Age of Information of an FCFS Queueing System in Heterogeneous Servers with Proactive Obsolete Packet Management

Y. Arun Kumar Reddy\textsuperscript{1} and T G Venkatesh\textsuperscript{1}

\textsuperscript{1}Affiliation not available

January 26, 2024
Age of Information of an FCFS Queueing System in Heterogeneous Servers with Proactive Obsolete Packet Management

Y. Arun Kumar Reddy, and T.G. Venkatesh

Abstract—The Age of Information (AoI) is a metric in a remote monitoring system to evaluate the timeliness of status updates received at the destination. In this paper, we study the AoI and Peak AoI (PAoI) in a queuing system with two parallel heterogeneous servers and no extra buffer. Analyzing AoI or PAoI with a system having two parallel heterogeneous servers is challenging because of the out-of-order reception of packets. A stochastic hybrid systems (SHS) approach has been used to analyze the performance of the system. We study the system’s performance under two different queueing disciplines: First Come First Serve (FCFS) and FCFS with probabilistic routing. The findings are then compared to a similar system with two homogeneous servers. A novel methodology called "Proactive Obsolete Packet Management" denoted as "POPMAN", has been proposed to proactively identify obsolete packets, thereby reducing server processing time and improving the AoI and PAoI of the system. POPMAN technique has been applied to both FCFS and FCFS with probabilistic routing queueing systems. We observe that the proposed schemes exhibit improvement compared to traditional methods and also packets receive in order of reception. Our simulations are validated for various parameters, such as the packet dropping probability, the probability of obsolete packets due to outdated information in the servers, probability that a packet is informative or successfully delivered, and optimal splitting probabilities for probabilistic routing.

Index terms—Age of Information, Peak Age of Information, probabilistic routing, stochastic hybrid system, Heterogeneous servers, first come first serve.

I. INTRODUCTION

In the future, it is expected that the 5G system will face challenges in managing the increasing mobile traffic. To tackle this problem, the development of the sixth generation (6G) mobile communication system is underway. This new 6G system aims to support different scenarios that demand either massive machine-type communication (MMTC) or ultra-reliable low-latency communication (URLLC). [1], [2]. These scenarios may involve remote monitoring and control, where the timeliness of information is pivotal.

The concept of the Age of Information (AoI) is arisen as a measure to evaluate the timeliness of information in remotely monitored processes. Its importance has been widely recognized in various communication systems, making it a subject of significant interest. To delve deeper into this subject, readers are advised to refer [3], [4]. Queueing systems are often used to represent information update systems, with incoming packets representing information packets.

Researchers from various fields have been drawn to AoI since its establishment. The appeal of AoI lies in its unique method of measuring the timeliness of information, which sets it apart from conventional metrics like delay or latency. AoI (simply 'Age') has been thoroughly examined in diverse systems, functioning as a concept, performance metric, and tool. Nonetheless, it is crucial to differentiate between timely updates and the goal of maximizing communication system utilization or minimizing delay in receiving status updates. Prioritizing utilization may cause delays in statuses due to message backlog, while reducing the update rate can decrease delay but may lead to outdated information.

The AoI is determined by measuring the time that has passed since the monitor received its latest successful update [5], [6]. The calculation of the AoI involves subtracting the time-stamp \( u(t) \) from the given time denoted as "t" when the monitor receives an update at time \( t \). This calculation is represented by the random process \( \Delta(t) = t - u(t) \).

The average AoI, commonly employed for AoI evaluation, is denoted as the time average of \( \Delta(t) \).

To determine the average age of status updates in a system, we need to examine the relationship between the variation in age \( \Delta(t) \) and time \( t \) on the monitor. By observing how the age variation \( \Delta(t) \) changes over time, as shown in Figure 1, we can analyze the pattern. Initially, at time \( t = 0 \) when the queue is empty, the age is denoted as \( \Delta_0 \). The status updates occur at different time points \( t_1, t_2, t_3, ..., t_n \). As a result, the AoI at the monitor, \( \Delta(t_i) \), is updated to \( t_i - t_i \) for each status update. The age function \( \Delta(t) \) follows a saw-tooth pattern, as depicted in Figure 1. To calculate the average age within a duration \((0,T)\), we can use the formula \( \bar{\Delta} = \frac{1}{T} \int_0^T \Delta(t)dt \).

Fig. 1: Variation in age at a monitor with FCFS queueing

Another important measure for evaluating AoI is Peak AoI (PAoI), which specifically assesses the maximum instantaneous AoI [7]. This PAoI metric plays a significant role in assessing the freshness of information.

A. Related works

Initially, researchers concentrated on examining the average AoI. Kaul et al. have investigated various queueing models,
which includes an M/M/1, M/D/1, and D/M/1 queues with FCFS discipline [5]. Various studies have carried out different scheduling strategies, packet management strategies and service disciplines to enhance the freshness of information [6]–[8]. Kaul et al. have analyzed the average AoI of the M/M/1 queue with Last Come First Serve (LCFS) discipline [6]. Similarly, Costa et al. have examined the average AoI of M/M/1/1, M/M/1/2, and M/M/1/2* queues and demonstrated that packet management can decrease the average AoI under high arrival rates. Subsequently, new metric called peak AoI (PAoI) has been introduced [7]. Pappas et al. have discussed the M/M/1 FCFS queueing system, where preemption is allowed in waiting queue [8]. Inoue et al. have obtained a general formula of AoI for the stationary distribution. Subsequently, an analysis has been conducted to determine the average AoI for both M/G/1 and G/M/1 systems in [9].

In the aforementioned studies, the focus was primarily on single server queue models for the status process. Nevertheless, it is crucial to acknowledge that these models are not a precise representation of networks having multiple paths for packet transmission. Despite the increased cost, it is advantageous to use such models in situations where strict information freshness is necessary, such as health monitoring, autonomous driving, and fire alarms. Additionally, including extra devices for redundancy is feasible in these scenarios. To address this limitation, researchers have also examined the average AoI with parallel servers in various networks [10]–[20]. For instance, Kam et al. explored the utilization of multiple servers, such as the M/M/2 and M/M/∞ queue under the FCFS queue model in [10] and [11]. Yates et al. have extended the investigation to include the general M/M/c queueing model with LCFS queue [12]. Additionally, Javani et al. have analyzed a LCFS queueing system with multiple source and multiple server, where preemption in service is allowed [13]. In studies [12] and [13], centralized queues were considered, allowing for communication between queues. Bhati et al. have analyzed average AoI in a queueing system with two parallel M/M/1 Queues [14]. Conversely, Doncel et al. have focused on a system with multiple sources having parallel queues. Notably, these queues have been assumed to be decentralized, meaning they could not communicate with each other [15]. Similarly AoI in dual sensor updating systems has been analyzed in [18]–[20].

The graphical method has been initially proposed and is widely using as a means to analyze average AoI and PAoI by decomposing the AoI process graphically in various studies [5]–[7], [9]–[11], [14], [19], [21]–[27]. However, Yates et al. have demonstrated that an alternative method for AoI analysis in a status update system is with a stochastic hybrid system (SHS) [28]. This SHS approach is particularly valuable in networks with packet loss, where graphical analysis of AoI and PAoI can be difficult [29]. SHS is widely utilized in AoI analysis [13], [15], [20], [28]–[34]. It is crucial to acknowledge that an SHS may not accurately model all updating systems. While earlier studies primarily focused on evaluating average AoI, more recent research Asvadi et al. derived a general expression for finding PAoI using an SHS [34]. For more information on SHS analysis, refer to [35].

In previous studies [12], [20], there is a gap that needs to find an alternative way that analyzes AoI more simply than the existing methods. An alternate way of analyzing AoI is significant when dealing with systems that experience out-of-order packet reception, as the increasing number of states makes an analysis challenging. To tackle this, in our previous research [36], we have introduced a new approach called Relative freshness SHS Markov chain (RF-SHS-MC) analysis, which helps to identify the obsolete packets on the server and makes analysis easier compared to the existing methods. We also recommend our another related work [37].

B. Aim and Unique contributions

This paper aims to analyze the system under consideration with the FCFS queueing discipline using the proposed POPMAN approach and also compare it with the FCFS queueing discipline using the traditional approach. We consider two types of FCFS queueing disciplines: 1) FCFS and 2) FCFS with probabilistic routing.

Unique contributions are as follows

1) In contrast to the previous work [12], where servers were modeled as ./M/1/1 with LCFS preemption, our analysis models servers as ./M/1/1 with FCFS and M/M/1/1 under the FCFS with probabilistic routing.

2) In contrast to the study conducted in [14], which focused on the average AoI in two parallel M/M/1 queues using the FCFS queueing system, our investigation examines the average AoI in two parallel M/M/1/1 queues also operating under the FCFS queueing system.

3) Unlike analyzing the decentralised network with LCFS queueing system in [15], we consider the centralized network which utilizes a proactive approach (proposed POPMAN) to identify and remove outdated packets from the servers before sending them to the monitor. This approach optimizes server resources, allowing them to handle incoming packets more efficiently with extra signal overhead and increased system complexity. The proactive approach is based on a RF-SHS-MC analysis, as described in [36], which enhances the AoI or PAoI. In addition, the packets received at the receiver are in order of reception. In our study, we introduce POPMAN approach to traditional queueing disciplines FCFS and FCFS with probabilistic routing, referred to as FCFS using the POPMAN approach and FCFS with probabilistic routing using the POPMAN approach.

4) In the context of traditional FCFS systems, a comparison has been conducted between a single server with a service rate of 1 and two servers with an equal cumulative service rate ($\mu_1 + \mu_2 = 1$). The findings revealed that when the service rate is divided among multiple servers, the AoI performance deteriorates. This implies that allocating a higher proportion of the service rate to a single server yields better AoI performance compared to splitting it among multiple servers.

5) A study has been conducted to compare the performance of single server systems with different service rates (1, 2, 4, 8, 16, 32) to that of two parallel homogeneous server systems with similar AoI characteristics using FCFS queueing discipline. The results shows that the equivalent service rates...
for each server in the parallel system are approximately 0.625 times the service rates of the single server system.

C. Main Contributions

The present research has involved an examination of the network comprising two parallel servers with heterogeneous characteristics and a comparison has been made with a scenario where the servers were homogeneous. In this study, the primary contributions are summarized as follows:

1) We have compared the AoI and PAoI performances of FCFS using the POPMAN approach and FCFS with probabilistic routing using the POPMAN approach with the traditional FCFS and FCFS with probabilistic routing methods respectively.

2) In addition to the average AoI and PAoI, packet dropping probability due to server busyness, the probability of obsolete or discarded packets due to outdated information in servers, the probability of informative packets or successfully delivered packets, and optimal splitting probabilities for all queueing disciplines FCFS, FCFS using the POPMAN approach, FCFS with probabilistic routing using the POPMAN approach queueing systems are presented in Section VI, Section VII and Section VIII respectively. Numerical results are discussed in section IX. Section X provides final remarks to conclude the discussion.

II. THE SHS ANALYSIS FRAMEWORK

A. AoI and PAoI

The SHS technique [35] is instrumental for the analysis of AoI. We briefly explain its main idea here, but for a more detailed understanding, we recommend [28].

The SHS approach utilizes a continuous-time finite-state Markov chain (CTMC) \( q(t) \in Q = \{0, 1, \ldots, m\} \) to demonstrate the progression of the system while employing continuous age-related processes \( x(t) = [x_0(t) \ldots x_n(t)] \in \mathbb{R}^{n+1} \) to portray the development of age-related processes within the system.

The Markov chain \( q(t) \) is represented by the graph \((Q, L)\). In this graphical representation, each node \( q \in Q \) represents a state, and each directed edge \((q, q') \in L\) represents a transition with a transition rate \( \lambda(q, q') \) when \( q(t) = q \). In this context, \( \delta_{q,q'}(t) \) is the Kronecker delta function, confirming that transition \( l \) occurs exclusively in state \( q \). Whenever a transition \( l \) occurs, the discrete state jumps to \( q' \), and the continuous state is reset from \( x \) to \( x' = xA_l \), where \( A_l \in \{0,1\}^{(n+1) \times (n+1)} \) is the reset map of transition \( l \). It is important to note that unlike a traditional continuous time Markov chain, the SHS may have self-transitions, where the state remains the same, but a reset occurs in the continuous state.

The piece-wise linear function that characterizes the continuous process in each state \( q \in Q \) is represented by the differential equation \( \dot{x} = b_q = [b_{q0} b_{q1} \ldots b_{qn}] \). The binary values in \( b_q \) indicate whether the age process \( x_i \), \( i \in \{0,1,\ldots,n\} \), increases at a unit rate \( (b_{qi} = 1) \) or does not \( (b_{qi} = 0) \) [28].

In order to compute the average AoI in an SHS, it is crucial to determine both the state probability and the correlation vector that quantifies the correlation between the discrete state \( q(t) \) and the age process \( x(t) \). The probability of the Markov chain being in state \( q \) is denoted as \( \pi_q \), while the correlation vector is represented by \( v_q \). These quantities, \( \pi_q \) and \( v_q \), are defined as follows [28]

\[
\pi_q(t) = E[\delta_{q,q(t)}] = Pr[q(t) = q],
\]

\[
v_q(t) = [v_{q0}(t) v_{q1}(t) \ldots v_{qn}(t)] = E[x(t)\delta_{q,q(t)}].
\]

The existence of the steady state probability vector \( \pi = [\pi_0 \ldots \pi_m] \) as the unique solution to the following equations is guaranteed under the assumption that the Markov chain \( q(t) \) is ergodic [28].

\[
\pi_q = \sum_{l \in L_q} \lambda(l) = \sum_{l \not\in L_q} \lambda(l), \quad \forall q \in Q,
\]

\[
\sum_{q \in Q} \pi_q = 1
\]

The sets \( L_q = \{l \in L : q' = q\} \), \( L_q = \{l \in L : q = q\} \) represent the incoming and outgoing transitions for a state \( q \) in the system, respectively. As the system stabilizes, the correlation vector \( v_q(t) \) approaches a non-negative limit \( \bar{v}_q \) as time \( t \to \infty \) \( \forall q \in Q \). In this particular scenario, it has

\[
\mathbb{E}[x(t)] = \lim_{t \to \infty} \mathbb{E}[x(t)] = \lim_{t \to \infty} \sum_{q \in Q} \mathbb{E}[x(t)\delta_{q,q(t)}] = \sum_{q \in Q} v_q
\]

The average AoI of the system under consideration can be obtained using the subsequent lemma.

**Lemma 1.** [28, Theorem 4] If the Markov chain \( q(t) \) has an ergodic property with a stationary distribution \( \pi \), one can find \( \bar{v} = [\bar{v}_0 \ldots \bar{v}_m] \) that is non-negative vector as the solution of

\[
\pi_q \sum_{l \in L_q} \lambda(l) = b_q \pi_q + \pi_q \sum_{l \not\in L_q} \lambda(l) \bar{v}_q A_l, \quad \forall q \in Q.
\]

Then the average AoI is given by

\[
\Delta = \mathbb{E}[x_0] = \sum_{q \in Q} v_q
\]

The average PAoI of the system under consideration can be obtained using the following lemma.

**Lemma 2.** [34, Theorem 3] Consider a stable stationary and ergodic updating system that is modeled by an SHS with no fake updates and the set of updating transitions \( L \). If the CTMC of this SHS is ergodic with steady-state distribution \( \overline{\pi} > 0 \), and \( \bar{v}_q \) is a non-negative solution of (3), then, in this updating system the average PAoI is given by

\[
\Delta_P = \frac{\sum_{l \in L} \lambda(l) \pi_{q_l,0}}{\sum_{l \in L} \lambda(l) \bar{v}_q}
\]
III. SYSTEM MODEL

We consider the system to consist of a source and destination connected by two parallel servers. We assume arrival rate is Poisson distributed with parameter $\lambda$ and updates from the source are delivered to the destination through the parallel servers. We also assume service times are exponentially distributed with parameters $\mu_1$ and $\mu_2$ for server 1 and server 2 respectively. The network configuration is illustrated in Figure 2. When an update is generated at the source, it needs to be assigned to a server and then transmitted to a remote monitor. The updates may reach the destination in a non-sequential order due to differences in the service time between the servers. In our separate work [36], we have developed analytical expressions to determine the average AoI in a single server system using a RF-SHS-MC approach for defining the states, which allows for the efficient use of SHS. Table I presents the novel symbols introduced in [36] to measure the relative freshness of packets in servers, that aids in the definition of states. In the symbol ‘$1^f$’ given in Table I, ‘$1$’ represents a source number and ‘$f$’ indicates a fresh or relatively fresh packet that can improve AoI when delivered. On the other hand, in the symbol ‘$1^o$’, ‘$o$’ represents an old packet and ‘$1$’ indicates that the packet can improve AoI when delivered. Similarly, in the symbol ‘$1^o$’, ‘$o$’ indicates that the packet does not improve AoI when delivered. These symbols can be extended for queueing systems with a capacity of more than two, such as $1^f$, $1^o$, $2^f$, $2^o$, $\ldots$, and $1^{10}$, where $1^o$ is relatively fresher than $2^f$ and $2^o$ is relatively fresher than $1^o$, and so on. Similarly, these symbols can be extended for multiple sources by replacing the source number ‘$1$’ with the respective source number 2, 3, and so on.

![Fig. 2: Heterogeneous network model with two parallel servers](image)

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1^f$</td>
<td>It is a fresh packet sent from source-1 to server-1 or server-2 during the transitional phase, which has the potential to enhance the AoI it reaches its intended destination. (OR) It is an old packet in server-1 or server-2 originated from other sources, which has the potential to enhance the AoI it reaches its intended destination.</td>
</tr>
<tr>
<td>$1^o$</td>
<td>It is an old packet in server-1 or server-2 originated from source-1 and relatively old compared to others, which can enhance the AoI it reaches its intended destination.</td>
</tr>
<tr>
<td>$\times 1^o$</td>
<td>It is an old packet in server-1 or server-2 originated from source-1 and relatively old compared to others, which does not have the potential to enhance the AoI it reaches its intended destination.</td>
</tr>
</tbody>
</table>

TABLE I: Symbols for RF-SHS-MC model

In this study, we expand our analysis carried out in [36] to include two parallel heterogeneous servers. We calculate the average AoI and the average PAoI, as with other relevant metrics.

Table II shows the actual states of the server in terms of relative freshness of packets using RF-SHS-MC method [36]. For instance, when $q = 0$, it signifies that both servers are idle, visually represented by two empty rectangular boxes stacked on top of each other. On the other hand, when $q = 1$, it indicates that server 1 is busy serving a 1$^f$ packet, which is either fresh/old packet (as defined in Table I), while server 2 remains idle. This state is depicted as 1$^f$ in the upper rectangular box and empty in the lower rectangular box. Lastly, when $q = 8$, it signifies that server 1 is busy serving a 1$^f$ packet, which is either fresh/old packet (as defined in Table I), while server 2 is serving an old packet (denoted as $\times 1^o$) that does not improve the AoI (as defined in Table I). This state represents both servers being busy, shown as 1$^f$ in the upper rectangular box and $\times 1^o$ in the lower rectangular box. For convenience we represent the actual states with the numbers as shown in Table II.

Table III shows the abbreviations used throughout this paper. In addition, following are the notations used throughout this paper.

$$
\gamma_1 = \frac{\mu_1}{\mu_1 + \mu_2}; \quad \gamma_2 = \frac{\mu_2}{\mu_1 + \mu_2}; \quad \gamma_3 = \frac{\lambda}{\lambda + \mu_1}; \quad \gamma_4 = \frac{\lambda}{\lambda + \mu_2} \\
\gamma_5 = \frac{\mu_1}{\lambda + \mu_1 + \mu_2}; \quad \gamma_6 = \frac{\mu_2}{\lambda + \mu_1 + \mu_2}; \quad \gamma_7 = \frac{\lambda}{\lambda + \mu_1 + \mu_2} \\
\gamma_8 = \frac{\mu_1}{\lambda + \mu_1}; \quad \gamma_9 = \frac{\mu_2}{\lambda + \mu_1}; \quad \gamma_{10} = \frac{1}{1 - \gamma_7}; \quad \gamma_{11} = \frac{\lambda + \mu_1}{\mu_1 + \lambda \mu_2} \\
\gamma_{12} = \frac{\lambda + \mu_2}{\mu_2 + \lambda \mu_1} \delta_1 = \lambda^2 + \mu_1 (\lambda + \delta_2); \quad \delta_2 = \lambda + \mu_1 + \mu_2
$$

IV. FCFS QUEUEING SYSTEM

In this FCFS queueing system, the arriving packet is assigned to the server with the highest service rate if it is free, resulting in a better AoI. If the server with a higher service rate is busy, packets are sent to the server with a lower service rate. If both servers are busy, packets are dropped. Packets are sent to the monitor after service completion.

Figure 3 represents the SHS Markov chain for FCFS queueing system. In this, the Markov chain’s state space is represented by the set $Q = \{0, 1, \ldots, 8\}$ with each state described in Table II.

The function $x(t)$ represents the change in age over continuous time $t$, where $x(t) = [x_0(t) \ x_1(t) \ x_2(t)]$. The value $x_0(t)$ indicates the source’s current AoI at time $t$. On the other hand, $x_1(t)$ and $x_2(t)$ represent the AoI that would result if the packets handled by server 1 and server 2 transmit to the monitor at time $t$. Therefore, in discrete state $q(t) = q$, the continuous state emerges as

$$
\dot{x}(t) = b_q = \begin{cases} 
1 & q = 0 \\
1 & q = 1, 5 \\
1 & q = 2, 6 \\
1 & q = 3, 4, 7, 8 
\end{cases}
$$
The average AoI of source 1 can be obtained by evaluating $\bar{\lambda} = \lambda + \mu_1 - \mu_2$. The stationary probabilities can be determined.

\[ \bar{\pi} = \begin{bmatrix} 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

\[ \bar{\pi}D = \bar{\pi}Q \]

The average AoI of source 1 can be obtained by evaluating the values of $\pi_{0, q_0}$, $\forall q \in Q$ using Table IV, equation (6), substituting the stationary probabilities into equation (3), and solving the resulting linear equations. Finally, by replacing the values of $\pi_{0, q_0}$, $\forall q \in Q$ in equation (4), we can calculate the average AoI of source 1. In addition to the AoI and PAoI, the other relevant metrics can be derived as follows.

1) Probability of Informative Packets: The arrival packet becomes an informative packet only if one of the servers can accommodate it, and its service must be completed before the arrival and service of a new packet. For example, in state 0, the arriving packet is assigned to server 2 with a probability of 1. Therefore, the arriving packet occupies the second server, and the system transitions to state 2. The first term $\pi_{0, q_0}$ in equation (7) is a product of probabilities of three events. The system being in state 0 (represented by $\pi_0$), the arriving packet

### Table II: Assignment of actual states of the server to numbers which represent discrete states in SHS Markov chain

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCFS</td>
<td>First Come First Serve queueing system for single source and single destination with two servers of average service rates $\mu_1$ and $\mu_2$, where the scenario can be homogeneous or heterogeneous or with single server system is explicitly mentioned.</td>
</tr>
</tbody>
</table>

### Table III: Abbreviations and its representation

To calculate the stationary probability vector $\bar{\pi}$, we use equations (1) and (2). By using (1) and the state transitions outlined in Table IV, it is possible to demonstrate that $\bar{\pi}$ fulfills the equation $\bar{\pi}D = \bar{\pi}Q$. Using $\bar{\pi}D = \bar{\pi}Q$ and equation (2), the stationary probabilities can be determined.

\[ \bar{\pi} = \begin{bmatrix} 0 & 0 & \lambda & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

The average AoI of source 1 can be obtained by evaluating the values of $\pi_{0, q_0}$, $\forall q \in Q$ using Table IV, equation (6), substituting the stationary probabilities into equation (3), and solving the resulting linear equations. Finally, by replacing the values of $\pi_{0, q_0}$, $\forall q \in Q$ in equation (4), we can calculate the average AoI of source 1. In addition to the AoI and PAoI, the other relevant metrics can be derived as follows.

1) Probability of Informative Packets: The arrival packet becomes an informative packet only if one of the servers can accommodate it, and its service must be completed before the arrival and service of a new packet. For example, in state 0, the arriving packet is assigned to server 2 with a probability of 1. Therefore, the arriving packet occupies the second server, and the system transitions to state 2. The first term $\pi_{0, q_0}$ in equation (7) is a product of probabilities of three events. The system being in state 0 (represented by $\pi_0$), the arriving packet

### Table IV: Table of transition for the SHS Markov chain in Fig.3

<table>
<thead>
<tr>
<th>$l$</th>
<th>$q_j \rightarrow q_i$</th>
<th>$\lambda^{(j)}$</th>
<th>$xA_1$</th>
<th>$A_1$</th>
<th>$v_{q_i, A_1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0 \rightarrow 2</td>
<td>$\lambda$</td>
<td>$[x_0 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{00} 0 0]$</td>
</tr>
<tr>
<td>2</td>
<td>1 \rightarrow 3</td>
<td>$\mu_1$</td>
<td>$[x_1 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{01} 0 0]$</td>
</tr>
<tr>
<td>3</td>
<td>1 \rightarrow 0</td>
<td>$\mu_2$</td>
<td>$[x_2 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{02} 0 0]$</td>
</tr>
<tr>
<td>4</td>
<td>2 \rightarrow 4</td>
<td>$\lambda$</td>
<td>$[x_0 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{00} 0 0]$</td>
</tr>
<tr>
<td>5</td>
<td>2 \rightarrow 0</td>
<td>$\mu_2$</td>
<td>$[x_2 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{02} 0 0]$</td>
</tr>
<tr>
<td>6</td>
<td>3 \rightarrow 3</td>
<td>$\mu_1$</td>
<td>$[x_1 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{01} 0 0]$</td>
</tr>
<tr>
<td>7</td>
<td>3 \rightarrow 2</td>
<td>$\mu_2$</td>
<td>$[x_2 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{02} 0 0]$</td>
</tr>
<tr>
<td>8</td>
<td>3 \rightarrow 5</td>
<td>$\mu_2$</td>
<td>$[x_2 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{02} 0 0]$</td>
</tr>
<tr>
<td>9</td>
<td>4 \rightarrow 4</td>
<td>$\lambda$</td>
<td>$[x_0 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{00} 0 0]$</td>
</tr>
<tr>
<td>10</td>
<td>4 \rightarrow 6</td>
<td>$\mu_1$</td>
<td>$[x_1 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{01} 0 0]$</td>
</tr>
<tr>
<td>11</td>
<td>4 \rightarrow 1</td>
<td>$\mu_2$</td>
<td>$[x_2 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{02} 0 0]$</td>
</tr>
<tr>
<td>12</td>
<td>5 \rightarrow 7</td>
<td>$\lambda$</td>
<td>$[x_0 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{00} 0 0]$</td>
</tr>
<tr>
<td>13</td>
<td>5 \rightarrow 0</td>
<td>$\mu_1$</td>
<td>$[x_1 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{01} 0 0]$</td>
</tr>
<tr>
<td>14</td>
<td>6 \rightarrow 8</td>
<td>$\lambda$</td>
<td>$[x_0 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{00} 0 0]$</td>
</tr>
<tr>
<td>15</td>
<td>6 \rightarrow 0</td>
<td>$\mu_2$</td>
<td>$[x_2 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{02} 0 0]$</td>
</tr>
<tr>
<td>16</td>
<td>7 \rightarrow 7</td>
<td>$\lambda$</td>
<td>$[x_0 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{00} 0 0]$</td>
</tr>
<tr>
<td>17</td>
<td>7 \rightarrow 2</td>
<td>$\mu_1$</td>
<td>$[x_1 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{01} 0 0]$</td>
</tr>
<tr>
<td>18</td>
<td>7 \rightarrow 5</td>
<td>$\mu_2$</td>
<td>$[x_2 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{02} 0 0]$</td>
</tr>
<tr>
<td>19</td>
<td>8 \rightarrow 8</td>
<td>$\lambda$</td>
<td>$[x_0 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{00} 0 0]$</td>
</tr>
<tr>
<td>20</td>
<td>8 \rightarrow 6</td>
<td>$\mu_1$</td>
<td>$[x_1 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{01} 0 0]$</td>
</tr>
<tr>
<td>21</td>
<td>8 \rightarrow 1</td>
<td>$\mu_2$</td>
<td>$[x_2 0 0]$</td>
<td>$[0 0 0]$</td>
<td>$[v_{02} 0 0]$</td>
</tr>
</tbody>
</table>
packet occupying server 2 and transitioning to state 2 with a probability of 1, and the service of server 2 occurs before a new arrival (represented by $\gamma_9$). When the system is in a state 0, the probability that the arriving packet reaches its destination before a new arrival is $\pi_0 * 1 * \gamma_9 = \pi_0 \gamma_9$. Similarly, $\pi_0 * 1 * \gamma_4 * \gamma_6$ represents the probability that the arriving packet reaches its destination after a new arrival. $\pi_0 * 1 * \gamma_4 * \gamma_7 * \gamma_6$ represents the probability that the arriving packet reaches its destination after two new arrivals. $\pi_0 * 1 * \gamma_4 * \gamma_7 * \gamma_6$ represents the probability that the arriving packet reaches its destination after three new arrivals and so on. Therefore, the probability that the arriving packet becomes informative when the system is in a state 0 is $\pi_0 \gamma_9 + \pi_0 \gamma_4 \gamma_6 + \pi_0 \gamma_4 \gamma_7 \gamma_6 + \pi_0 \gamma_4 \gamma_7 \gamma_6 + \cdots = \pi_0 \gamma_9 + \pi_0 \gamma_4 \gamma_6 \gamma_7$. Similarly, we can find the probabilities that the arriving packet becomes an informative packet for all other states and sum it up to get the final equation (7).

$$P_{del} = \pi_0 \gamma_9 + \pi_0 \gamma_4 \gamma_6 \gamma_7 + \pi_1 \gamma_6 \gamma_7 + \pi_1 \gamma_7 \gamma_6 + \pi_2 \gamma_7 \gamma_6 + \pi_2 \gamma_5 \gamma_7 + \pi_3 \gamma_7 + \pi_4 \gamma_6 + \pi_4 \gamma_5 + \pi_5 \gamma_5 + \pi_6 \gamma_5 + \pi_7 \gamma_5$$

2) Packet dropping probability: Packet drop occurs only when both servers are busy. Therefore,

$$P_{drop} = \pi_3 + \pi_4 + \pi_7 + \pi_8$$

3) Probability of obsolete packets:

$$P_{dis} = 1 - P_{del} - P_{drop}$$

4) Average PAoI: From Lemma 2, we can derive the average PAoI as follows.

$$\Delta_{PAoI} = (\mu_1(v_{10} + v_{80}) + \mu_2(v_{20} + v_{70}) + (\mu_1 + \mu_2)(v_{30} + v_{40})) / (\mu_1 + \pi_8 + \mu_2 + \pi_7 + (\mu_1 + \mu_2)(\pi_3 + \pi_4))$$

(VI. FCFS QUEUING SYSTEM USING THE POPMAN APPROACH)

Figure 4 represents the Markov chain for FCFS using the POPMAN approach queuing system. Unlike FCFS queuing system, the Markov chain’s state space is represented by the set $Q = \{0, 1, \ldots, 4\}$ with each state described in Table II. The POPMAN approach ensures that no discard packet is stored in the server, and as soon as it becomes obsolete, it is promptly removed. For instance, when the system is in state 3 if packet in server 2 completed service earlier, system transitions to state 0 rather than to state 5 in a traditional queuing system.

The continuous age evolution is $x(t) = [x_0(t), x_1(t), x_2(t)]$, where $x_0(t), x_1(t)$ and $x_2(t)$ are same as described in FCFS queuing system. In discrete state $q(t) = q$, the continuous state evolves according to equation (11) for the set $Q$.

$$\dot{x}(t) = b_q = \left\{ \begin{array}{ll} 1 & 0 \ 1 & 0 \ 1 & 1 \end{array} \right\}, \quad q = 0 \ n \ q = 1 \ n \ q = 2 \ n \ q = 3, 4$$

To calculate the stationary probability vector $\pi$, we use equations (1) and (2). By utilizing (1) and the state transitions outlined in Table V, it is possible to demonstrate that the
substituting the stationary probabilities into equation (3), and other relevant metrics can be derived as follows.

The average AoI of source 1 can be obtained by evaluating the values of $\pi_q$, $\forall q \in Q$ using Table V, equation (11), substituting the stationary probabilities into equation (3), and solving the resulting linear equations. Finally, by replacing the values of $\pi_q$, $\forall q \in Q$ in equation (4), we can calculate the average AoI of source 1. In addition to the AoI, PAoI and other relevant metrics can be derived as follows.

\begin{align}
P_{\text{del}} &= \pi_0 \gamma_9 + \pi_0 \gamma_4 \gamma_6 \gamma_10 + \pi_1 \gamma_6 \gamma_10 \nonumber \\
&\quad + \pi_1 \gamma_10 (\gamma_9 \gamma_5 + \gamma_4 \gamma_7 \gamma_6 \gamma_10) + \pi_2 \gamma_5 \gamma_10 

&\quad + \pi_2 \gamma_10 (\gamma_8 \gamma_6 + \gamma_3 \gamma_5 \gamma_6 \gamma_10) 

P_{\text{drop}} &= \pi_3 + \pi_4 

P_{\text{dis}} &= 1 - P_{\text{del}} - P_{\text{drop}}
\end{align}
4) Average P AoI: From Lemma 2, we can derive the average PAoI as follows.
\[
\Delta_{P\text{-AoI}} = \frac{\mu_1 \bar{v}_{10} + \mu_2 \bar{v}_{20} + \lambda_1 \bar{v}_{30} + \lambda_2 \bar{v}_{40}}{\mu_1 \pi_1 + \mu_2 \pi_2 + \lambda_1 \pi_3 + \lambda_2 \pi_4} \tag{15}
\]

VII. FCFS WITH PROBABILISTIC ROUTING

In FCFS with probabilistic routing, packets are assigned to servers with arbitrary probabilities \( p_1 \) and \( p_2 \), with the sum of probabilities equaling 1. If the server is busy, packets are dropped. Packets are sent to the monitor after service completion. This assignment model, although not widely utilized, provides numerous possibilities. For example, if we consider the conditions \( \lambda_1 = p_1 \lambda \) and \( \lambda_2 = \lambda (1 - p_1) \) for server 1 and server 2 respectively, these rates are independent of the servers’ current state. This independence grants us the flexibility to align \( \lambda_1 \) and \( \lambda_2 \) with the desired service rates \( \mu_1 \) and \( \mu_2 \) in any manner we choose. Figure 5 represents the Markov chain for FCFS with probabilistic routing queueing system. The Markov chain’s state space is represented by the set \( Q = \{0, 1, ..., 8\} \) with each state described in Table II. The continuous age evolution is \( x(t) = [x_0(t) \ x_1(t) \ x_2(t)] \), where \( x_0(t), x_1(t) \) and \( x_2(t) \) are same as described in FCFS queueing system. In discrete state \( q(t) = q \), the continuous state evolves according to equation (6) for the set \( Q \). To calculate the stationary probability vector \( \pi \), we use equations (1) and (2). By utilizing (1) and the state transitions outlined in Table VI, it is possible to demonstrate that the stationary probability vector \( \pi \) fulfills the equation \( \pi D = \pi Q \). Using \( \pi D = \pi Q \) and equation (2), the stationary probabilities can be determined.

\[
D = \text{diag}[\lambda, \lambda + \mu_1, \lambda + \mu_2, \lambda + \mu_1 + \mu_2, \lambda + \mu_1 + \mu_2, \lambda + \mu_1 + \mu_2, \lambda + \mu_1 + \mu_2, \lambda + \mu_1 + \mu_2] \\
Q = \\
\begin{bmatrix}
0 & p_1 \lambda & p_2 \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\
p_1 \lambda & 0 & p_2 \lambda & 0 & 0 & 0 & 0 & 0 & 0 \\
p_2 \lambda & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

The average AoI of source 1 can be obtained by evaluating the values of \( \pi \), \( \forall q \in Q \) using Table VI, equation (6), substituting the stationary probabilities into equation (3), and solving the resulting linear equations. Finally, by replacing the values of \( \pi \), \( \forall q \in Q \) in equation (4), we can calculate the average AoI of source 1. In addition to the AoI, PAoI and other relevant metrics can be derived as follows.

1) Packet dropping probability: The arriving packet drops at server 1 when it reaches server 1 with probability \( p_1 \) and server 1 is occupied. With probability \( p_1 \) arriving packets goes to server 1, and server 1 is full in states 1,3,4,5,7 and 8. Therefore, the arriving packet drops at server 1 with probability \( p_1(\pi_1 + \pi_3 + \pi_4 + \pi_5 + \pi_7 + \pi_8) \). Similarly the arriving packet drops at server 2 with probability \( p_2(\pi_2 + \pi_3 + \pi_4 + \pi_6 + \pi_7 + \pi_8) \).

\[
P_{\text{drop}} = p_1(\pi_1 + \pi_3 + \pi_4 + \pi_5 + \pi_7 + \pi_8) + p_2(\pi_2 + \pi_3 + \pi_4 + \pi_6 + \pi_7 + \pi_8) \tag{16}
\]

2) Probability of obsolete packets: The arriving packet becomes obsolete only when there is a chance to occupy a server, and a new packet should come and complete its service earlier. For example, when the system is in state 0, and if the arriving packet to server 1 becomes obsolete after two new arrivals. The first new arrival drops at server 1. The second new arrival occupies server 2 and complete its service earlier. \( \pi_0p_1p_2\gamma_3\gamma_6 \) represents the probability that the arriving packet to server 1 in state 0 becomes obsolete after a new arrival in server 2 and complete its service earlier than the packet in the server 1. \( \pi_0p_1p_2\gamma_3\gamma_6 \) represents the probability that the arriving packet to server 1 becomes obsolete after two new arrivals. The first new arrival drops at server 1. The second new arrival occupies server 2 and complete its service earlier. \( \pi_0p_1p_2\gamma_3\gamma_6 \) represents the probability that the arriving packet to server 1 becomes obsolete after three new arrivals. The first new arrival drops at server 1. The second new arrival occupies server 2, where at this stage, both servers are occupied and the third new
arrival may drop at server 1 or server 2. Finally, the packet in server 2 completes its service earlier than the packet in server 1. \(\pi_0 p_1 (p_1 \gamma_3)^2 p_2 \gamma_3 \gamma_6\) represents the probability that the arriving packet to server 1 becomes obsolete after three new arrivals. The first new arrival drops at server 1. The second new arrival again drops at server 1 and the third new arrival occupies server 2 and completes its service earlier than the packet in server 1. \(\pi_0 p_1 p_2 \gamma_3 \gamma_7 \gamma_6\) represents the probability that the arriving packet to server 1 becomes obsolete after three new arrivals. The first new arrival occupies server 2, where at this stage, both servers are occupied. The second new arrival drops at server 1 or server 2 and third new arrival drops at server 1 or server 2 and finally the packet in server 2 completes its service earlier than the packet in server 1 and so on. To get the first term in equation (17) we have to add \(= \pi_0 p_1 p_2 \gamma_3 \gamma_6 (1 + p_1 \gamma_3 + \gamma_7 + p_1 \gamma_3 \gamma_7 + (p_1 \gamma_3)^2 + \gamma_7^2 + \cdots) = \pi_0 p_1 p_2 \gamma_3 \gamma_6 (1 + p_1 \gamma_3 + (p_1 \gamma_3)^2 + \cdots) (1 + \gamma_7 + \gamma_7^2 + \cdots) = \pi_0 p_1 p_2 \gamma_3 \gamma_7 \gamma_6\). Similarly, we can get other terms and add up results in equation (17).

\[
P_{\text{del}} = 1 - P_{\text{drop}} - P_{\text{dis}}
\]

4) Average PAoI: From Lemma 2, we can derive the average PAoI as follows.

\[
\bar{\Delta}_{\text{PAoI}} = (\mu_1 (\bar{v}_{10} + \bar{v}_{01}) + \mu_2 (\bar{v}_{20} + \bar{v}_{01}) + (\mu_1 + \mu_2)(\bar{v}_{30} + \bar{v}_{01}))
\]

\[
/ (\mu_1 (\pi_1 + \pi_8) + \mu_2 (\pi_2 + \pi_7) + (\mu_1 + \mu_2)(\pi_3 + \pi_4))
\]

VIII. FCFS WITH PROBABILISTIC ROUTING USING THE POPMAN APPROACH

\[A = \begin{bmatrix}
p_1 \lambda & p_2 \lambda & p_1 \lambda & p_2 \lambda \\
p_2 \lambda & p_1 \lambda & p_2 \lambda & p_1 \lambda \\
p_1 \lambda & p_2 \lambda & p_1 \lambda & p_2 \lambda \\
p_2 \lambda & p_1 \lambda & p_2 \lambda & p_1 \lambda \\
\end{bmatrix} \]

\[Q = \begin{bmatrix}
p_1 \lambda & p_2 \lambda & 0 & 0 \\
p_1 \lambda & p_2 \lambda & 0 & 0 \\
p_2 \lambda & 0 & p_2 \lambda & 0 \\
p_2 \lambda & 0 & p_2 \lambda & 0 \\
\end{bmatrix} \]

The average AoI of source 1 can be obtained by evaluating the values of \(\tau_{q_0}\), \(\forall q \in Q\) using Table VII, equation (11), substituting the stationary probabilities into equation (3), and solving the resulting linear equations. Finally, by replacing the values of \(\pi_{q_0}\), \(\forall q \in Q\) in equation (4), we can calculate the average AoI of source 1. In addition to the AoI, PAoI and other relevant metrics can be derived as follows.

1) Packet dropping probability:

\[
P_{\text{drop}} = p_1 (\pi_1 + \pi_3 + \pi_4) + p_2 (\pi_2 + \pi_3 + \pi_4)
\]

2) Probability of obsolete packets:

\[
P_{\text{dis}} = \pi_0 p_1 p_2 \gamma_3 \gamma_7 \gamma_6 + \pi_0 p_2 p_1 \gamma_3 \gamma_7 \gamma_6 + p_1 p_2 \gamma_3 \gamma_7 \gamma_6 + p_2 p_1 \gamma_3 \gamma_7 \gamma_6
\]

3) Probability of informative Packets:

\[
P_{\text{del}} = 1 - P_{\text{drop}} - P_{\text{dis}}
\]
4) Average PAoI: From Lemma 2, we can derive the average PAoI as follows.

\[
\Delta_{P_{\text{AoI}}} = \frac{\mu_1 \bar{v}_{10} + \mu_2 \bar{v}_{20} + (\mu_1 + \mu_2) \bar{v}_{30} + (\mu_1 + \mu_2) \bar{v}_{40}}{\mu_1 \bar{\pi}_1 + \mu_2 \bar{\pi}_2 + (\mu_1 + \mu_2) \bar{\pi}_3 + (\mu_1 + \mu_2) \bar{\pi}_4} \tag{23}
\]

IX. NUMERICAL RESULTS

Fig. 7: A comparison between FCFS and FCFS using the POPMAN approach for different probabilities \((P_{\text{det}}, P_{\text{drop}}\) and \(P_{\text{dis}}\)) in heterogeneous scenario.

Fig. 8: A comparison between FCFS with probabilistic routing and FCFS with probabilistic routing using the POPMAN approach for different probabilities \((P_{\text{det}}, P_{\text{drop}}\) and \(P_{\text{dis}}\)) in heterogeneous scenario.

Fig. 9: A comparison between FCFS in single-server and FCFS in two-server homogeneous scenarios for different probabilities \((P_{\text{det}}, P_{\text{drop}}\) and \(P_{\text{dis}}\)).

Fig. 10: A comparison between FCFS and FCFS using the POPMAN approach for average AoI in heterogeneous and homogeneous scenarios.

In Figure 9, at an arrival rate of \(\lambda = 9.8\), the two-server scheme shows a 49.15% increase in the probability of informative packets compared to the single-server scheme under the traditional FCFS queueing system. This improvement is attributed to the parallel serving of packets by the additional server and their efficient transmission to the destination.

In Figure 10, all queueing disciplines exhibit similar AoI performance at low arrival rates because servers are mostly available to handle incoming packets and finish their service. However, as the arrival rate increases, the performance varies for each queueing discipline due to disparities in packet assignment to servers. This reasoning applies to all the performance results, which include the average AoI and PAoI that have been depicted. For instance, at an arrival rate of \(\lambda = 9.8\), compared to FCFS with a single server, the AoI decreases by 37.056%, 43.28%, 58.54%, and 62.14% for FCFS and FCFS using the POPMAN approach in a homogeneous scenario, FCFS and FCFS using the POPMAN approach in a heterogeneous scenario (for \(\mu_1 = 1\) and \(\mu_2 = 2\), respectively.

In Figure 11, at intermediate arrival rates, FCFS with probabilistic routing experiences a significant degradation in performance compared to traditional FCFS schemes in both heterogeneous and homogeneous scenarios. For instance, in FCFS with probabilistic routing in homogeneous scenario...
at $p_1 = 0.1$, the server 1 is underutilized despite both servers having equal capacity to handle incoming packets. The performance analysis of AoI improves gradually from $p_1 = 0$ to $p_1 = 0.5$ and decreases from $p_1 = 0.5$ to $p_1 = 1$. However, in heterogeneous scenario, $p_1 = 0.5$ is not the optimum splitting probability as the server with a higher service rate remains underutilized. The optimal splitting probabilities, $p_1$, depend on the arrival rate and the service rates of the heterogeneous servers. Figure 17 and Figure 18 shows the optimal Splitting probabilities, $p_1$, for traditional FCFS with probabilistic routing and FCFS with probabilistic routing using the POPMAN approach. It can be observed that the optimal splitting probability, $p_1$, is relatively higher in the POPMAN scheme. Nevertheless, at higher rates, FCFS with probabilistic routing approaches the performance level of traditional FCFS schemes. This is because as the arrival rate increases servers are efficiently used for transmitting the packets to the destination. For example, at an arrival rate of $\lambda = 9.8$, there is a $12.5\%$, $1.8\%$, $3.19\%$, and $1.72\%$ increase in AoI for FCFS with probabilistic routing at $p_1 = 0.1$ and at $p_1 = 0.5$ in the homogeneous scenario, FCFS with probabilistic routing at $p_1 = 0.5$ and at $p_1 = 0.3$ in the heterogeneous scenario, respectively, compared to traditional FCFS schemes.

Similarly, in Figure 12, at an arrival rate of $\lambda = 9.8$, there is an improvement of $7.07\%$, $9.52\%$, $7.42\%$, and $6.57\%$ for FCFS with probabilistic routing using the POPMAN approach at $p_1 = 0.1$ and at $p_1 = 0.5$ in the homogeneous scenario, and FCFS with probabilistic routing using the POPMAN approach at $p_1 = 0.5$ and at $p_1 = 0.3$ in the heterogeneous scenario, respectively compared to FCFS with probabilistic routing queueing system.

In Figure 13, it is observed that the service rate of each server in the parallel system is approximately 0.625 times lower than that of the single server system to achieve similar AoI performance. This information can be used to determine the number of parallel servers required to achieve similar average AoI performance as the single server system. For example, 8 parallel servers with a service rate of $\mu = 1$ can replace a single server system with a service rate of $\mu = 4.096$.

In Figure 14, the results show that the splitting of the service rate of the server increases, the performance in terms of AoI degrades. In Figure 15, compare to the PAoI for the single server system, at arrival rate $\lambda = 9.8$ PAoI decreases by $55.4\%$, $60.06\%$, $53.7\%$, and $57.53\%$ for the FCFS, FCFS using the POPMAN approach, FCFS with probabilistic routing, and FCFS with probabilistic routing using the POPMAN approach queueing systems, respectively.
Optimum Splitting Probability $p$

FCFS with probabilistic routing identifies and discards outdated packets opens up new directions for research in the field of AoI.

Multiple servers. We believe that our proactive approach to extend our analysis to encompass multiple sources and demonstrate that this proposed scheme enhances performance to handle incoming packets more efficiently. Our analysis enhances the utilization of server resources, enabling them to manage incoming packets more efficiently. Our analysis demonstrates that this proposed scheme enhances performance in all scenarios. Our proposed POPMAN method can also be applied to LCFS queueing systems. Additionally, we aim to extend our analysis to encompass multiple sources and multiple servers. We believe that our proactive approach to identifying and discarding outdated packets opens up new directions for research in the field of AoI.

X. CONCLUSION

Our research is focused on comprehending the control and optimization of information freshness in two parallel heterogeneous servers. We have analyzed various metrics, such as the average AoI and average PAoI, in different queueing systems with Poisson arrivals and exponential service times. These systems include FCFS, FCFS using the POPMAN approach, FCFS with probabilistic routing, and FCFS with probabilistic routing using the POPMAN approach. Our proposed POPMAN technique involves utilizing a centralized network to proactively detect and eliminate outdated packets from the servers before transmitting them to the monitor. This approach enhances the utilization of server resources, enabling them to handle incoming packets more efficiently. Our analysis demonstrates that this proposed scheme enhances performance in all scenarios. Our proposed POPMAN method can also be applied to LCFS queueing systems. Additionally, we aim to extend our analysis to encompass multiple sources and multiple servers. We believe that our proactive approach to identifying and discarding outdated packets opens up new directions for research in the field of AoI.

REFERENCES


Y. Arun Kumar Reddy received B.Tech degree in Electronics and Communication Engineering from VNRVJET, India in 2008, M.Tech degree in Telecommunication systems Engineering from IIT Kharagpur, India, in 2011 and working as an Assistant professor in RUKT RK Valley University, India. Currently he is pursuing Ph.D in the Department of Electrical Engineering, Indian Institute of Technology, Madras; his area of work and specialization is concentrated in Wireless Networks.

T. G. Venkatesh received the B.E degree in electronics and instrumentation engineering from Annamalai University, India in 1986, the M.E degree in applied electronics from Bharathiar University, Coimbatore, India, in 1988, and the Ph.D degree from the Indian Institute of Science, Bangalore, India, in 1993. He was a Scientific Officer at ISIC, while earning his Ph.D. For a brief duration, he was with the Centre for Development of Telematics, and Indian Space Research Organization, Bangalore. From 1994 to 1999, he was a faculty member with the Indian Institute of Technology, Delhi, India. He is currently a faculty member with the Electrical Engineering Department, Indian Institute of Technology, Madras, Chennai, India. His research group currently focuses on design and performance evaluation of medium access layer protocol in wireless networks, and multicore architecture. He has authored a book on Developing Multimedia Applications With the Java Media Framework and co-authored the book Computer Systems Design and Architecture, published by Pearson.