondingly, the AoI at S_i at time slot t is denoted as $A_i^{\mathrm{S}}(t)$, whose specific value can be obtained by Equation (10).

$$A_i^{\mathrm{S}}(t) = t - U_i^{\mathrm{S}}(t) \tag{10}$$

When the newly collected sample reaches S_i and is retained, the sample at S_i is updated. At this time, $U_i^{\mathrm{S}}(t) = t$, that is, $A_i^{\mathrm{S}}(t) = 0$. Otherwise, $A_i^{\mathrm{S}}(t)$ continues increasing until the next new sample is retained. Hence, we have Equation (11).

$$A_i^{\mathrm{S}}(t) = \begin{cases} 0, & \text{if } n_i(t) = 1 \\ A_i^{\mathrm{S}}(t-1) + 1, & \text{otherwise} \end{cases} \tag{11}$$

The AoI from the source node at the BS is self-updating. That is, the generation time of the latest completely arrived sample is used in the AoI calculation from the source node at the BS. If a new sample completely arrives at the BS at time slot t, the generation time of this sample is used. Otherwise, the AoI from the source node at the BS continues increasing. Therefore, we have Equation (12).

$$A_i^{\mathrm{B}}(t) = \begin{cases} A_i^{\mathrm{S}}(t), & \text{if a new sample at } S_i \text{ completely arrives at the BS} \\ & \text{at time slot } t \\ A_i^{\mathrm{B}}(t-1)+1, & \text{otherwise} \end{cases} \tag{12}$$

We assume that the sample at S_i can be completely transmitted at time slot t. If it is scheduled for transmission, we have $A_i^{\mathrm{B}}(t) = A_i^{\mathrm{S}}(t)$. Otherwise, we have $A_i^{\mathrm{B}}(t) = A_i^{\mathrm{B}}(t-1)+1$. Thus, the AoI gain from S_i at the BS at time slot t, denoted as $\Delta_i(t)$, can be calculated by Equation (13).

$$\Delta_i(t) = A_i^{\mathrm{B}}(t-1) + 1 - A_i^{\mathrm{S}}(t) \tag{13}$$

4.2. **Avoidance of sample extrusion.** If the sample has not completely arrived at the BS in the current sampling period, when the next sampling period arrives, the arrival of a new sample will cause the phenomenon of sample extrusion. Therefore, to alleviate the sample extrusion, we use $p_i(t)$ to describe the probability of the next sample arrival of S_i at time slot t and give priority to transmitting the sample at the source node with a larger $p_i(t)$. The specific value of $p_i(t)$ can be calculated by Equation (14).

$$p_i(t) = \frac{1 + (t - F_i) \bmod T_i}{T_i} \tag{14}$$

The non-preemptive strategy prioritizes the transmission of the sample that has not been completely transmitted in the previous time slot. Only after the transmission of this sample is completed, other sample information can be transmitted, which is likely to cause sample extrusion at other more source nodes. Therefore, we adopt a more flexible preemptive scheduling strategy to avoid the above problem. The preemptive scheduling strategy is not able to transmit some large samples as soon as possible. Instead, AoI and sample extrusion are considered comprehensively, and these link resources are used to transmit more small samples to reduce the extrusion of small samples. Since only a part of the large sample is transmitted, and the partially arriving samples cannot really reduce the AoI from their source nodes at the BS, some of the AoI is sacrificed. In terms of the overall IoT system, it is necessary to allow more source nodes to transmit instant sample information.

4.3. **Proposed algorithm: GSSA.** Combining 3.1, 3.2 and the link capacity required for each source node to complete the transmission of the remaining sample at the node, we can calculate the impact of the unit link capacity on AoI and sample extrusion, assuming that each sample can completely reach the BS. In addition, the influence of the weight of each node is taken into account. The calculation method of w_i is shown in Equation (3), and w_i is the normalized result of W_i. Therefore, we design a selection variable $ST_i(t)$, which can be calculated by Equation (15).

$$ST_i(t) = \frac{\Delta_i(t) p_i(t) w_i}{BL_i(t)} \tag{15}$$

In summary, we propose a source node sample scheduling algorithm GSSA based on a greedy strategy. The time cost of the algorithm mainly lies in finding the source node with $BL_i(t) > 0$ and the largest $ST_i(t)$ from the nodes that have not been scheduled in the current time slot, whose time complexity is $O(N)$. In the worst case, the link capacity in a time slot is sufficient for each source node to transmit data, and each source node must be allocated data transmission units, whose time complexity is $O(N^2)$. Therefore, the time complexity of the algorithm GSSA is $O(N^2)$.

Denote $RM(t)$ as the number of remaining available data transmission units at time slot t, which is updated in real time during the scheduling process. Only when $RM(t) > 0$, other source node samples can be selected to continue transmission. The update method of $RM(t)$ can be described as Equation (16).

$$RM(t) = M - \sum_{i=1}^{N} x_i(t) \tag{16}$$

The specific process is shown in Algorithm 1.

Algorithm 1. GSSA

Input: W_i, L_i, T_i, F_i, t, N, M;
Output: $x(t)$;
$RM(t) = M$;
while $RM(t) > 0$ **do**
  Find S_i with $BL_i(t) > 0$ and the largest $ST_i(t)$ among the nodes that have not been scheduled in the current time slot;
  if S_i does not exist **then**
    Exit while loop;
  else
    if $BL_i(t) <= RM(t)$ **then**
      Add $<i, BL_i(t)>$ to $x(t)$;
      $AL_i(t) = 0$;
      $WT_i(t+1) = 0$;
      $RM(t) = RM(t) - BL_i(t)$;
    else
      Add $<i, RM(t)>$ to $x(t)$;
      $AL_i(t) = BL_i(t) - RM(t)$;
      $RM(t) = 0$;
    end if
  end if
end while
if S_j is not scheduled **then**
  Add $<j, 0>$ to $x(t)$;
end if

5. **Performance Evaluation.** This experiment is run on a PC with an Intel (R) Core (TM) i7-4720HQ, 2.60 GHz CPU, 8 GB memory, and 64-bit Windows 10 operating system. The experimental code is written in Java 1.7.

In the experimental simulation, two scenarios are considered: one is the same scenario as in [20] (denoted as G1, as shown in Table 1), and the other is a scenario where the sampling size variance between each source node is large (denoted as G2, as shown in Table 2). The GSSA algorithm proposed in this paper is compared with the JUVENTAS algorithm proposed in [20] in terms of both the weighted average AoI value and the average number of sample extrusions. We assume that each source node collects at least 100 samples.

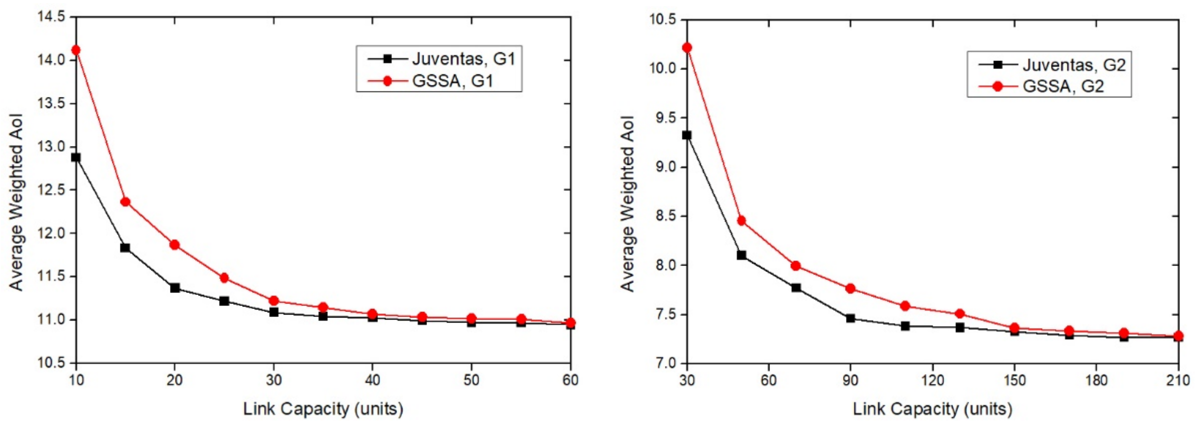
5.1. **The impact of link capacity on the average weighted AoI.** The impact of link capacity on the average weighted AoI of the two algorithms can be observed in Figure 2. As the link capacity increases, more data transmission units are available for allocation, and the average weighted AoI values of the two algorithms gradually decrease and stabilize to a fixed value. Figure 2(a) shows that in the G1 scenario, the average weighted AoI

TABLE 1. G1 scenario

Source node	W_i	L_i (units)	T_i (time slots)
1	6	1	10
2	33	15	12
3	25	11	45
4	39	10	2
5	36	19	25
6	46	13	9
7	35	13	49
8	17	18	36
9	35	17	26
10	10	12	24

TABLE 2. G2 scenario

Source node	W_i	L_i (units)	T_i (time slots)
1	7	10	7
2	6	1	3
3	15	70	10
4	14	30	5
5	20	50	17
6	8	3	1
7	26	65	20
8	25	5	7
9	33	90	35
10	27	20	12

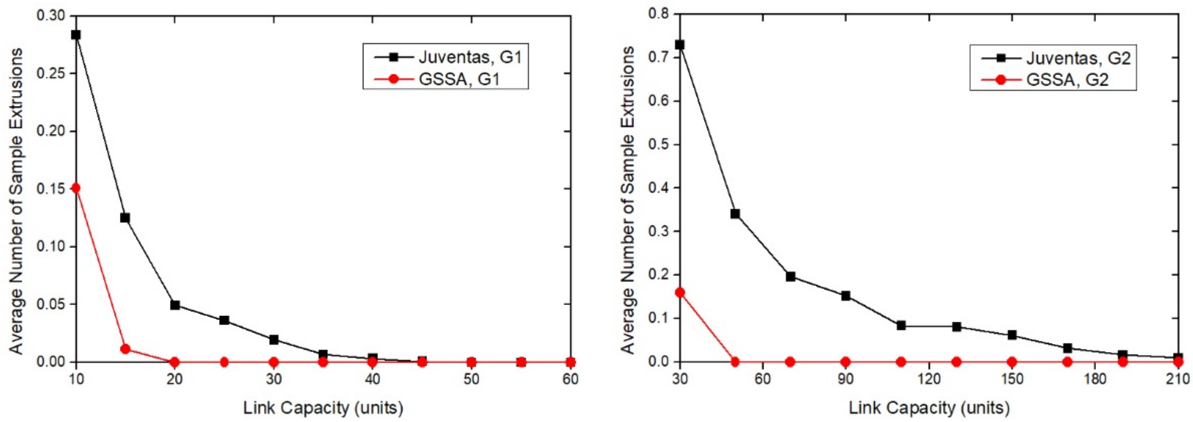


(a) In the G1 scenario (b) In the G2 scenario

FIGURE 2. The impact of link capacity on the average weighted AoI

obtained by GSSA is larger than that obtained by Juventas. Figure 2(b) shows that in the G2 scenario, the average weighted AoI obtained by GSSA is larger than that obtained by Juventas. Large samples are difficult to be completely transmitted, and the AoI from this source node at the BS will continue growing and will be large in the later period. Juventas transmits the remaining sample that has only been partially transmitted in the previous TTI as soon as possible, which can prevent the growth of the AoI at the BS faster than GSSA. This is due to the fact that the AoI from large samples at the BS tends to be larger than that of small samples because of the difficulty in completing the transmission. GSSA comprehensively selects samples based on the selection variable and may not continue transmitting the large sample. However, partial transmission of the sample will not contribute to the reduction of AoI. The difference between the results obtained by the two algorithms is always small in both G1 and G2 scenarios, especially when the link capacity is large, the results obtained by the two algorithms almost overlap.

5.2. **The impact of link capacity on the number of sample extrusions.** Figure 3 shows the impact of link capacity on the average number of sample extrusions. As the link capacity increases, the number of data transmission units available for allocation increases, and the average sample extrusion number for both algorithms gradually decreases and converges towards zero. Figure 3(a) shows that in the G1 scenario, the average sample extrusion number of GSSA is always less than that of Juventas when the link capacity is less than 45. GSSA reaches the zero point when the link capacity is 20, while Juventas does not approach the zero point until 45, which indicates that GSSA requires less link resources to eliminate the sample extrusion. Figure 3(b) shows that in the G2 scenario, the number of sample extrusions of GSSA is always smaller than that of Juventas, and the impact is more pronounced than that of the G1 scenario. The GSSA sample extrusion number reaches zero when the link capacity is 50, while Juventas still has a small number of extruded samples when the link capacity is 210. If Juventas is used, the large sample that has not been completely transmitted in the previous TTI always is transmitted first, which occupies more TTIs to transmit this sample, preventing other samples from being transmitted. When the source nodes collect new samples, they will be extruded. However, GSSA makes flexible choices. The preemptive strategy is adopted to allow the sample with a higher possibility of extrusion to be transmitted within a TTI without waiting for the completely transmission of the large sample. In summary, GSSA has a significant impact on sample extrusion.



(a) In the G1 scenario (b) In the G2 scenario

FIGURE 3. The impact of link capacity on the number of sample extrusions

5.3. **The impact of duration on the running time.** The duration of the algorithm refers to the number of times that each source node is guaranteed to collect samples. Figure 4 shows the impact of duration on the running time when the link capacity is 50. As the duration increases, the running time of the two algorithms in the two scenarios also increases. Unlike Juventas, it is unnecessary for GSSA to determine whether the

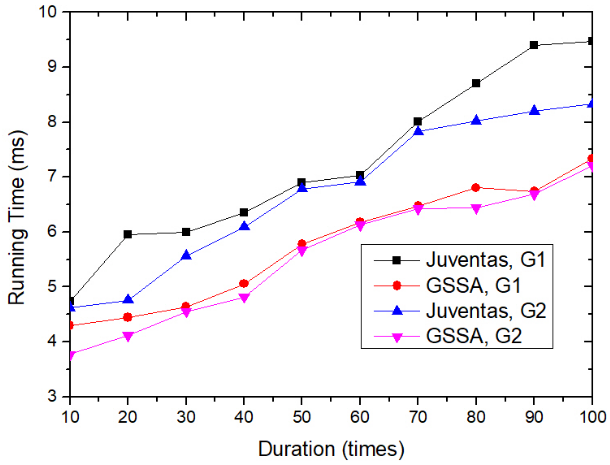


FIGURE 4. The impact of duration on the running time

sample transmitted in the previous TTI has been completely transmitted before allocating resources in the current TTI. Therefore, the running time of GSSA is shorter than Juventas in both scenarios. In addition, the average sample size in the G2 scenario is larger than that in the G1 scenario, so the average scheduling times in each TTI is relatively less in the G2 scenario. Therefore, the running time of the two algorithms in the G2 scenario is shorter than that in the G1 scenario.

6. **Conclusions.** AoI, a freshness of information measure, has received extensive attention when studying the node scheduling problem in time-sensitive applications. Aiming at the data collection problem in the multi-source IoT system, the existing scheduling algorithms only consider the minimization of AoI and ignore the existence of the sample extrusion problem. In this paper, we propose a resource allocation algorithm GSSA based on a greedy strategy, which simultaneously considers the two optimization goals of AoI and sample extrusion. Simulation experimental analysis shows that the comprehensive performance of the algorithm proposed in this paper is better than the other existing algorithm. In this work, the unreliability and dynamics of the channel have not been taken into account, and only the case in single-hop networks has been studied rather than multi-hop networks. In future work, we will improve on the above aspects.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (Grant No. 61802085), the Guangxi Natural Science Foundation (Grant No. 2020GXNSFAA159038), the Foundation of Guilin University of Technology (Grant No. GUTQDJJ2002018), the Guangxi Key Laboratory of Trusted Software Foundation (Grant No. kx202011), and the Guangxi Universities Key Laboratory Director Fund of Embedded Technology and Intelligent Information Processing (Grant No. 2020-1-7). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] S.-C. Wang, W.-S. Hsiung, C.-F. Hsieh and Y.-T. Tsai, Optimal consensus achievement for the Internet of Things based on fog computing within dual faulty transmission media, *ICIC Express Letters, Part B: Applications*, vol.10, no.9, pp.773-780, 2019.

[2] B. Kang, P. Nguyen and H. Choo, Delay-efficient energy-minimized data collection with dynamic traffic in WSNs, *IEEE Sensors Journal*, vol.18, no.7, pp.3028-3038, 2018.

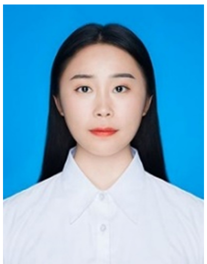
[3] C. C. Lin and C. Y. Chang, An energy balanced data collection mechanism for maximizing throughput using uncontrolled mobile sink in WSNs, *Proc. of the 2020 IEEE International Conference on Consumer Electronics – Taiwan (ICCE-Taiwan)*, Taoyuan, Taiwan, pp.1-2, 2020.
[4] Y. Liu, Y. Zhou, J. Yuan and L. Liu, Delay aware flow scheduling for time sensitive fronthaul networks in centralized radio access network, *IEEE Trans. Communications*, vol.68, no.5, pp.2992-3009, 2020.
[5] S. Kaul, M. Gruteser, V. Rai and J. Kenney, Minimizing age of information in vehicular networks, *Proc. of the 2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Salt Lake City, UT, USA, pp.350-358, 2011.
[6] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh and E. Modiano, Scheduling policies for minimizing age of information in broadcast wireless networks, *IEEE/ACM Trans. Networking*, vol.26, no.6, pp.2637-2650, 2018.
[7] R. Talak, S. Karaman and E. Modiano, When a heavy tailed service minimizes age of information, *Proc. of the 2019 IEEE International Symposium on Information Theory (ISIT)*, Paris, France, pp.345-349, 2019.
[8] F. Wu, H. Zhang, J. Wu, Z. Han, H. V. Poor and L. Song, UAV-to-device underlay communications: Age of information minimization by multi-agent deep reinforcement learning, *IEEE Trans. Communications*, vol.69, no.7, pp.4461-4475, 2021.
[9] M. Li, C. Chen, H. Wu, X. Guan and X. Shen, Age-of-information aware scheduling for edge-assisted industrial wireless networks, *IEEE Trans. Industrial Informatics*, vol.17, no.8, pp.5562-5571, 2021.
[10] Y. Hsu, E. Modiano and L. Duan, Scheduling algorithms for minimizing age of information in wireless broadcast networks with random arrivals, *IEEE Trans. Mobile Computing*, vol.19, no.12, pp.2903-2915, 2020.
[11] R. D. Yates, M. Tavan, Y. Hu and D. Raychaudhuri, Timely cloud gaming, *Proc. of the IEEE INFOCOM 2017 – IEEE Conference on Computer Communications*, Atlanta, GA, USA, pp.1-9, 2017.
[12] R. D. Yates and S. K. Kaul, The age of information: Real-time status updating by multiple sources, *IEEE Trans. Information Theory*, vol.65, no.3, pp.1807-1827, 2019.
[13] B. Buyukates, A. Soysal and S. Ulukus, Age of information scaling in large networks, *Proc. of the ICC 2019 – 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, pp.1-6, 2019.
[14] C. Kam, J. P. Molnar and S. Kompella, Age of information for queues in tandem, *Proc. of the MILCOM 2018 – 2018 IEEE Military Communications Conference (MILCOM)*, Los Angeles, CA, USA, pp.1-6, 2018.
[15] Y. Inoue, H. Masuyama, T. Takine and T. Tanaka, A general formula for the stationary distribution of the age of information and its application to single-server queues, *IEEE Trans. Information Theory*, vol.65, no.12, pp.8305-8324, 2019.
[16] M. A. Abd-Elmagid, N. Pappas and H. S. Dhillon, On the role of age of information in the Internet of Things, *IEEE Communications Magazine*, vol.57, no.12, pp.72-77, 2019.
[17] T. Shreedhar, S. K. Kaul and R. D. Yates, ACP: Age control protocol for minimizing age of information over the Internet, *Proc. of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom'18)*, New York, NY, USA, pp.699-701, 2018.
[18] X. Wang and L. Duan, Dynamic pricing for controlling age of information, *Proc. of the 2019 IEEE International Symposium on Information Theory (ISIT)*, Paris, France, pp.962-966, 2019.
[19] M. Xie, Q. Wang, J. Gong and X. Ma, Evaluation of age of information for LDPC coded transmission over AWGN channels, *Proc. of the 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, Kuala Lumpur, Malaysia, pp.1-5, 2019.
[20] C. Li, S. Li and Y. T. Hou, A general model for minimizing age of information at network edge, *Proc. of the IEEE INFOCOM 2019 – IEEE Conference on Computer Communications*, Paris, France, pp.118-126, 2019.
[21] C. Li, Y. Huang, Y. Chen, B. Jalaian, Y. T. Hou and W. Lou, Kronos: A 5G scheduler for AoI minimization under dynamic channel conditions, *Proc. of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, pp.1466-1475, 2019.
[22] Q. Abbas, S. A. Hassan, H. Pervaiz and Q. Ni, A Markovian model for the analysis of age of information in IoT networks, *IEEE Wireless Communications Letters*, vol.10, no.7, pp.1596-1600, 2021.
[23] S. Feng and J. Yang, Age of information minimization for an energy harvesting source with updating erasures: Without and with feedback, *IEEE Trans. Communications*, vol.69, no.8, pp.5091-5105, 2021.

[24] T. Park, W. Saad and B. Zhou, Centralized and distributed age of information minimization with nonlinear aging functions in the Internet of Things, *IEEE Internet of Things Journal*, vol.8, no.10, pp.8437-8455, 2021.

Author Biography

Wei Hao received the B.S. degree from Tiangong University, Tianjin, China, in 2019. She is currently pursuing the M.S. degree at Guilin University of Technology, Guangxi, China. Her research interests include age of information and mobile edge computing.

Hengzhou Ye received the B.S. and Ph.D. degrees from Guilin University of Technology and Guangxi University, Guangxi, China, in 2002 and 2019, respectively. He is currently a Full Professor with Guilin University of Technology. His research interests include age of information, service composition, mobile edge computing, and multi-objective optimization.



Fengyi Huang received the B.S. degree from Wuhan College, Hubei, China, in 2019. She is currently pursuing the M.S. degree at Guilin University of Technology, Guangxi, China. Her research interests include age of information and mobile edge computing.