Age of information: in Systems with Multi-source, Limited Buffers, and LCFS-S

Kangrui Li[1], Xiang Ji[1], Zicong Huang[1], and Shujie Yang[1]
[1]State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China

In recent years, an increasing number of real-time applications have become more sensitive to the freshness of information, which requires that packets reach the receiver as promptly as possible. As a measure of information freshness, it is of great interest to measure the age of information (AoI) on multi-source networks. In this paper, we propose a new queueing system: the systems with N sources, Single buffer, Non-source-aware, and LCFS-S (NSLS-Q system). To simplify the study, we first studied the queueing system for Two sources, Single buffer, Non-source-aware, and LCFS-S (TSLS-Q system). We then generalize the conclusions to the NSLS-Q system. We model the queueing system using a stochastic hybrid system (SHS) to solve for the age of information in the queueing system. In this, Markov chains are used to represent the state transitions. We then compared the system with other queueing systems through numerical results. The results show that the queue model performs better in terms of AoI compared to the traditional queue model.

Index Terms—**Age of information, markov chain, stochastic hybrid systems, multi-source queueing systems, random processes, communication networks.**

I. Introduction

There is a class of systems that are only interested in up-to-date information about the content, such as autopilots, monitors, and vital signs monitors. In this type of system, untimely information can easily lead to accidents. The freshness of the information is very important. Traditional metrics such as throughput, delay, etc. are not sufficient to describe and evaluate such systems, so in recent years a new metric has been introduced: age of information, which represents how fresh the information is. This concept was originally proposed in the context of vehicular networks [1] [2]. Later, AoI was introduced into the network model. Solving AoI for different queueing systems has become a hot topic of research in recent years.

The study of queueing systems was first presented in [3]. The authors of [3] analysed the AoI of the three queueing systems M/M/1, M/D/1 and D/M/1. Since then, the solution of the AoI of queueing systems has been developed. The work [4] [5] verified conclusions of [3] by experiment. The work [6] further derives formulas for AoI of M/M/1/1, M/M/1/2, and M/M/1/2*. The authors of [7]–[12] examine the AoI of M/G/1, G/M/1, and G/G/1 queueing systems. The above work is to solve the model of the infinite queue of First-Come, First-Served (FCFS) systems. In FCFS systems, three methods exist in the past literature to optimise the system AoI: reducing the buffer size [13] [14], using random deadlines [14]–[17], and changing the packet replacement rules [6] [14] [18] [19].

After studying single-source systems, numerous studies began to study multi-source systems [20]–[31]. The authors of [25] [26] discuss multi-source, multi-hop wireless networks. The authors of [27] propose energy-efficient variants of the two scheduling policies to reduce the system energy consumption.

Manuscript received December 31, 2022; revised March 23, 2023. Corresponding author: Shujie Yang (email: sjyang@bupt.edu.cn).

Two Last-Come, First-Served (LCFS) preemption policies are proposed in [20]: LCFS-S and LCFS-W. LCFS-S is the LCFS policy that allows preemption of packets in queues and services; LCFS-W is the LCFS policy that only allows preemption of packets in queues.

For the current papers on optimising AoI, excluding different scenarios and changes to the queuing model, it is often a matter of finding the optimal data update rate λ. and the best service rate μ. Existing AoI optimisation strategies include: changing the service policy of the system from the traditional FCFS to LCFS; modifying the priority of the queuing system, where AoI-sensitive services have a high priority, essentially a priority queue system (priority queue). There are also a variety of optimisation strategies, and the corresponding queuing theory models are highly variable. Therefore, we propose a new **q**ueueing system: the systems with **N** sources, **S**ingle buffer, Non-source-aware, and **LCFS-S** (**NSLS-Q** system). We will refer to this queuing system as **NSLS-Q** for short. To simplify the study, we first studied the **q**ueueing system for **T**wo sources, **S**ingle buffer, Non-source-aware, and **LCFS-S** (**TSLS-Q** system). We then extrapolated our conclusions to N sources.

A. Contributions

The main contributions of this paper are as follows:

In this system we first studied the **TSLS-Q** system. Specifically, the system has the following characteristics: there are two sources, and both sources share the same server; the system (server and waiting queue) contains at most two packets, which means that, in addition to the packet being served, the queue contains at most one waiting packets (the waiting room space is 1); non-source-aware, that is, new packets can preempt packets from other sources; and if the system is full when a new packet arrives, the oldest packet (the one being served) will be replaced by the newly arrived one. As shown in Fig. 1.

We derive the age of information for TSLS-Q system using SHS. We then generalize the conclusion for two sources to N sources system, **NSLS-Q** system. Also, numerical results are used to compare this model with some other traditional queueing systems from a variety of perspectives. The results show that the queue model performs better in terms of AoI compared to the traditional queue model.

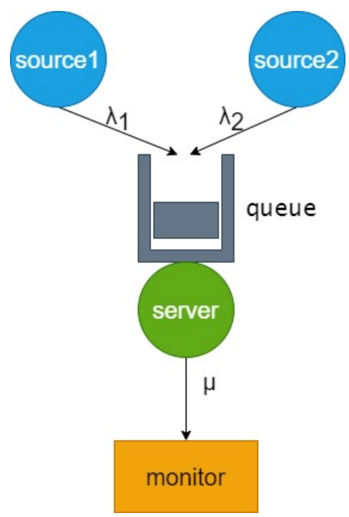

Fig. 1. the two-sources, single buffer and LCFS-S system (TSLS-Q system)

B. Related Work

If packet management is performed in the queue, then it will significantly reduce the AoI. This is what many research groups are working on recently.

Regarding this paper, there are mainly two directions of related work. One direction is to use SHS to analyze age, and the other direction is different queue models and packet management strategies. We therefore introduce related work from these two directions. SHS are first modeled and analyzed in [32]. The authors of [32] describe a model of SHS in which transitions between discrete modes are triggered by random events. The analysis of AoI increasingly made use of SHS [20]–[23] [33]–[38]. [20] proposes a new simple method for computing AoI based on SHS. By applying SHS, [21] first studied systems with one source and n servers and then extended the results to systems with m sources and n servers. In [37], the M/M/c* preemptive parallel server system was analyzed using SHS. The authors of [22] investigate three source-aware packet management strategies. Policy 1 has a queue length of 2 and a preemption method of LCFS-W. Policy 2 has a queue length of 1 and a preemption method of LCFS-W. Policy 3 has a queue length of 1 and a preemption method of LCFS-S. The authors of [23] consider AoI for the case where the queue length is 0. The authors of [39] derive a first order linear differential equation for the moment generating function (MGF) based on the SHS. The authors of [36] solve for the AoI of each node in the LCFS server of a multi-hop network. The authors of [38] correct some of the errors of the previous SHS-based analysis of AoI and assesses them numerically with minimal error.

Although a large number of queueing systems have been proposed, including M/M/1 [3], M/D/1 [3], M/M/1/1 [6], M/M/1/2* [6], PH/PH/1/1 [40] and M/PH/1/2 [40], M/G/1/1 [10], M/M/c* [37] and so on. But we think there are still new queueing systems worth exploring.

C. Organization

The paper is organized as follows. In Sections II, We introduce the basic definition of AoI and some other related concepts, and list some simple models of AoI. The SHS for AoI analysis is introduced in Section III and In Section IV we calculate the AoI of the TSLS-Q system. In Section V we derive our conclusions to the AoI of the NSLS-Q system by using the AoI formula of the TSLS-Q system. In Section VI We have done some numerical experiments. Finally, We conclude the paper and pointe out the direction of future research in Section VII.

II. Preliminaries

In order to better understand this paper, in this section, we introduce some basic definitions related to AoI: state age, age of information, peak age, and change of age with server utilization.

The basic notations used in the system model are listed in TABLE I.

A. Status Age And Average Age

We derive the average state update age for a system whose sources update the state of monitors by sending a steady stream of packets to the queue. We assume that it starts at time $t = 0$. The sending time of each data packet is respectively $t_1, t_2, t_3, ..., t_n$. The arrival time of each data packet is ${t_1}', {t_2}', {t_3}', ..., {t_n}'$ respectively.

For each data packet, we record a timestamp, the symbol is u, indicating the time when the data packet was generated. status age is defined as the elapsed time since the latest data available at the destination was generated by the source, denoted as $\Delta(t) = t - u(t)$ [3]. The status age changes as shown in the Fig. 20. We can see from the figure that the age of information changes as a jagged curve over time. The message age increases linearly when no new packets arrive. When the next packet arrives, the message age is reset to a new, smaller value.

In addition, another concept arose naturally: age of information.over an interval $(0, T)$, the average age (also known as AoI) is

$$\Delta_T = \frac{1}{T}\int_0^T \Delta(t)\,dt. \tag{1}$$

What's more: AoI is for the target node, not in units of packets. Instead, delays are for individual packets. So Fig. 20 is a graph for a certain target node.

When $T \to \propto$, we assume that the limit of Δ_T exists and is not infinite, then the average age will converge to a value. So the average status update age can be obtained as [3].

$$\Delta = \lim_{T \to \propto} \Delta_T = \lambda\left(E[XT] + E\left[X^2/2\right]\right), \tag{2}$$

TABLE I
EXPLANATIONS OF BASIC NOTATIONS

Symbol	Explanation
t_i	The time when the i-th packet was sent
$t_i{}'$	The arrival time of the i-th packet
u	Timestamp of when the packet record was generated
$\Delta(t)$	status age at time t
Δ_T	the average age (also known as AoI)
Δ	The average status update age
X_k	the difference between the sending time of the k-th packet and the sending time of the k-1-th packet, recorded as $X_k = t_k - t_{k-1}$
T_k	the difference between the arrival time of the k-th packet and the sending time, recorded as $T_k = t_k{}' - t_k$
A_k	k-th peak age
Q_k	k-th area
λ	arrival rate of status update packets
λ_i	arrival rate of status update packets of source i
μ	service rate
ρ	load,and the expression is $\rho = \frac{\lambda}{\mu}$
q(t)	a continuous-time finite-state Markov chain that describes the occupancy of the system
x(t)	a continuous process that describes the change of age-related processes at the sink
$v(t)$	the correlation between the continuous variable x(t) and the state q(t)
π_{q_l}	the probability that the Markov chain is in state q_l

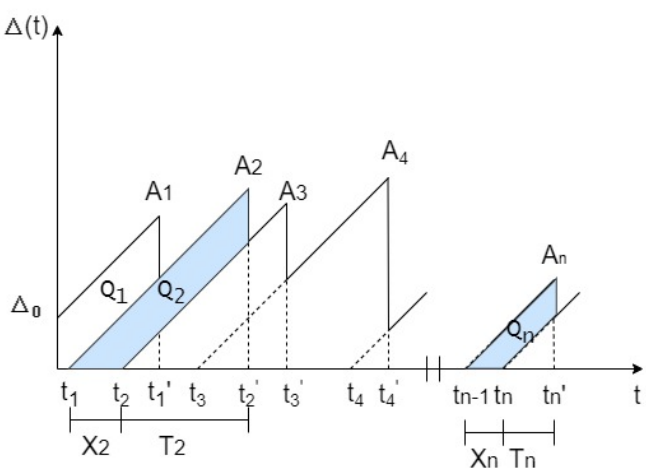

Fig. 2. Age of information

where X_k is the difference between the sending time of the k-th packet and the sending time of the k-1-th packet, and T_k is the difference between the arrival time of the k-th packet and the sending time. The expression is

$$X_k = t_k - t_{k-1}, \tag{3}$$

$$T_k = t_k{}' - t_k, \tag{4}$$

So when can the system obtain the minimum age? For the FCFS systems, the minimum age is achieved when the previous packet is just completed and the next packet is sent just in time.That is $t_i = t'_{i-1}$. In this case, each packet has no waiting time and is served as soon as it arrives.

From [4] [5], We have obtained through experiments that average age Δ first decreases and then increases with the change of load ρ. For example, for the M/M/1 queueing system, when ρ^*=0.53, Δ reaches the minimum value. In the M/M/1 queueing system, when the load ρ is less than ρ^*=0.53, Δ decreases as the load increases. When the load ρ is greater than ρ^*=0.53, Δ increases with the load. We will introduce these changes in detail in section II-C.

B. Peak Age

Due to the complexity of calculating the average age, previous studies usually used the peak age of information (PAoI) instead of the average age. From [41], the age of information is proposed, which is represented by the symbol A_k. Peak age can represent the worst-case scenario for age. And compared with the average age, the peak age does not require complex calculations such as integrals, making the analysis more convenient. For the k-th data packet, its peak age can be obtained by the following formula:

$$A_k = X_k + T_k. \tag{5}$$

We further introduced the average peak age. The average peak age is defined as:

$$\begin{aligned} E\left[A_k\right] &= \frac{1}{k}\sum_{k=1}^{K} A_k \\ &= \frac{1}{k}\sum_{k=1}^{K} \left(X_k + T_k\right) \\ &= E\left[X_k\right] + E\left[T_k\right]. \end{aligned} \tag{6}$$

[41] studied the average of a new metric called peak age. And from [41], We got the average peak age of M/M/1/1 queue and M/M/1/2* model, respectively as follows.

1) The average peak age of the M/M/1/1 queue is as follows

$$E\left[A_k\right] = \frac{1}{\lambda} + \frac{2}{\mu}. \tag{7}$$

2) The average peak age of the M/M/1/2 model is as follows*

$$E\left[A_k\right] = \frac{1}{\lambda} + \frac{2}{\mu} + \frac{1}{\lambda+\mu} + \frac{\lambda^2\mu}{\left(\lambda+\mu\right)^2\left(\lambda^2+\lambda\mu+\mu^2\right)}, \tag{8}$$

where λ is the arrival rate of status update packets, and μ is the service rate. What's more, we propose the load, symbolized as ρ , and the expression is $\rho = \frac{\lambda}{\mu}$.

TABLE II
AGE OF DIFFERENT QUEUEING SYSTEMS

queueing systems	**Δ (The average status update age)**
M/M/1	$\Delta = \frac{1}{\mu}\left(1+\frac{1}{\rho}+\frac{\rho^2}{1-\rho}\right)$
M/D/1	$\Delta = \frac{1}{\mu}\left(\frac{1}{2(1-\rho)}+\frac{1}{2}+\frac{(1-\rho)exp(\rho)}{\rho}\right)$
D/M/1	$\Delta = \frac{1}{\mu}\left(\frac{1}{2\rho}+\frac{1}{1-\beta}\right)$
M/M/1*	$\Delta = \frac{1}{\mu}\left(1+\frac{1}{\rho}\right)$
M/M/1/1	$\Delta = \frac{1}{\lambda}+\frac{2}{\mu}-\frac{1}{\lambda+\mu}$
M/M/1/2	$\Delta = \frac{1}{\lambda}+\frac{3}{\mu}-\frac{2(\lambda+\mu)}{\lambda^2+\lambda\mu+\mu^2}$
M/M/1/2*	$\Delta = \frac{1}{\lambda}+\frac{2}{\mu}+\frac{\lambda}{(\lambda+\mu)^2}+\frac{1}{\lambda+\mu}-\frac{2(\lambda+\mu)}{\lambda^2+\lambda\mu+\mu^2}$
M/D/1*	$\Delta = \frac{1}{\mu}\frac{exp(\rho)}{\rho}$
D/M/1*	$\Delta = \frac{1}{\mu}\left(1+\frac{1}{2\rho}\right)$
M/M/c*	$\Delta = \frac{1}{\mu}\left[\frac{1}{c}\prod_{i=1}^{c-1}\frac{\rho}{i+\rho}+\frac{1}{\rho}\right]$ $+\frac{1}{\mu}\left[\frac{1}{\rho}\sum_{l=1}^{c-1}\prod_{i=1}^{l}\frac{\rho}{i+\rho}\right]$
...	...

C. Change Of Age With Server Utilization

After the concept of age of information was introduced, numerous studies started to explore the variation of AoI for different queuing systems. From [3], the AoI of M/M/1, M/D/1 and D/M/1 were investigated. [4] [5] verified the conclusion of [3] by experiments. AoI of M/M/1* was analysed in [6], where * represents allowing new arriving packets to preempt updates in the service. [6] further derives formulas for AoI of M/M/1/1, M/M/1/2, and M/M/1/2*, and also proposes a new metric called peak age. For some more complex models, [40] built models to study PH/PH/1/1 and M/PH/1/2. In [9], we see the AoI of M/D/1* and D/M/1*. In [37], the M/M/c* preemptive parallel server system was analyzed using SHS.

1) Single Source and Single Server

For a system that has a source updating a monitor through a first-come-first-served single server packet queue, The ages of several different queueing systems are shown in the TABLE II.

In TABLE II, we list the formulas for AoI for the different queueing systems. We can then derive these formulas. The point where the derivative is equal to 0 is the point at which AoI obtains its minimum value. For example, for M/M/1 queueing system, the solution is $\rho^* = 0.53$. For D/M/1 queueing system, $\rho^* = 0.515$.

From [4] [5], we experimentally conclude that AoI decreases and then rises as the change in load ρ increases. For example, for the M/M/1 queueing system, the AoI reaches a minimum at $\rho^* = 0.53$. In the M/M/1 queueing system when the load ρ is less than $\rho^* = 0.53$, the AoI decreases as the load increases. At loads greater than $\rho^* = 0.53$, AoI increases with increasing load. This is not difficult to explain. Because the queue is usually empty before ρ^*, arriving packets can be served immediately. As the load increases, packets are sent at a faster rate and updated more quickly, helping to reduce message age. With queued packets in the queue after ρ^*, arriving packets are not served immediately and need to be queued, causing the information age to increase due to waiting in the queue.

2) Single Source and Multiple Servers

In [21],The average age of information at the monitor for single- source n-server network where each server has a LCFS queue is:

$$\Delta = \frac{1}{\mu}\left[\frac{1}{n\rho}\sum_{j=1}^{n-1}\prod_{i=1}^{j}\frac{\rho\left(n-i+1\right)}{i+\left(n-i\right)\rho}+\frac{1}{n\rho}+\frac{1}{n^2}\prod_{i=1}^{n-1}\frac{\rho\left(n-i+1\right)}{i+\left(n-i\right)\rho}\right]. \tag{9}$$

3) Multiple Sources and Single Server

After studying single source systems, numerous studies [20]–[27] began to study multi-source systems. We build a system with N sources and a monitor, which using M/M/1 queue systems. Each source i offers update packets as a rate λ_i. The service rate is μ for all packets. So the load for each source is: $\rho_i = \frac{\lambda_i}{\mu}$. The total load is:

$$\rho = \sum_{i=1}^{N}\rho_i = \sum_{i=1}^{N}\frac{\lambda_i}{\mu}. \tag{10}$$

[20] [22] [24] studied the age of information of different services in the system of N sources and one monitor. These different services and the corresponding age of information of the i-th source are as follows:

FCFS: first come first served.

$$\Delta_i = \frac{1}{\mu}\left[\frac{\rho_i^2\left(1-\rho\rho_{-i}\right)}{\left(1-\rho\right)\left(1-\rho_{-i}\right)^3}+\frac{1}{1-\rho_{-i}}+\frac{1}{\rho_i}\right]. \tag{11}$$

LCFS-S: last come first served, and a new update packet preempts any update packet currently in service. (it permit preemption in both service and waiting)

$$\Delta_i = \frac{1}{\mu}\left(1+\rho\right)\frac{1}{\rho_i}. \tag{12}$$

LCFS-W: last come first served, and a new packet replaces any older packet waiting in the queue (but it does not permit preemption in service)

$$\Delta_i = \frac{1}{\mu}\left[\alpha_W\left(\rho\right)+\left(1+\frac{\rho^2}{1+\rho}\right)\frac{1}{\rho_i}\right], \tag{13}$$

where

$$\alpha_W\left(\rho\right) = \frac{\left(1+\rho+\rho^2\right)^2+2\rho^3}{\left(1+\rho+\rho^2\right)\left(1+\rho\right)^2}. \tag{14}$$

III. Introduction to Stochastic Hybrid Systems

We use the Stochastic Hybrid Systems (SHS) [32] [33] [34] for AoI analysis in this paper, so in this section we give a brief introduction to the system. We refer to [20] [21] and apply the method first proposed in [20] to calculate AoI.

A SHS is defined by a stochastic differential equation (SDE) in [32].

$$\dot{x} = f(q, x, t) + g(q, x, t)\dot{n},$$

$$f : Q \times \mathbb{R}^n \times [0, \propto) \to \mathbb{R}^n$$
$$g : Q \times \mathbb{R}^n \times [0, \propto) \to \mathbb{R}^{n\times k}$$

where $q : [0, \propto) \to Q$ represents the discrete state of a sequence of jumping processes, $x : [0, \propto) \to \mathbb{R}^n$ represents the continuous state of the change of age of information. In this paper, R represents the set of real numbers, and N represents the set of non-negative integers.

The SHS consists of discrete states q_t and continuous states x_t, denoted as state pairs (q_t, x_t). We model the queuing system with the state pair (q_t, x_t). More details about the SHS can be seen in [20] [21].

x(t) is a continuous variable representing the time at which the state of a different monitor or queue is updated at moment t. x(t) describes the continuous evolution of age of information for a series of age-related processes. Continuous-time continuous state $x(t) = \begin{bmatrix} x_0(t) & x_1(t) & \dots & x_n(t) \end{bmatrix}$ is the stochastic process for AoI. With this definition, we have $x_0 \leq x_1 \leq \dots \leq x_n$, for any time. When there are no state jumps, all x(t) grows linearly at a unit rate. q(t) is a discrete quantity, representing different states. q(t) is a continuous-time finite-state Markov chain that describes the occupancy of a service facility. The set of all states is noted as Q, and $Q = \begin{bmatrix} q_0 & q_1 & \dots & q_n \end{bmatrix}$. Looking at the graph, for example in Fig.3, q represents each node.

A transition will be generated when a packet arrives or leaves. That is, a change between different states q. We will denote the transition as $\lambda^{(l)}$. For a system with two sources, the range of values of $\lambda^{(l)}$ is: $\lambda^{(l)} \in \{\lambda_1, \lambda_2, \mu\}$, where λ_1 and λ_2 are the arrival rates of status update packets for source 1 and source 2 respectively, μ is the service rate for all packets. The arrival times are exponentially distributed according to the M/M/1 model. We record the state transitions of the Markov chain as a triplet: $\left(q_1, q_2, \lambda^{(l)}\right)$. In addition, self-to-self transitions are possible. For each state q, $L_{\bar{q}}$ and $L'_{\bar{q}}$ are the respective sets of incoming and outgoing transitions. $L_{\bar{q}}$ denotes the set of all possible states that can leap to state q. $L'_{\bar{q}}$ denotes the set of possible states to which state q can leap. As shown in the equation below:

$$L_{\bar{q}} = \{l \in L : q_l = q\}, q \in Q.$$
$$L'_{\bar{q}} = \{l \in L : q'_l = q\}, q \in Q.$$

For each transition l, there is a mapping of transitions that makes x(t) produce a jump. When the state is transitioned from q to q', x becomes x'. The transition maps are $A_l \in \{0, 1\}^{(n+1)*(n+1)}$.

$$\boldsymbol{x'} = \boldsymbol{x A_l}. \quad (15)$$

What's more, the continuous state x evolves into a piecewise linear function via a differential equation $\dot{\boldsymbol{x}} = \frac{\partial x(t)}{\partial t} = \boldsymbol{b_q}$, where $\boldsymbol{b_q} = \begin{bmatrix} b_{q0} & b_{q1} & \dots & b_{qn} \end{bmatrix}$, $\bar{q} \in Q$. If the age process xj(t) increases at a unit rate, we have $b_{qj} = 1$; otherwise , $b_{qj} = 0$.

We then introduced $\boldsymbol{v(t)} = \begin{bmatrix} v_0(t) & v_1(t) & \dots & v_n(t) \end{bmatrix}$ to represent the correlation between the continuous variable x(t) and the state q(t).

$$\boldsymbol{v_q(t)} = \begin{bmatrix} v_{q0}(t) & v_{q1}(t) & \dots & v_{qn(t)} \end{bmatrix}, \qquad \bar{q} \in Q. \quad (16)$$

π_{q_l} denotes the probability that the Markov chain is in state q_l. So the probability vector of the state is $\boldsymbol{\pi(t)} = \begin{bmatrix} \pi_0(t) & \pi_1(t) & \dots & \pi_n(t) \end{bmatrix}$. And the probability sum of all π is 1.

$$\pi_{q_l}(t) = P[q(t) = q_l].$$
$$\sum_{q \in Q} \pi_q = 1. \quad (17)$$

Lemma 1: From [20]. Our analysis of age of information assumes that Markov chains are traversable. Otherwise, analyzing the average age of information is pointless. Under this assumption, for a piecewise linear SHS with linear reset maps, $\boldsymbol{\pi(t)} = \begin{bmatrix} \pi_0(t) & \pi_1(t) & \dots & \pi_n(t) \end{bmatrix}$ always converges to a unique stationary probability vector $\bar{\boldsymbol{\pi}} = \begin{bmatrix} \bar{\pi}_{\bar{0}} & \bar{\pi}_{\bar{1}} & \dots & \bar{\pi}_{\bar{n}} \end{bmatrix}$. $\bar{\boldsymbol{\pi}}$ satisfies:

$$\sum_{q \in Q} \bar{\pi}_{\bar{q}} = 1,$$
$$\bar{\pi}_{\bar{q}} \sum_{l \in L_{\bar{q}}} \lambda^{(l)} = \sum_{l \in L'_{\bar{q}}} \lambda^{(l)} \bar{\pi}_{\bar{q}l}, \qquad \bar{q} \in Q. \quad (18)$$

Lemma 2: From [[20], Theorem 4]. If the discrete-state Markov chain $q(t)$ is ergodic with stationary distribution π and we can find a non-negative solution $\bar{v} = \begin{bmatrix} \bar{v}_0 & \dots & \bar{v}_m \end{bmatrix}$ such that

$$\bar{\boldsymbol{v}}_{\bar{\boldsymbol{q}}} \sum_{l \in L_{\bar{q}}} \lambda^{(l)} = \boldsymbol{b_{\bar{q}}} \bar{\pi}_{\bar{q}} + \sum_{l \in L'_{\bar{q}}} \lambda^{(l)} \bar{\boldsymbol{v}}_{\boldsymbol{ql}} \boldsymbol{A_l}, \qquad \bar{q} \in Q, \quad (19)$$

then the average age of the AoI SHS is given by

$$\Delta = \sum_{q \in Q} \bar{v}_{q0} \quad (20)$$

Eq.(20) is the formula for solving the age of information, so we only need to obtain the $\bar{\boldsymbol{v}}_{\bar{\boldsymbol{q}}}$ corresponding to different q to obtain the age of information. And we can solve for $\bar{\boldsymbol{v}}_{\bar{\boldsymbol{q}}}$ by Eq.(19).

IV. AoI Analysis of the TSLS-Q System

In this section, we use the SHS technique to calculate the age of information for TSLS-Q System. According to Eq.(20) introduced in section III, we only need to calculate $\bar{v}_{q0}, \forall q \in Q$. We can calculate $\bar{v}_{q0}, \forall q \in Q$ by using Eq.(19). So we need to first calculate the π and $\bar{v}_{ql} A_l$ needed in Eq.(19).

This section will be divided into three subsections, A: we will perform the calculation of π_q and b_q; B: we will perform the derivation of $\bar{v}_{ql} A_l$; C: we will perform the calculation of AoI based on the results of the first two sections.

q(t) is a continuous-time finite-state Markov chain that describes the occupancy of a service facility. In Strategy 1, our state update system has a buffer queue length of 1. The

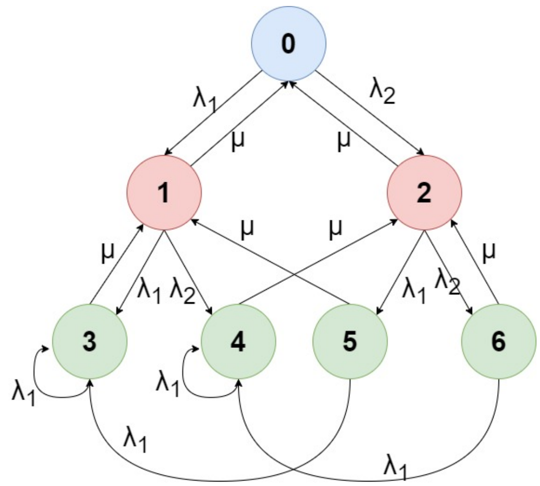

Fig. 3. The SHS Markov chain for TSLS-Q System

TABLE III
THE STATE SPACE OF THE MARKOV CHAIN

State	Source index of the packet in the server	Source index of the packet in the queue	Server
0	-	-	I
1	1	-	B
2	2	-	B
3	1	1	B
4	1	2	B
5	2	1	B
6	2	2	B

state space of the Markov chain is $Q = \{0, 1, 2, 3, 4, 5, 6\}$. The meaning of each state is shown in the TABLE III. The SHS Markov chain for updates of source 1 is shown in the Fig.3.

The meanings of the various states of the Markov chain are as follows. $q = 0$ means that the system is idle (denoted by I). $q = 1$ means that the server is busy (indicated by B) and the queue is empty, and the packet in service is from source 1. $q = 2$ means the server is busy but the queue is empty, and the packet in service is from source 2. $q = 3$, $q = 4$, $q = 5$, $q = 6$, all indicate that the server is busy and the queue is busy. $q = 3$, indicates that the data packet in the service and the data packet in the queue all come from source 1. $q = 6$, indicates that the data packet in the service and the data packet in the queue all come from source 2. $q = 4$, indicates that the data packet in the service comes from source 1, and the data packet in the queue comes from source 2. $q = 5$, indicates that the data packet in the service comes from source 2, and the data packet in the queue comes from source 1. We present the above in TABLE III.

$\boldsymbol{x}(\boldsymbol{t}) = \begin{bmatrix} x_0(t) & x_1(t) & \dots & x_n(t) \end{bmatrix}$ describes the continuous evolution of age of information for a series of age-related processes. $x(t)$ enables tracking of the age of source 1 updates at the monitor. Our continuous state vector is $\boldsymbol{x}(\boldsymbol{t}) = \begin{bmatrix} x_0(t) & x_1(t) & x_2(t) \end{bmatrix}$ is the current age at the monitor of source 1 at time instant t, namely $\Delta_1(t)$; $x_1(t)$ is what $\Delta_1(t)$ will become if the in-service packet is delivered to the monitor at time t; $x_2(t)$ is what $\Delta_1(t)$ will become if the first packet in the queue is delivered to the receiver at time t.

Then, how does $\boldsymbol{x}(\boldsymbol{t}) = \begin{bmatrix} x_0(t) & x_1(t) & x_2(t) \end{bmatrix}$ change? When no new data packet arrives and no new data packet leaves, that is, when the state Q does not change, $x_0(t)$, $x_1(t)$, and $x_2(t)$ all increase linearly with time at a unit rate. We only target source 1 for research, and the same goes for source 2. Therefore, in the following, when we refer to age of information, we refer to the age of information of source 1 unless otherwise specified. For source 1, when a new packet from source 1 arrives at the monitor, $x_0(t)$ is reset to a new smaller value related to the newly arrived packet at the monitor, $x_0(t) = x_1(t)$; When a new packet from source 1 enters the server, $x_1(t)$ is reset to 0, $x_1(t) = 0$; when a new packet from source 1 enters the queue, $x_2(t)$ is reset to 0, $x_2(t) = 0$. At the same time, for source 1, the arrival of data packets from source 2 to the monitor will not affect the age of information of source 1.

In the following, we will start calculating the AoI.

A. *Calculate π_q and b_q*

As long as the state q(t) does not change, x increases at a unit rate. Therefore $b_{\bar{q}} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$.

We first introduce two matrices: D and Q. Since there are seven states in the system, both D and Q are 7-row and 7-column matrices. In the matrix D and Q, the i-th row of Q represents the i-th state, and the i-th column also represents the i-th state. D is a diagonal matrix, and the value of the diagonal of row i in D is $\sum_{l \in L_{\bar{i}}} \lambda^{(l)}$. In the matrix Q, The value in column j of row i represents the transition from state q_i to q_j. For example, the value λ_1 in row 0, column 1 represents the transition from state 0 to state 1 through λ_1. The matrices D and Q are shown as follows.

$$D = \begin{bmatrix} \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda+\mu & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda+\mu & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_1+\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_1+\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_1+\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1+\mu \end{bmatrix} \tag{21}$$

$$Q = \begin{bmatrix} 0 & \lambda_1 & \lambda_2 & 0 & 0 & 0 & 0 \\ \mu & 0 & 0 & \lambda_1 & \lambda_2 & 0 & 0 \\ \mu & 0 & 0 & 0 & 0 & \lambda_1 & \lambda_2 \\ 0 & \mu & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & \lambda_1 & 0 & 0 \\ 0 & \mu & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & \mu & 0 & \lambda_1 & 0 & 0 \end{bmatrix} \tag{22}$$

There are seven states in this system, so probability vector is $\bar{\pi} = \begin{bmatrix} \pi_0 & \pi_1 & \pi_2 & \pi_3 & \pi_4 & \pi_5 & \pi_6 \end{bmatrix}$,

We introduced in Chapter III the Eq.(17) and Eq.(18). Using Eq.(18), we can conclude that $\bar{\pi}$ satisfies $\bar{\pi}D = \bar{\pi}Q$, And the probability sum of all π is 1, $\sum_{i=0}^{6} \pi_i = 1$, Thus, we obtain the following system of equations:

$$\begin{cases} \sum_{i=0}^{6} \pi_i = 1 \\ \bar{\pi} D = \bar{\pi} Q \end{cases} \tag{23}$$

Substituting Eq.(21) and Eq.(22) into Eq.(23), we get the following system of equations:

$$\begin{cases} \pi_0 + \pi_1 + \pi_2 + \pi_3 + \pi_4 + \pi_5 + \pi_6 = 1 \\ \mu(\pi_1 + \pi_2) = \lambda \pi_0 \\ \lambda_1 + \mu\pi_3 + \mu\pi_5 = \pi_1(\lambda + \mu) \\ \lambda_2 + \mu\pi_4 + \mu\pi_6 = \pi_2(\lambda + \mu) \\ \lambda_1(\pi_1 + \pi_3 + \pi_5) = \pi_3(\lambda_1 + \mu) \\ \lambda_2\pi_1 + \lambda_1(\pi_4 + \pi_6) = \pi_4(\lambda_1 + \mu) \\ \lambda_1\pi_2 = \pi_5(\lambda_1 + \mu) \\ \lambda_2\pi_2 = \pi_6(\lambda_1 + \mu) \end{cases} \tag{24}$$

By solving the above system of equations, we derive $\bar{\pi}$. The value of $\bar{\pi}$ is as follows:

$$\bar{\pi} = \frac{1}{1+\rho+\rho^2}\kappa, \tag{25}$$

where

$$\kappa = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \rho_1^2 \frac{\rho+1}{\rho_1+1} & \rho_1\rho_2 \frac{\rho+1}{\rho_1+1} & \frac{\rho_1\rho_2}{\rho_1+1} & \frac{\rho_2^2}{\rho_1+1} \end{bmatrix}. \tag{26}$$

B. Calculate $\bar{v}_{ql} A_l$

We record the state transitions of the Markov chain as a triplet: (q_1, q_2, ω), where q_1 and q_2 represent two different states of the Markov chain in the Fig.3, $q_1 \in \{0,1,2,3,4,5,6\}$ $q_2 \in \{0,1,2,3,4,5,6\}$; ω represents the arrival or departure rate of packets from various sources, $\omega \in \{\lambda_1, \lambda_1, \mu\}$. For example, $(0,1,\lambda_1)$ indicates that a packet from source 1 enters the server at rate λ_1, and jumps from the state 0 to the state 1.

We summarise transitions for the Markov chain in TABLE IV. The specific derivations of TABLE IV are shown as follows:

1) $l = 1$, $(0,1,\lambda_1)$:

The original state of the system is 0. At this point packet from source 1 arrives at the empty system and the state jumps to 1. Since no packets are leaving, $x_0{}' = x_0$. Newly arrived packet enters the server, so $x_1{}' = 0$. The queue is still empty, so $x_2{}' = x_2$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & 0 & x_2 \end{bmatrix}$.

2) $l = 2$, $(0,2,\lambda_2)$:

This transition is similar to $l = 1$, so $x_0{}' = x_0$ and $x_2{}' = x_2$. But the difference is that the packet arriving is from source 2. The newly arrived packet enters the server, but the packet is from source 2 and has nothing to do with the AoI from source 1, so $x_1{}' = x_0$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_0 & x_2 \end{bmatrix}$.

3) $l = 3$, $(1,0,\mu)$:

The original state before the transition is 1. At this point the packet from source 1 leaves the system and the state leaps to 0. Since the packet from source 1 in the server leaves, $x_0{}' = x_1$. The other AoI do not change, so $x_1{}' = x_1$ and $x_2{}' = x_2$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_1 & x_1 & x_2 \end{bmatrix}$.

4) $l = 4$, $(2,0,\mu)$:

At this point the packet from source 2 leaves the system and the state jumps to 0. Since the packet from source 2 in the server leaves, this process does not affect the AoI of source 1 at all. So $x_0{}' = x_1$, $x_1{}' = x_1$ and $x_2{}' = x_2$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}$.

5) $l = 5$, $(1,3,\lambda_1)$:

The original state before the transition is 1, at which point packet from source 1 arrives on the system. The server is not empty and the queue is empty, so newly arrived packet from source 1 enters the queue, so $x_2{}'$ is reset to 0, $x_2{}' = 0$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_1 & 0 \end{bmatrix}$.

6) $l = 6$, $(1,4,\lambda_2)$:

The server is not empty and the queue is empty, so newly arrived packet from source 2 enters the queue. Packets from source 2 do not change the AoI of source 1. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_1 & x_1 \end{bmatrix}$.

7) $l = 7$, $(3,1,\mu)$:

The original state before the transition is 3. At this point the packets in both the server and the queue come from source 1. Packet in the server leaves the system and packet in the queue enters the server. So we get $\boldsymbol{x'} = \begin{bmatrix} x_1 & x_2 & x_2 \end{bmatrix}$.

8) $l = 8$, $(4,2,\mu)$:

At this point the packet from source 1 in the server leaves the system and the packet from source 2 in the queue goes to the server. So we get $\boldsymbol{x'} = \begin{bmatrix} x_1 & x_1 & x_2 \end{bmatrix}$.

9) $l = 9$, $(3,3,\lambda_1)$:

The original state before the transition is 3, at which point the packet from source 1 arrives at the system. Since the system is already full, the newly arrived packet from source 1 preempts the packet from source 1 in the service. So $x_1{}'$ is reset to 0, $x_1{}' = 0$. The packet in the queue is unaffected, so $x_2{}' = x_2$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & 0 & x_2 \end{bmatrix}$.

10) $l = 10$, $(4,4,\lambda_1)$:

This transition is similar to $l = 9$. The newly arrived packet from source 1 preempts the packet from source 1 in the service, so $x_1{}'$ is reset to 0. The queue is for packet from source 2, which do not affect the AoI of source 1, so $x_2{}' = x_1{}' = 0$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & 0 & 0 \end{bmatrix}$.

11) $l = 11$, $(2,5,\lambda_1)$:

The original state before the transition is 2, at which point packet from source 1 arrives on the system. The server is not empty and the queue is empty, so newly arrived packet from source 1 enters the queue, so $x_2{}'$ is reset to 0, $x_2{}' = 0$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_0 & 0 \end{bmatrix}$.

TABLE IV
TABLE OF TRANSITIONS FOR THE CHAIN IN FIG.3.

l	$q_1 \to q_2$	λ	(q_1, q_2, ω)	xA_l	A_l	$\bar{v}_{ql}A_l$
1	$0 \to 1$	λ_1	$(0, 1, \lambda_1)$	$\begin{bmatrix} x_0 & 0 & x_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{00} & 0 & \bar{v}_{02} \end{bmatrix}$
2	$0 \to 2$	λ_2	$(0, 2, \lambda_1)$	$\begin{bmatrix} x_0 & x_0 & x_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{00} & \bar{v}_{00} & \bar{v}_{02} \end{bmatrix}$
3	$1 \to 0$	μ	$(1, 0, \mu)$	$\begin{bmatrix} x_1 & x_1 & x_2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{11} & \bar{v}_{11} & \bar{v}_{12} \end{bmatrix}$
4	$2 \to 0$	μ	$(2, 0, \mu)$	$\begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{20} & \bar{v}_{21} & \bar{v}_{22} \end{bmatrix}$
5	$1 \to 3$	λ_1	$(1, 3, \lambda_1)$	$\begin{bmatrix} x_0 & x_1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{10} & \bar{v}_{11} & 0 \end{bmatrix}$
6	$1 \to 4$	λ_2	$(1, 4, \lambda_2)$	$\begin{bmatrix} x_0 & x_1 & x_1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{10} & \bar{v}_{11} & \bar{v}_{11} \end{bmatrix}$
7	$3 \to 1$	μ	$(3, 1, \mu)$	$\begin{bmatrix} x_1 & x_2 & x_2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{31} & \bar{v}_{32} & \bar{v}_{32} \end{bmatrix}$
8	$4 \to 2$	μ	$(4, 2, \mu)$	$\begin{bmatrix} x_1 & x_1 & x_2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{41} & \bar{v}_{41} & \bar{v}_{42} \end{bmatrix}$
9	$3 \to 3$	λ_1	$(3, 3, \lambda_1)$	$\begin{bmatrix} x_0 & 0 & x_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{30} & 0 & \bar{v}_{32} \end{bmatrix}$
10	$4 \to 4$	λ_1	$(4, 4, \lambda_1)$	$\begin{bmatrix} x_0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{40} & 0 & 0 \end{bmatrix}$
11	$2 \to 5$	λ_1	$(2, 5, \lambda_1)$	$\begin{bmatrix} x_0 & x_0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{20} & \bar{v}_{20} & 0 \end{bmatrix}$
12	$2 \to 6$	λ_2	$(2, 6, \lambda_2)$	$\begin{bmatrix} x_0 & x_0 & x_0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{20} & \bar{v}_{20} & \bar{v}_{20} \end{bmatrix}$
13	$5 \to 1$	μ	$(5, 1, \mu)$	$\begin{bmatrix} x_0 & x_2 & x_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{50} & \bar{v}_{52} & \bar{v}_{52} \end{bmatrix}$
14	$6 \to 2$	μ	$(6, 2, \mu)$	$\begin{bmatrix} x_0 & x_0 & x_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{60} & \bar{v}_{60} & \bar{v}_{62} \end{bmatrix}$
15	$5 \to 3$	λ_1	$(5, 3, \lambda_1)$	$\begin{bmatrix} x_0 & 0 & x_2 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{50} & 0 & \bar{v}_{52} \end{bmatrix}$
16	$6 \to 4$	λ_1	$(6, 4, \lambda_1)$	$\begin{bmatrix} x_0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} \bar{v}_{60} & 0 & 0 \end{bmatrix}$

12) $l = 12$, $(2, 6, \lambda_2)$:

The packet from source 2 arrives at the system at this point. After the arrival of a new packet from source 2, the packets in the server and queue come from source 2, so ${x_2}' = {x_1}' = {x_0}' = x_0$. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_0 & x_0 \end{bmatrix}$.

13) $l = 13$, $(5, 1, \mu)$:

The packet from source 2 in the server leaves the system and the packet from source 1 in the queue goes to the server. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_2 & x_2 \end{bmatrix}$.

14) $l = 14$, $(6, 2, \mu)$:

The packet from source 2 in the server leaves the system and the packet from source 2 in the queue goes to the server. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_0 & x_2 \end{bmatrix}$.

15) $l = 15$, $(5, 3, \lambda_1)$:

This transition is similar to $l = 9$, newly arrived packet from source 1 preempts the packet from source 2 in the server. ${x_1}'$ is reset to 0. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & 0 & x_2 \end{bmatrix}$.

16) $l = 16$, $(6, 4, \lambda_1)$ *:*

This transition is similar to $l = 10$, newly arrived packet from source 1 preempts the packet from source 2 in the server. So we get $\boldsymbol{x'} = \begin{bmatrix} x_0 & 0 & 0 \end{bmatrix}$.

Above for each of the sixteen transitions, we find the corresponding $\boldsymbol{x'}$. And from Eq.(15), we know $\boldsymbol{x'} = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix} \boldsymbol{A_l}$, so we can derive $\boldsymbol{A_l}$. We can then solve for $\bar{v}_{ql} A_l$ by using $\bar{v}_{ql} A_l = \begin{bmatrix} \bar{v}_{q0} & \bar{v}_{q1} & \bar{v}_{q2} \end{bmatrix} \boldsymbol{A_l}$. Thus for each of the sixteen transitions, we again get the corresponding $\bar{v}_{ql} A_l$.

For example when $l = 1$, substituting $\boldsymbol{x'} = \begin{bmatrix} x_0 & 0 & x_2 \end{bmatrix}$ into Eq.(15), we get the value of $\boldsymbol{A_1}$:

$$\boldsymbol{A_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{27}$$

so $\boldsymbol{\bar{v}_0 A_1} = \begin{bmatrix} \bar{v}_{00} & \bar{v}_{01} & \bar{v}_{02} \end{bmatrix} \boldsymbol{A_1} = \begin{bmatrix} \bar{v}_{00} & 0 & \bar{v}_{02} \end{bmatrix}$.

C. Calculate AoI

In order to calculate AoI, we first calculate $\bar{v}_{ql} A_l$ by Eq.(19). In the above, we have already found π_q, b_q and $\bar{v}_{ql} A_l$. We substitute these variables into the Eq.(19), so we arrive at the system of equations: Eq.(28)

Using Eq.(28), we find the values of $\bar{v}_{00}$, $\bar{v}_{10}$, $\bar{v}_{20}$, $\bar{v}_{30}$, $\bar{v}_{40}$, $\bar{v}_{50}$ and $\bar{v}_{60}$. Their values are as follows:

$$\bar{v}_{00} = \pi_0 \frac{\rho_1^3 + \rho_1^2\rho_2 + 3\rho_1^2 + 4\rho_1\rho_2 + 3\rho_1 + \rho_2^2 + \rho_2 + 1}{\mu\rho_1 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \tag{29}$$

where,

$$\pi_0 = \frac{1}{1 + \rho + \rho^2}.$$

$$\bar{v}_{10} = \pi_0 \frac{4\rho_1^2 + 4\rho_1\rho_2 + 3\rho_1 + \rho_2^2 + \rho_2 + 1}{\mu \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)}. \tag{30}$$

$$\bar{v}_{20} = \pi_0 \rho_2 \frac{4\rho_1^2 + 4\rho_1\rho_2 + 3\rho_1 + \rho_2^2 + \rho_2 + 1}{\mu\rho_1 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)}. \tag{31}$$

$$\begin{aligned} \bar{v}_{30} = {} & \pi_0 \rho_2 \frac{\rho_1^5 + 3\rho_1^4\rho_2 + 8\rho_1^4 + 3\rho_1^3\rho_2^2 + 17\rho_1^3\rho_2 + 17\rho_1^3}{\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \\ & + \pi_0 \rho_2 \frac{\rho_1^2\rho_2^3 + 12\rho_1^2\rho_2^2 + 25\rho_1^2\rho_2 + 15\rho_1^2 + 3\rho_1\rho_2^3}{\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \\ & + \pi_0 \rho_2 \frac{10\rho_1\rho_2^2 + 13\rho_1\rho_2 + 6\rho_1 + \rho_2^3 + 2\rho_2^2 + \rho_2 + 1}{\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)}. \end{aligned} \tag{32}$$

$$\begin{aligned} \bar{v}_{40} = {} & \pi_0 \rho_2^2 \frac{\rho_1^5 + 3\rho_1^4\rho_2 + 8\rho_1^4 + 3\rho_1^3\rho_2^2 + 17\rho_1^3\rho_2 + 17\rho_1^3}{\rho_1\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \\ & + \pi_0 \rho_2^2 \frac{\rho_1^2\rho_2^3 + 12\rho_1^2\rho_2^2 + 25\rho_1^2\rho_2 + 15\rho_1^2 + 3\rho_1\rho_2^3}{\rho_1\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \\ & + \pi_0 \rho_2^2 \frac{10\rho_1\rho_2^2 + 13\rho_1\rho_2 + 6\rho_1 + \rho_2^3 + 2\rho_2^2 + \rho_2 + 1}{\rho_1\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)}. \end{aligned} \tag{33}$$

$$\begin{aligned} \bar{v}_{50} = {} & \pi_0 \rho_2 \frac{5\rho_1^3 + 6\rho_1^2\rho_2 + 9\rho_1^2 + 2\rho_1\rho_2^2}{\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \\ & + \pi_0 \rho_2 \frac{6\rho_1\rho_2 + 5\rho_1 + \rho_2^2 + \rho_2 + 1}{\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)}. \end{aligned} \tag{34}$$

$$\begin{aligned} \bar{v}_{60} = {} & \pi_0 \rho_2^2 \frac{5\rho_1^3 + 6\rho_1^2\rho_2 + 9\rho_1^2 + 2\rho_1\rho_2^2}{\rho_1\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \\ & + \pi_0 \rho_2^2 \frac{6\rho_1\rho_2 + 5\rho_1 + \rho_2^2 + \rho_2 + 1}{\rho_1\mu \left(\rho_1 + 1\right)^2 \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)}. \end{aligned} \tag{35}$$

After finding the values of $\bar{v}_{00}$, $\bar{v}_{01}$, $\bar{v}_{02}$, $\bar{v}_{03}$, $\bar{v}_{04}$, $\bar{v}_{05}$ and $\bar{v}_{06}$, we substitute them into the Eq.(20) to find the value of AoI without difficulty. The values of AoI for source 1 are as follows:

$$\begin{aligned} \Delta_1 = {} & \sum_{q \in Q} \bar{v}_{q0} \\ = {} & \bar{v}_{00} + \bar{v}_{10} + \bar{v}_{20} + \bar{v}_{30} + \bar{v}_{40} + \bar{v}_{50} + \bar{v}_{60} \\ = {} & \pi_0 \frac{\rho_1^5 + 4\rho_1^4\rho_2 + 6\rho_1^4 + 6\rho_1^3\rho_2^2 + 17\rho_1^3\rho_2 + 9\rho_1^3}{\rho_1\mu \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \\ & + \pi_0 \frac{4\rho_1^2\rho_2^3 + 17\rho_1^2\rho_2^2 + 18\rho_1^2\rho_2 + 7\rho_1^2 + \rho_1\rho_2^4 + 7\rho_1\rho_2^3}{\rho_1\mu \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)} \\ & + \pi_0 \frac{11\rho_1\rho_2^2 + 10\rho_1\rho_2 + 4\rho_1 + \rho_2^4 + 2\rho_2^3 + 3\rho_2^2 + 2\rho_2 + 1}{\rho_1\mu \left(\rho_1^2 + 2\rho_1\rho_2 + 2\rho_1 + \rho_2^2 + \rho_2 + 1\right)}. \end{aligned} \tag{36}$$

The above equation is the AoI of source 1 in this system. Similarly, if we want to find the AoI of source 2, we simply interchange ρ_1 and ρ_1 in the above equation.At the same time, the following regularities can be observed: $\bar{v}_{20} = \frac{\rho_2}{\rho_1}\bar{v}_{10}$, $\bar{v}_{40} = \frac{\rho_2}{\rho_1}\bar{v}_{30}$, $\bar{v}_{60} = \frac{\rho_2}{\rho_1}\bar{v}_{50}$. This is also not difficult to explain. In Fig.3, nodes 1 and 2, nodes 3 and 4, and nodes 5 and 6, are symmetrical.

V. AoI Analysis of the NSLS-Q System

In the section above, we solved for AoI of the TSLS-Q system, which is two sources. We can extend this to N sources. In this section, we discuss the AoI formulation of the NSLS-Q system. The system is shown in Fig. 4. The total load of an N-source system is:

$$\rho = \sum_{i=1}^{N} \rho_i. \tag{37}$$

The updates of source i compete for the server against the aggregate other-source updating load. The load of the competing source of source i is:

$$\rho_{-i} = \rho - \rho_i. \tag{38}$$

Thus, we can consider a system of N sources as a two-source system. The load of source 1 in the TSLS-Q system is the load of source i, and the load of source 2 in the TSLS-Q system is the sum of the loads of the other sources. That is, $\rho_1 = \rho_i$, $\rho_2 = \rho_{-i}$. We substitute $\rho_1 = \rho_i$, $\rho_2 = \rho_{-i}$ into the Eq.36 to find the AoI of source i in **NSLS-Q** system.

$$\begin{cases} \begin{bmatrix} \bar{v}_{00} & \bar{v}_{01} & \bar{v}_{02} \end{bmatrix} \lambda = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \pi_0 + \mu \begin{bmatrix} \bar{v}_{11} & \bar{v}_{11} & \bar{v}_{12} \end{bmatrix} + \mu \begin{bmatrix} \bar{v}_{20} & \bar{v}_{21} & \bar{v}_{22} \end{bmatrix} \\ \begin{bmatrix} \bar{v}_{10} & \bar{v}_{11} & \bar{v}_{12} \end{bmatrix} (\lambda + \mu) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \pi_1 + \lambda_1 \begin{bmatrix} \bar{v}_{00} & 0 & \bar{v}_{02} \end{bmatrix} + \mu \begin{bmatrix} \bar{v}_{31} & \bar{v}_{32} & \bar{v}_{32} \end{bmatrix} + \mu \begin{bmatrix} \bar{v}_{50} & \bar{v}_{52} & \bar{v}_{52} \end{bmatrix} \\ \begin{bmatrix} \bar{v}_{20} & \bar{v}_{21} & \bar{v}_{22} \end{bmatrix} (\lambda + \mu) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \pi_2 + \lambda_2 \begin{bmatrix} \bar{v}_{00} & \bar{v}_{00} & \bar{v}_{02} \end{bmatrix} + \mu \begin{bmatrix} \bar{v}_{41} & \bar{v}_{41} & \bar{v}_{42} \end{bmatrix} + \mu \begin{bmatrix} \bar{v}_{60} & \bar{v}_{60} & \bar{v}_{62} \end{bmatrix} \\ \begin{bmatrix} \bar{v}_{30} & \bar{v}_{31} & \bar{v}_{32} \end{bmatrix} (\lambda_1 + \mu) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \pi_3 + \lambda_1 \begin{bmatrix} \bar{v}_{10} & \bar{v}_{11} & 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} \bar{v}_{30} & 0 & \bar{v}_{32} \end{bmatrix} + \lambda_1 \begin{bmatrix} \bar{v}_{50} & 0 & \bar{v}_{52} \end{bmatrix} \\ \begin{bmatrix} \bar{v}_{40} & \bar{v}_{41} & \bar{v}_{42} \end{bmatrix} (\lambda_1 + \mu) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \pi_4 + \lambda_2 \begin{bmatrix} \bar{v}_{10} & \bar{v}_{11} & \bar{v}_{11} \end{bmatrix} + \lambda_1 \begin{bmatrix} \bar{v}_{40} & 0 & 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} \bar{v}_{60} & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} \bar{v}_{50} & \bar{v}_{51} & \bar{v}_{52} \end{bmatrix} (\lambda_1 + \mu) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \pi_5 + \lambda_1 \begin{bmatrix} \bar{v}_{20} & \bar{v}_{20} & 0 \end{bmatrix} \\ \begin{bmatrix} \bar{v}_{60} & \bar{v}_{61} & \bar{v}_{62} \end{bmatrix} (\lambda_1 + \mu) = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \pi_6 + \lambda_2 \begin{bmatrix} \bar{v}_{20} & \bar{v}_{20} & \bar{v}_{20} \end{bmatrix} \end{cases} \tag{28}$$

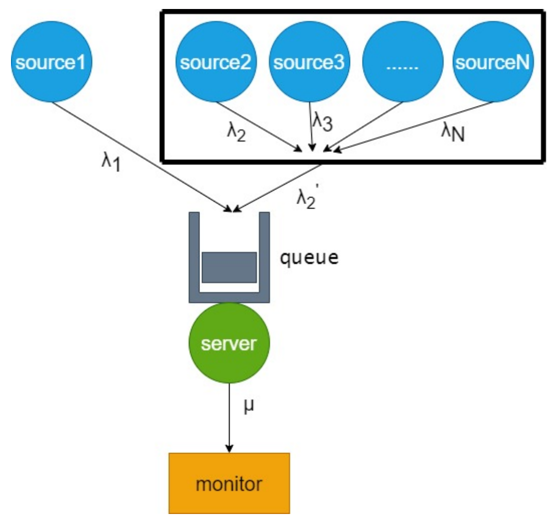

Fig. 4. NSLS-Q System

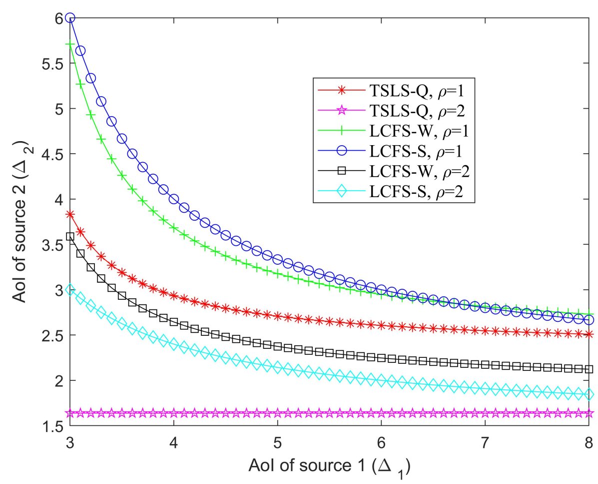

Fig. 5. Relationship between AoI of source 1 and AoI of source 2 for different models, where $\mu = 1$

VI. Numerical Results

In this section, we verify that our queueing system performs better in terms of AoI with some numerical results.

The system we study in this paper is **TSLS-Q** system and **NSLS-Q** system. The age of the information in the first source of the **TSLS-Q** system is given in Eq.36. We compare **TSLS-Q** system with the following queue model systems. We study the age of information of other queueing systems for multi-source systems in Section II. The age of information of the ith source of the FCFS system is given in Eq.11. The age of information of the ith source of the LCFS-W system is given in Eq.13. The age of information of the ith source of the LCFS-S system is given in Eq.12.

We first compare the relationship between the AoI of source 1 and the AoI of source 2 for different queueing systems. As shown in Fig.5. For the convenience of our study, we set the parameter $\mu = 1$ and also set $\rho = 1$ or $\rho = 2$. In Fig.5, we can find that the age of information of **TSLS-Q** system is significantly lower than the age of information of the other two queueing systems when ρ is equal. In addition, we can find that for the same queueing system, the age of information is smaller when ρ is larger.

Fig.6 presents comparisons of relationship between Sum AoI $(\Delta_1 + \Delta_2)$ and μ, where $\rho_1 = \rho_2 = 0.5$. We can observe that the sum AoI of various models has been decreasing continuously as the service rate (μ) increases. When μ is equal, sum AoI of **TSLS-Q** system is always lower than the LCFS-W system and LCFS-S system.

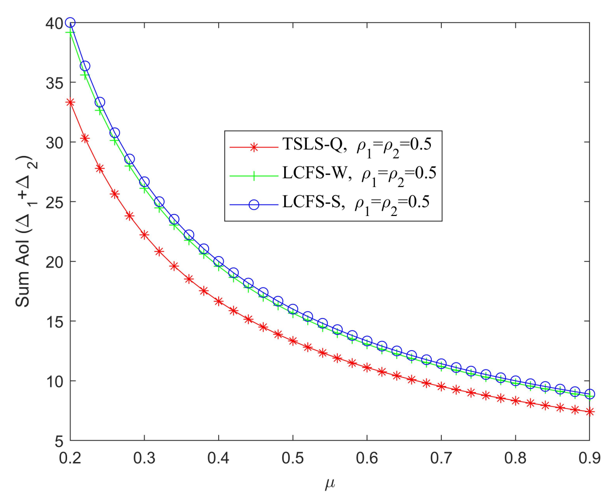

Fig. 6. Relationship between Sum AoI $(\Delta_1+\Delta_2)$ and μ for different models, where $\rho_1 = \rho_2 = 0.5$

We then examined relationship between Sum AoI $(\Delta_1+\Delta_2)$

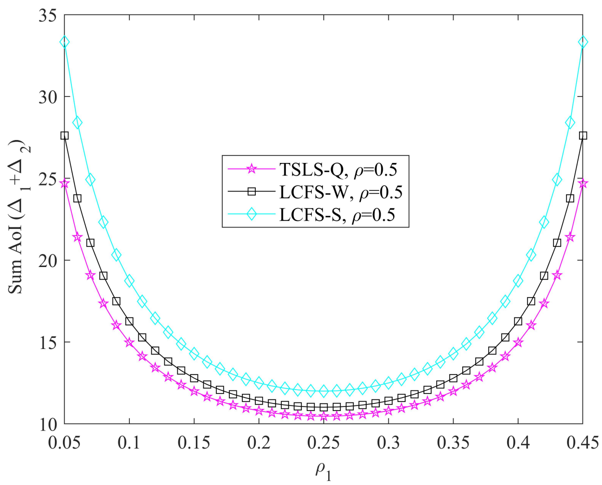

Fig. 7. Relationship between Sum AoI ($\Delta_1 + \Delta_2$) and ρ_1 for different models, where $\mu = 1$ and $\rho = 0.5$

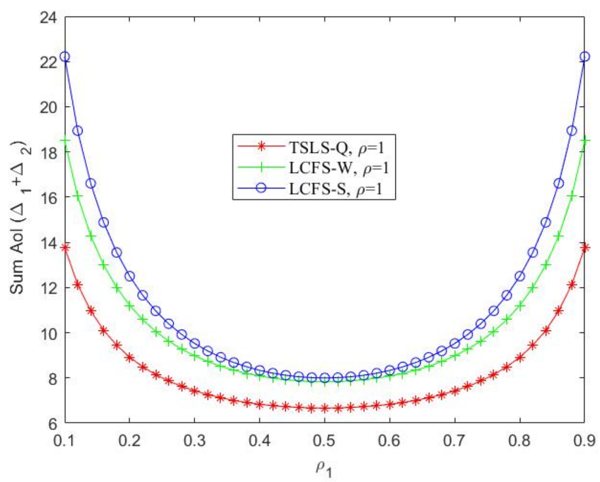

Fig. 8. Relationship between Sum AoI ($\Delta_1 + \Delta_2$) and ρ_1 for different models, where $\mu = 1$ and $\rho = 1$

and ρ_1 for different models. In all cases, we all set the parameter $\mu = 1$. When we set $\rho = 0.5, \rho = 1, \rho = 2, \rho = 6$ respectively, the results are shown in Fig.7, Fig.8, Fig.9, and Fig.10 respectively. By looking at Fig.7, Fig.8, Fig.9, and Fig.10, we can conclude that the sum AoI ($\Delta_1+\Delta_2$) of **TSLS-Q** system is smaller than LCFS-W system and LCFS-S system. At the same time, we found that the sum AoI ($\Delta_1 + \Delta_2$) of various models decreases first and then increases as ρ_1 changes.

VII. Conclusion and Future Work

In this paper, we consider a multi-source single-server single-monitor state update system. We first study the queueing system for two sources, single buffer, non-source-aware, and LCFS-S. We then extrapolated our conclusions to N sources. We derive the age of information of **TSLS-Q** system and **NSLS-Q** system using the stochastic hybrid system. And it is shown by numerical results that the queueing system has a smaller information age compared to some other conventional queueing systems in the same case.

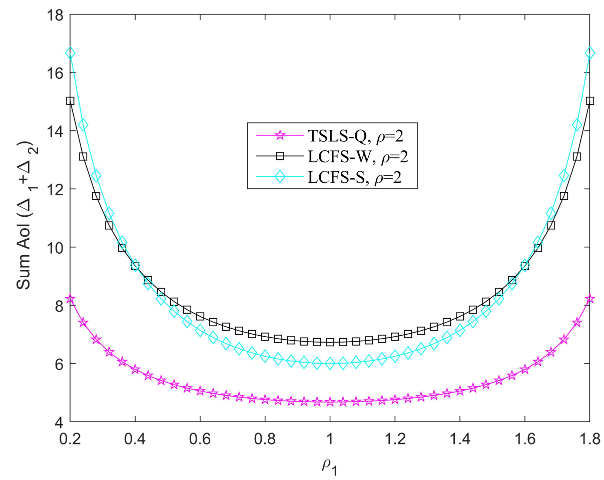

Fig. 9. Relationship between Sum AoI ($\Delta_1 + \Delta_2$) and ρ_1 for different models, where $\mu = 1$ and $\rho = 2$

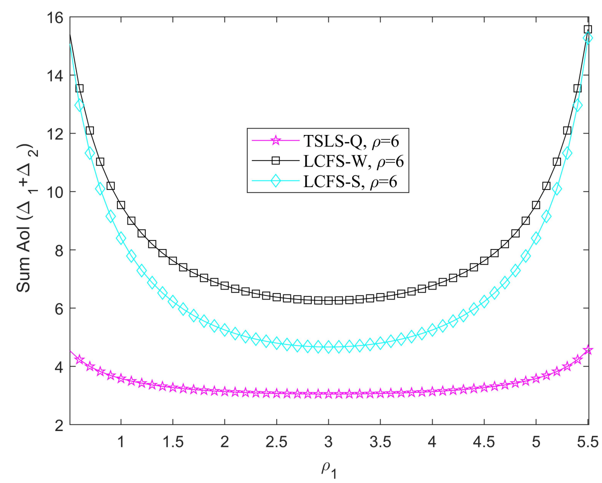

Fig. 10. Relationship between Sum AoI ($\Delta_1 + \Delta_2$) and ρ_1 for different models, where $\mu = 1$ and $\rho = 6$

In this paper, the **TSLS-Q** system and the **NSLS-Q** system, are both modeled in a multi-source and single-hop network, i.e., only one hop between the sender and the server. In some utilities, packets are forwarded to their destination through a multi-hop queueing network. So in our future work. Our work will be extended to multi-hop queuing networks. Further, our network should be more complex, which includes wireless access points, routers, base stations, wired networks, wireless networks, and so on, as shown in Fig.11. And the model will be applied to the actual complex network in the future. Also, the research has a role in many currently popular areas. A promising direction in edge computing is low latency edge assistance applications. It is expected that updated deliveries are computed in the cloud in a timely manner. AoI provides the freshness of the individual packets to the system.

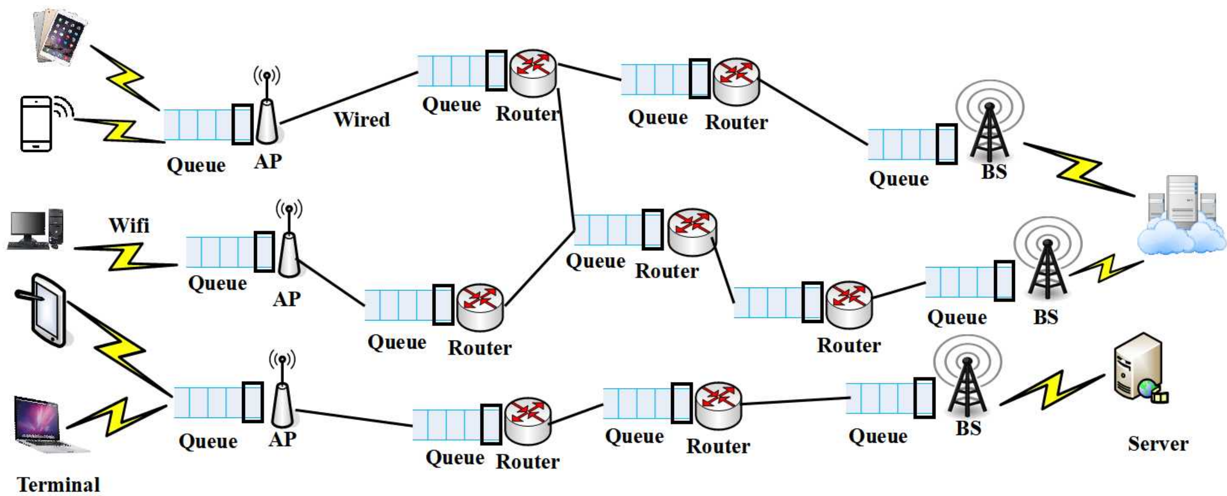

Fig. 11. multihop queueing network

References

[1] Sanjit Kaul, Marco Gruteser, Vinuth Rai, and John Kenney. Minimizing age of information in vehicular networks. In *2011 8th Annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks*, pages 350–358. IEEE, 2011.

[2] Sanjit Kaul, Roy Yates, and Marco Gruteser. On piggybacking in vehicular networks. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pages 1–5. IEEE, 2011.

[3] Sanjit Kaul, Roy Yates, and Marco Gruteser. Real-time status: How often should one update? In *2012 Proceedings IEEE INFOCOM*, pages 2731–2735. IEEE, 2012.

[4] Clement Kam, Sastry Kompella, and Anthony Ephremides. Experimental evaluation of the age of information via emulation. In *MILCOM 2015-2015 IEEE Military Communications Conference*, pages 1070–1075. IEEE, 2015.

[5] Canberk Sönmez, Sajjad Baghaee, Abdussamed Ergişi, and Elif Uysal-Biyikoglu. Age-of-information in practice: status age measured over tcp/ip connections through wifi, ethernet and lte. In *2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–5. IEEE, 2018.

[6] Maice Costa, Marian Codreanu, and Anthony Ephremides. On the age of information in status update systems with packet management. *IEEE Transactions on Information Theory*, 62(4):1897–1910, 2016.

[7] Longbo Huang and Eytan Modiano. Optimizing age-of-information in a multi-class queueing system. In *2015 IEEE international symposium on information theory (ISIT)*, pages 1681–1685. IEEE, 2015.

[8] Elie Najm and Rajai Nasser. Age of information: The gamma awakening. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 2574–2578. Ieee, 2016.

[9] Yoshiaki Inoue, Hiroyuki Masuyama, Tetsuya Takine, and Toshiyuki Tanaka. A general formula for the stationary distribution of the age of information and its application to single-server queues. *IEEE Transactions on Information Theory*, 65(12):8305–8324, 2019.

[10] Elie Najm and Emre Telatar. Status updates in a multi-stream m/g/1/1 preemptive queue. In *IEEE Infocom 2018-Ieee Conference On Computer Communications Workshops (Infocom Wkshps)*, pages 124–129. IEEE, 2018.

[11] Alkan Soysal and Sennur Ulukus. Age of information in g/g/1/1 systems: Age expressions, bounds, special cases, and optimization. *arXiv preprint arXiv:1905.13743*, 2019.

[12] Jaya Prakash Champati, Hussein Al-Zubaidy, and James Gross. On the distribution of aoi for the gi/gi/1/1 and gi/gi/1/2 systems: Exact expressions and bounds. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 37–45. IEEE, 2019.

[13] Sanjit Kaul, Marco Gruteser, Vinuth Rai, and John Kenney. Minimizing age of information in vehicular networks. In *2011 8th Annual IEEE communications society conference on sensor, mesh and ad hoc communications and networks*, pages 350–358. IEEE, 2011.

[14] Clement Kam, Sastry Kompella, Gam D Nguyen, Jeffrey E Wieselthier, and Anthony Ephremides. Controlling the age of information: Buffer size, deadline, and packet replacement. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 301–306. IEEE, 2016.

[15] Clement Kam, Sastry Kompella, Gam D Nguyen, Jeffrey E Wieselthier, and Anthony Ephremides. On the age of information with packet deadlines. *IEEE Transactions on Information Theory*, 64(9):6419–6428, 2018.

[16] Yoshiaki Inoue. Analysis of the age of information with packet deadline and infinite buffer capacity. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2639–2643. IEEE, 2018.

[17] Jie Li, Yong Zhou, and He Chen. Age of information for multicast transmission with fixed and random deadlines in iot systems. *IEEE Internet of Things Journal*, 7(9):8178–8191, 2020.

[18] Jie Gong, Qiaobin Kuang, Xiang Chen, and Xiao Ma. Reducing age-of-information for computation-intensive messages via packet replacement. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2019.

[19] Antzela Kosta, Nikolaos Pappas, Anthony Ephremides, and Vangelis Angelakis. Age of information performance of multiaccess strategies with packet management. *Journal of Communications and Networks*, 21(3):244–255, 2019.

[20] Roy D Yates and Sanjit K Kaul. The age of information: Real-time status updating by multiple sources. *IEEE Transactions on Information Theory*, 65(3):1807–1827, 2018.

[21] Alireza Javani, Marwen Zorgui, and Zhiying Wang. Age of information in multiple sensing. In *2020 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2020.

[22] Mohammad Moltafet, Markus Leinonen, and Marian Codreanu. Average aoi in multi-source systems with source-aware packet management. *IEEE Transactions on Communications*, 69(2):1121–1133, 2020.

[23] Sanjit K Kaul and Roy D Yates. Age of information: Updates with priority. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2644–2648. IEEE, 2018.

[24] Roy D Yates and Sanjit Kaul. Real-time status updating: Multiple sources. In *2012 IEEE International Symposium on Information Theory Proceedings*, pages 2666–2670. IEEE, 2012.

[25] Rajat Talak, Sertac Karaman, and Eytan Modiano. Minimizing age-of-information in multi-hop wireless networks. In *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 486–493. IEEE, 2017.

[26] Shahab Farazi, Andrew G Klein, John A McNeill, and D Richard Brown. On the age of information in multi-source multi-hop wireless status update networks. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5. IEEE, 2018.

[27] Vishrant Tripathi and Sharayu Moharir. Age of information in multi-source systems. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.

[28] Mohammad Moltafet, Markus Leinonen, and Marian Codreanu. On the age of information in multi-source queueing models. *IEEE Transactions on Communications*, 68(8):5003–5017, 2020.

[29] Ahmed M Bedewy, Yin Sun, Sastry Kompella, and Ness B Shroff. Age-optimal sampling and transmission scheduling in multi-source systems. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 121–130, 2019.

[30] Nikolaos Pappas, Johan Gunnarsson, Ludvig Kratz, Marios Kountouris, and Vangelis Angelakis. Age of information of multiple sources with queue management. In *2015 IEEE international conference on communications (ICC)*, pages 5935–5940. IEEE, 2015.

[31] Ahmed M Bedewy, Yin Sun, Sastry Kompella, and Ness B Shroff. Optimal sampling and scheduling for timely status updates in multi-source networks. *IEEE Transactions on Information Theory*, 67(6):4019–4034, 2021.

[32] Joao P Hespanha. Modelling and analysis of stochastic hybrid systems. *IEE Proceedings-Control Theory and Applications*, 153(5):520–535, 2006.

[33] Domokos Vermes. Optimal dynamic control of a useful class of randomly jumping processes. 1980.

[34] Mark HA Davis. Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(3):353–376, 1984.

[35] Lee DeVille, Sairaj Dhople, Alejandro D Domínguez-García, and Jiangmeng Zhang. Moment closure and finite-time blowup for piecewise deterministic markov processes. *SIAM Journal on Applied Dynamical Systems*, 15(1):526–556, 2016.

[36] Roy D Yates. Age of information in a network of preemptive servers. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 118–123. IEEE, 2018.

[37] Roy D Yates. Status updates through networks of parallel servers. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2281–2285. IEEE, 2018.

[38] Sanjit K Kaul and Roy D Yates. Timely updates by multiple sources: The m/m/1 queue revisited. In *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2020.

[39] Roy D Yates. The age of information in networks: Moments, distributions, and sampling. *IEEE Transactions on Information Theory*, 66(9):5712–5728, 2020.

[40] Nail Akar, Ozancan Doğan, and Eray Unsal Atay. Finding the exact distribution of (peak) age of information for queues of ph/ph/1/1 and m/ph/1/2 type. *IEEE Transactions on Communications*, 68(9):5661–5672, 2020.

[41] Maice Costa, Marian Codreanu, and Anthony Ephremides. Age of information with packet management. In *2014 IEEE International Symposium on Information Theory*, pages 1583–1587. IEEE, 2014.