

A Tandem Queue Model for Two-Server Resequencing System*

Yu Liu, *yuliu@tele.pitt.edu*

School of Information Science, University of Pittsburgh, Pittsburgh, PA 15260, USA

Zhisheng Niu, Xiaokang Lin, *{niu zhs, lin xk}@tsinghua.edu.cn*

Department of Electronic Engineering, Tsinghua University, Beijing, China

Abstract

A two-server resequencing system has two heterogeneous servers and two buffers. One is an arrival buffer, which holds incoming jobs waiting for service. The other is a resequencing buffer, which resequences served jobs back to their arrival orders before departure. Such a system can be modeled as a two-stage tandem queue where the jobs are always kept in the arrival orders but the servers swap the positions upon any job departure. With the assumption of phase-type distributions for job interarrival and service time, the Markovian properties of the state transition is preserved in the model. The stationary probability distribution and the stationary conditions are obtained by the matrix-geometric solution approach. The performance is evaluated in five numerical experiments. The main conclusions are: a small resequencing buffer reduces the mean delay of the jobs significantly; the burstness of arrival traffic has positive influence on the mean delay; a heterogeneous system with more-balanced servers achieves better performance; and the slow server in a less-balanced system has negative effect which might eventually “starve” the fast server under heavy traffic load.

Keywords: tandem queue, resequencing, matrix-geometric solution, phase-type distribution, PH/PH/2, mean-value analysis

1 Introduction

The economy of scale tells us that more savings can be achieved in a system with one large server rather than multiple small servers if costs are proportional to service rates. However, in the real

*This paper is an English extension of [12]

world, multiple small servers are used frequently as an alternative to a large server due to their availability and/or extremely low costs. In a multiple-server system, it is often required to have departure orders of a stream of jobs follow their arrival orders. A common approach is to use a resequencing buffer to reorder the served jobs. This resequencing constraint brings additional delay to the jobs in the system. Understanding the tradeoff between performance degradation due to resequencing and the cost saved by multiple small servers is important for the design of such resequencing systems.

The applications of resequencing systems can be found throughout computer and communications areas, such as parallel computing (multiple processors, parallel web servers), distributed storage (Redundant Array of Inexpensive Disks – RAID), and networking (IBM SNA, X.25 Multilink, inverse multiplexing, and multipath routing).

Many works have been published in modeling and analyzing resequencing systems. An excellent survey on the resequencing system is given by Baccelli and Makowski [1], where resequencing problems are categorized and related works are extensively covered. The performance bounds are obtained by the stochastic ordering theory and the ergodic theory. Yum and Ngai [2] analyze an M/M/k/B queue with resequencing constraints. The stationary state probability is found for M/M/k/B first. The conditional probability of a job bypassing its former jobs is derived from the memoryless property of the service time distribution. The final resequencing delay equation can be obtained by aggregating all delays under the conditional probabilities. The resequencing buffer size, hence, is infinite in this paper. Iliadis and Lien [3] propose a queueing model to analyze the delay for the resequencing system with two heterogeneous servers and *threshold-type* job dispatching policy. Jobs are assigned to the slow server only when the length of the arrival buffer reaches a certain threshold. The results in the paper are based on two unbalanced servers with service rates at 56 and 19.2 packet/second respectively. Such heterogeneous two-server systems can not have as good performance as their bandwidth-equivalent homogeneous systems.

Lien also proposes a Markovian state space description for an M/M/2/B queue with resequencing. This model is also used by Varma in [4] and solved by the matrix-geometric solution approach [5]. In their model, the resequencing buffer is infinite and the arrival buffer is limited at B. Both arrival and resequencing buffers are considered together in the state space of the model. However, each state has three dimensions, representing server occupations, job in-sequence states

and buffer occupations respectively. Such state space description is hard to scale and its application to more complicated system is greatly limited. The results are also short of more general conclusions about the performance of resequencing systems.

In contrast, this paper presents a novel tandem queue model to formulate the same resequencing system. Compared to Varma's work, our results of using this new model provide a stationary condition and disclose the logical properties of the resequencing system.

A similar fork-join system is analyzed by Towsley et al. [6]. The system is modeled as a Markov chain and its approximated response time of three different scheduling algorithms are compared. Jean-Marie and Gün [7] provide some stochastic analysis for resequencing systems with multiple parallel queues where the jobs are dispatched to multiple queues. This multiple-queue system is different from other resequencing systems we mentioned earlier where jobs are dispatched to multiple servers after waiting in a common buffer. From the economy of scale, the performance of a common-queue multiple-server system is better than that of a multiple-queue system. Recent related publications also include an approximation solution by Bilgen and Altintas [8] and a job dispatching method by Gogate and Panwar [9].

This paper presents a new tandem queue model for a two-server resequencing system which has two heterogenous servers and two buffers. The arrival buffer is used for queueing the incoming jobs. The resequencing buffer reorders the served jobs back to their arrival orders before departure. This system is different from the above $M/M/2/B$ with resequencing system because of the sizes of the two buffers. The model, moreover, is also different from other models in literature. The occupations of the arrival and resequencing buffers and the orders of the jobs in the servers are used in the state space representation in the tandem queue model. It preserves the Markovian property of the state transitions and embeds resequencing constraints within the structure of the tandem queues. All the jobs in the model follow their arrival orders all the time. In order to keep such ordering without loss of the Markovian property of the state, the labels of two servers are swapped upon job departure. With the assumption of the phase type distributions for the interarrival and service time of jobs, the stationary state probability and the stationary conditions of the model are found by the matrix-geometric solution approach. Five numerical experiments are performed and metrics such as mean delay, maximum system throughput and server utilization are compared.

The paper is organized as follows. Section 2 introduces the two-server resequencing system and

its tandem queue model. Section 3 describes state transitions and formulates the model, its matrix geometric solution, performance metrics and stationary conditions. Section 4 presents the results of numerical experiments and their analysis. The conclusions are in Section 5.

2 Resequencing system and its tandem queue model

The system includes two heterogeneous servers providing service for a stream of jobs as shown in Figure 1. The system has an infinite arrival buffer Q1 before two servers and a resequencing buffer Q2 with limited size of n jobs. A job in Q1 will be dispatched to an idle server. When both servers are idle, it will be sent to the fast server. After the service, all jobs will be sent to the resequencing buffer Q2 according to their arrival orders. A job can leave Q2 as soon as there is no job arriving earlier than it in the system. A filled Q2 will block the further access to one empty server and the system will wait for the other busy server to finish. This blocking scenario will be explained in the following model. Without loss of generality, the service rates are fixed, and given as a_1 and a_2 for the fast and slow servers respectively, where $a_1 + a_2 = 1$ and $a_1 \geq a_2$. For simplicity, we also use a_1 and a_2 to denote these two servers.

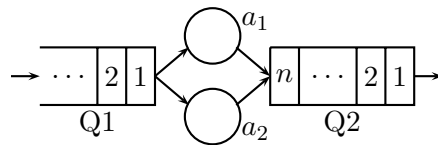


Figure 1: The two server resequencing system

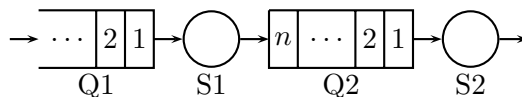


Figure 2: The tandem queue model

This resequencing system can be modeled by a two-stage tandem queue. It is an extension of A. B. Clark’s tandem queueing model introduced in §5.3 of [5]. In Clark’s model, servers are placed in series with queues separating them. “The novel feature is that a busy service unit prevents the access of the new customers to servers further down the line.”

In our tandem queue model in Figure 2, two queues represent the arrival and resequencing buffers Q1 and Q2 respectively. The jobs in the model always follow their arrival orders. The

two servers are labeled according to the orders of their jobs. S2 always has the earliest job in the system. S1 is serving a job which arrives earlier than any other jobs in Q1 but later than any jobs in Q2 and S2.

The service principles here are different from those in Clark’s tandem queue model. They are modified to keep the jobs following their arrival orders while the server labels are swapped upon job departure. When there is a job leaving S1, the job will enter Q2 waiting for the job in S2 to finish and the first waiting job in Q1 then enters S1. When a job leaves S2, all the jobs in Q2 will leave the system simultaneously. Meanwhile, the job in S1 will be moved to S2 and the service rate of S1 will be swapped with the rate of S2. In this way, we always keep S2 serving the earliest job. If there is no job in S2, S2 will keep the higher service rate a_1 . Consequently, a new arrival job will enter the fast server when both servers are idle. When Q2 is full, the job finished by S1 will wait before entering Q2. This blocks other jobs in Q1 from entering S1, until the job in S2 is finished, Q2 is empty and S1 is idle. These service principles guarantee that all jobs always follow their arrival orders in the model.

The service rates of S1 and S2 depend on their represented servers in the system where the jobs are currently being served. For homogenous servers, the server label exchange is enough to keep the Markovian property of the state transitions due to the memoryless property of the identical service time distribution. It results in a smaller state space and is easier to scale [10, 12]. For heterogeneous servers in this paper, we exchange the service rates following the label swaps to guarantee the Markovian property of the state transitions. In this model, a job being served in the system is never interrupted. This continuously serving period might include one label switch of its server from S1 to S2 when the earliest job departs. A nice consequence of this model is that the Markovian property is kept and the job arrival orders are embedded in the tandem queue.

| State | Q1 | S1 | Q2 | S2 | Explanation |
|-------|-----|----|-----|----|------------------------------------------------------------|
| 1 | EDC | B | | A | First two jobs are served. S2 is the fast server a_1 . |
| 2a | ED | C | | B | Job A is left. Job C enter a_1 which is relabeled as S1. |
| 2b | ED | C | B | A | Job B is served and enters Q2. Job C enters $S1=a_2$. |
| 5b | F | E | DCB | A | Job E blocks S1 when it is served and Q2 is full. |

Table 1: An example of the tandem queue model with $n = 3$

An example is given in Table 1. In state 1, five jobs are in the system. The first job A is in the

fast server a_1 labeled as S2. In state 2a, Job A is served and leaves the system. Job C enters the fast server. In order to keep the orders, the slow server a_2 is relabeled as S2 which is still serving job B. The fast server a_1 is relabeled as S1 at the same time. In the other possible state 2b, Job B is served before Job A and enters Q2 waiting for departure. Job C enters the slow server a_2 which is still S1. If Job A is still serving in S2 while Q2 is full as shown in state 5b, the finished job E will block further service in S1.

3 Matrix geometric solution and performance metrics

With the assumption that job arrival and service intervals follow the *phase-type* distributions, the two server resequencing system is a PH/PH/2 queue with an additional resequencing buffer. “PH” represents the phase type distribution introduced in Chapter 2 of [5]. It is a versatile class of probability distribution which utilize memoryless distributions, such as exponential or geometric distributions, to approximate other general distributions. It has been proved that the phase type distribution can approximate any general distributions if the state space is large enough. The most used exponential distribution, noted as “M” in Kendall’s queueing system representation, is a trivial case of the phase type distribution, “PH”.

The service length of arrival jobs follows a r_2 -order phase type distribution (β, S, S^0) . These jobs arrive at the system as a phase-type Markov renewal process with the interarrival time following a r_1 -order phase type distribution (α, T, T^0) . The service rates for the fast and slow servers are a_1 and a_2 respectively and $a_1 + a_2 = 1$, where $a_1 \geq a_2$. From the phase-type job service length distribution, the service time of the jobs in two servers will follow two phase type distributions represented by (β, S_1, S_1^0) and (β, S_2, S_2^0) respectively, where $S_1 = a_1 S$, $S_1^0 = a_1 S^0$, $S_2 = a_2 S$ and $S_2^0 = a_2 S^0$.

The tandem queueing model with heterogeneous servers is studied as a Quasi-Birth-Death (QBD) process with the state space

$$\begin{aligned}
 E &= \{(0, j, k, h) : 0 \leq j \leq 2n + 4; 1 \leq k \leq r_1; 1 \leq h \leq r_2\} \\
 &\cup \{(i, j, k, h) : 0 < i; 1 \leq j \leq 2n + 4; 1 \leq k \leq r_1; 1 \leq h \leq r_2\},
 \end{aligned}$$

where k and h represent the phases of the arrival PH-type Markov renewal process of order r_1 and

the service time PH-type distribution of order r_2 respectively. Index i denotes the number of jobs in S1 and Q1. Index j provides information on the server labels and the number of jobs after S1 (in S2, Q2 and blocked in S1). When $1 \leq j \leq n+2$, S2 represents the fast server a_1 and the number of jobs after S1 is j . When $n+3 \leq j \leq 2n+4$, S2 represents the slow server a_2 and the number of jobs after S1 is $j-n-2$.

The infinitesimal generator matrix Q is given by:

$$Q = \begin{matrix} & i \\ & 0 \\ & 1 \\ & 2 \\ & \vdots \end{matrix} \begin{pmatrix} B_0 & C_0 & & 0 \\ B_1 & A_1 & A_0 & \\ & A_2 & A_1 & A_0 \\ 0 & \ddots & \ddots & \ddots \end{pmatrix},$$

where, the block matrices in Q is given by:

$$B_0 = \begin{matrix} & j \\ & 0 \\ & 1 \\ & \vdots \\ & n+2 \\ & n+3 \\ & \vdots \\ & 2n+4 \end{matrix} \left(\begin{array}{c|ccc|ccc} T & T^0 \alpha \otimes \beta & 0 & \dots & \dots & \dots & 0 \\ \hline I \otimes S_1^0 & T \oplus S_1 & & & & & \\ \vdots & \vdots & \ddots & & & & 0 \\ I \otimes S_1^0 & & & T \oplus S_1 & & & \\ \hline I \otimes S_2^0 & & & & T \oplus S_2 & & \\ \vdots & \vdots & 0 & & \ddots & & \\ I \otimes S_2^0 & & & & & T \oplus S_2 & \end{array} \right);$$

$$B_1 = \begin{matrix} & j \\ & 1 \\ & \vdots \\ & n+1 \\ & n+2 \\ & n+3 \\ & \vdots \\ & 2n+3 \\ & 2n+4 \end{matrix} \left(\begin{array}{c|cccc|cccc} 0 & 0 & I \otimes S_2^0 \beta & 0 & I \otimes S_1^0 \beta & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \dots & \vdots \\ 0 & & & 0 & I \otimes S_2^0 \beta & I \otimes S_1^0 \beta & 0 & \dots & 0 \\ 0 & 0 & & & 0 & I \otimes S_1^0 \beta & 0 & \dots & 0 \\ \hline 0 & I \otimes S_2^0 \beta & 0 & \dots & 0 & 0 & I \otimes S_1^0 \beta & & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & & \ddots & \ddots & \vdots \\ 0 & I \otimes S_2^0 \beta & 0 & \dots & 0 & & & 0 & I \otimes S_1^0 \beta \\ 0 & I \otimes S_2^0 \beta & 0 & \dots & 0 & & & & 0 \end{array} \right);$$

$$C_0 = \begin{matrix} j \\ 0 \\ 1 \\ 2 \\ \vdots \\ 2n+4 \end{matrix} \begin{pmatrix} 0 & 0 & \cdots & 0 \\ T^0\alpha \otimes I & & & 0 \\ & T^0\alpha \otimes I & & \\ & & \ddots & \\ 0 & & & T^0\alpha \otimes I \end{pmatrix}; A_0 = \begin{matrix} j \\ 1 \\ \vdots \\ 2n+4 \end{matrix} \begin{pmatrix} T^0\alpha \otimes I & 0 \\ & \ddots \\ 0 & T^0\alpha \otimes I \end{pmatrix};$$

$$A_1 = \begin{matrix} j \\ 1 \\ \vdots \\ n+1 \\ n+2 \\ n+3 \\ \vdots \\ 2n+3 \\ 2n+4 \end{matrix} \left(\begin{array}{ccc|ccc} T \oplus S & & 0 & & & \\ & \ddots & & & & 0 \\ & & T \oplus S & & & \\ 0 & & & T \oplus S_1 & & \\ \hline & & & & T \oplus S & 0 \\ & & & & \ddots & \\ 0 & & & & & T \oplus S \\ & & & 0 & & T \oplus S_2 \end{array} \right);$$

$$A_2 = \begin{matrix} j \\ 1 \\ \vdots \\ n+1 \\ n+2 \\ n+3 \\ \vdots \\ 2n+3 \\ 2n+4 \end{matrix} \left(\begin{array}{cccc|cccc} 0 & I \otimes S_2^0\beta & 0 & & I \otimes S_1^0\beta & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \vdots & \vdots & \cdots & \vdots \\ 0 & & 0 & I \otimes S_2^0\beta & I \otimes S_1^0\beta & 0 & \cdots & 0 \\ 0 & & & 0 & I \otimes S_1^0\beta & 0 & \cdots & 0 \\ \hline I \otimes S_2^0\beta & 0 & \cdots & 0 & 0 & I \otimes S_1^0\beta & & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \\ I \otimes S_2^0\beta & 0 & \cdots & 0 & 0 & & 0 & I \otimes S_1^0\beta \\ I \otimes S_2^0\beta & 0 & \cdots & 0 & 0 & 0 & & 0 \end{array} \right).$$

In the above equations, “ \otimes ” and “ \oplus ” are Kronecker product and Kronecker summation of matrices. Identity matrices I used before or after “ \otimes ” have dimensions of r_1 and r_2 respectively. Empty spaces in the matrices are all zeros.

The first row of Q describes the transitions when Q1 and S1 are empty ($i = 0$). In these states, if S2 is idle, an arrival job will be served in S2 which moves the state from $j = 0$ to $j = 1$ with $T^0\alpha \otimes \beta$ in the first row of B_0 . If S2 is not idle ($j > 0$), an arrival job will be served by S1 which moves the state $i = 0, j > 0$ to $i = 1, j > 0$ as shown in C_0 . When S1 is idle but S2 is busy ($i = 0, j > 0$), the departure of the job in S2 empties Q2 and moves the state to $i = 0, j = 0$ in B_0 .

The second row of Q is about the transitions when there is one job in Q1 and S1 ($i = 1$). The first column of zero matrices in B_1 is for the states $j = 0$. All these zero matrices have the size of $r_1 r_2 \times r_1$. These mean that the system will never go to the state of $j = 0$ in one transition, because any departure in S2 will result the relabeling of S1 and S2 ($i = 0, j > 0$). The other four block matrices bounded by solid lines at the right of B_1 are square matrices with size of $(n + 2) \times r_1 r_2$ each. They represent the state transitions from $i = 1$ to $i = 0$ when a job leaves S2 or S1. In the following, let's consider an example when a job in S2 is done. If S2 is the fast server a_1 and there are n jobs waiting in Q2 ($j = n + 1$), the state will change to $j = n + 3$ by moving the job in S1 to S2 and empty all the jobs in Q2. Likewise, if S2 is the slow server a_2 and there are n jobs in Q2 ($j = 2n + 3$), the state will change to $j = 1$ by moving the job in S1 down to S2. Moreover, for the job finished in S1, the state j will only be increased by one because the job is appended to Q2 waiting for resequencing. When Q2 is full, i.e., $j = n + 2$ or $j = 2n + 4$, the job finished in S1 will block S1 and no further transition is possible as shown in B_1 .

The arrival transitions are represented in A_0 where arrival jobs are simply appended to Q1. The diagonal matrix A_1 mainly represents the phase transitions for the arrival and service phase type distributions. In states $i > 1$, the departure matrix A_2 can be found in a similar fashion as to the B_1 above.

The stationary state probability of the tandem queue model is defined as: $x_{i,j,k,h} = P\{N_1 = i, N_2 = j, K = k, H = h\}$, where N_1 is the number of jobs in Q1 and S1; N_2 is the number of jobs after S1; K and H are the phase states of the arrival process and the service distribution. The stationary state probability vectors are given by:

$$\begin{aligned}
x &= (x_0, x_1, x_2, \dots), \\
x_0 &= (x_{0,0}, x_{0,1}, x_{0,2}, \dots, x_{0,2n+4}), \\
x_i &= (x_{i,1}, x_{i,2}, \dots, x_{i,2n+4}), \quad i \geq 1, \\
x_{i,j} &= (x_{i,j,1,1}, \dots, x_{i,j,1,r_2}, \dots, x_{i,j,r_1,1}, \dots, x_{i,j,r_1,r_2}), \quad (i, j) \in E.
\end{aligned}$$

From equations $xQ = 0$ and $xe = 1$, we have

$$x_i = x_1 R^{i-1}, i \geq 1, \tag{1}$$

where the non-negative rate matrix R is the minimal nonnegative solution to the equation

$$A_0 + RA_1 + R^2A_2 = 0. \quad (2)$$

The boundary vector (x_0, x_1) is obtained by equations

$$(x_0, x_1) \begin{pmatrix} B_0 & C_0 \\ B_1 & A_1 + RA_2 \end{pmatrix} = (0, 0) \quad (3)$$

$$x_0e + x_1(I - R)^{-1}e = 1. \quad (4)$$

From the Little's theorem, the mean delay of the jobs in the system is given by

$$\begin{aligned} \bar{d} &= \bar{\lambda}^{-1} \sum_{(i,j) \in E} (i+j)x_{ij}e \\ &= \bar{\lambda}^{-1} \left[\sum_{j=1}^{n+2} j(x_{0,j}e + x_{0,j+n+2}e) + \sum_{i=1}^{\infty} \sum_{j=1}^{n+2} (i+j)(x_{i,j}e + x_{i,j+n+2}e) \right] \\ &= \bar{\lambda}^{-1} \{x_0\Lambda e + x_1[(I - R)^{-1}\Lambda e + (I - R)^{-2}e]\}, \end{aligned} \quad (5)$$

where Λ is given by

$$\Lambda = \begin{pmatrix} I_{r_1 \times r_2} & & & & 0 \\ & 2I_{r_1 \times r_2} & & & \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & (n+2)I_{r_1 \times r_2} \\ \hline I_{r_1 \times r_2} & & & & 0 \\ & 2I_{r_1 \times r_2} & & & \\ & & \ddots & & \\ 0 & & & & (n+2)I_{r_1 \times r_2} \end{pmatrix}.$$

The server utilization formulas are given by:

$$\eta_1 = 1 - x_{00}e - \sum_{j=n+3}^{2n+4} x_{0j}e \quad (6)$$

$$\eta_2 = 1 - x_{00}e - \sum_{j=1}^{n+2} x_{0j}e \quad (7)$$

We can also draw some conclusions on the stationary conditions.

Theorem 3.1 *The tandem queue model is stable if and only if*

$$\frac{\bar{\lambda}}{\bar{\mu}} < 1 - a_1^{n+3} - a_2^{n+3} = \rho, \quad (8)$$

where a_1 and a_2 are the normalized service rates with $a_1 + a_2 = 1$; $\bar{\mu} = \phi S^0 e$ is the total service rate of the two servers; $\bar{\lambda} = \theta T^0 e$ is the job arrival rate; θ and ϕ are the stationary probability vectors of $T + T^0 \alpha$ and $S + S^0 \beta$ respectively; n is the length of Q_2 ; and ρ is the maximum system throughput.

Because the model has infinite buffer size in Q_1 , in the unstable condition, the length of Q_1 and the mean delay will increase to infinity. For homogenous servers at $a_1 = a_2$, the condition is equivalent to $\frac{\bar{\lambda}}{\bar{\mu}} < 1 - 2^{-n-2} = \rho$. This simplified stationary condition gives the maximum system throughput ρ which forms vertical asymptotic lines for the mean delay curves in Figure 3, 4 and 5 in next section. The proof of the theorem is given in the Appendix.

4 Numerical experiments and analysis

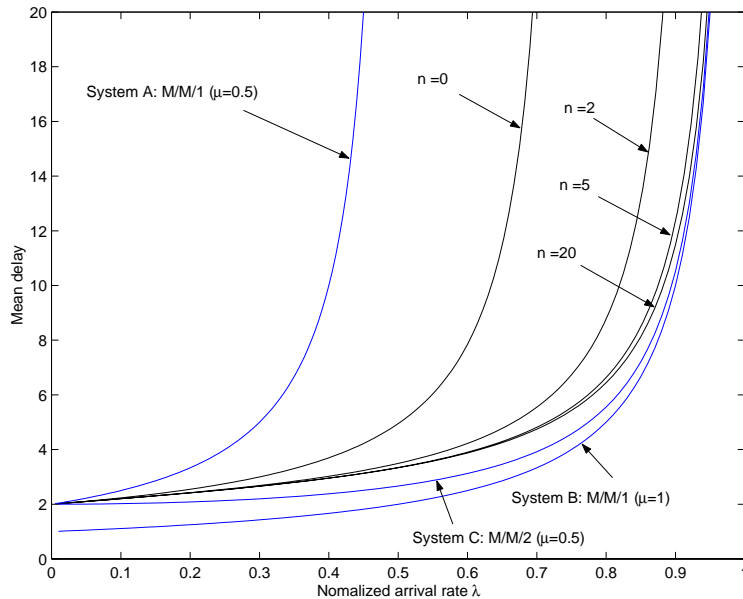


Figure 3: Mean delay \bar{d} versus normalized arrival traffic λ under different resequencing buffer sizes

In Figure 3, we compare the mean delay \bar{d} of $M/M/2$ with resequencing systems to other three general systems – A: $M/M/1$ with service rate $\mu = 1/2$; B: $M/M/1$ with $\mu = 1$; and C: $M/M/2$ with $\mu = 1/2$. The $M/M/2$ queues in this experiment have two homogeneous servers with service rate $\mu = 1/2$ each. The horizontal axis shows different arrival rates and the vertical axis gives the

mean delay \bar{d} . The size of the resequencing buffer n is varied to be 0, 2, 5 and 20. Their maximum system throughput are 0.75, 0.9375, 0.9922, and 0.999997 from (8). They are also the maximum arrival rates λ in the mean delay curves since $\bar{\mu} = 2\mu = 1$. The comparison systems A and C provide the upper and lower bounds of the mean delay in the resequencing systems. When $n = 0$, there is no resequencing buffer. A lot of jobs will be blocked in S1 when S2 is busy. By increasing n from 0 to 5, the mean delay is reduced dramatically. Hence, a small resequencing buffer will significantly reduce \bar{d} . Further increase of n will continuously reduce \bar{d} , but the reduction speed is much slower. When $n > 20$, such reduction is hardly perceivable in the figure. Hence, the mean delay when $n = \infty$ can be well approximated by that at a sufficiently large n . There is always a gap in mean delays between resequencing system and system C (M/M/2) no matter how large n is. This gap comes from the resequencing delay of a constant number of jobs being reordered.

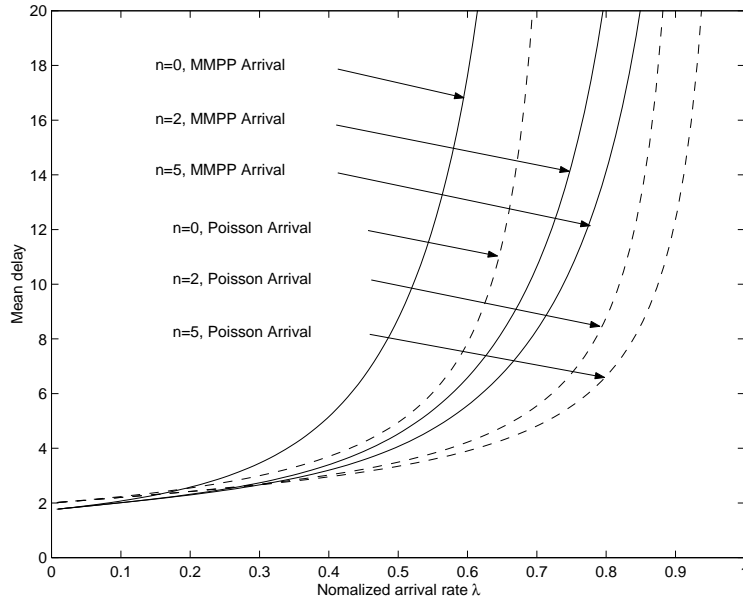


Figure 4: Mean delay \bar{d} versus normalized arrival traffic λ under different resequencing buffer sizes for both MMPP and Poisson arrivals

In the second experiment, the impact of the bursty traffic on the mean delay is analyzed in Figure 4. We use a two-state Markovian modulated Poisson process (MMPP) to simulate a bursty traffic. The parameters of MMPP are

$$\alpha = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, T = \begin{pmatrix} -\lambda_1 - c_1 & c_1 \\ c_2 & -\lambda_2 - c_2 \end{pmatrix}, T^0 = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

where $\lambda_1 = 1.54, \lambda_2 = 0.29, c_1 = 0.074, c_2 = 0.098$ [11]. This MMPP has a squared coefficient

of variation $C_a^2 = 2.25$ and a coefficient of covariation $\theta = 0.2$. The results for both MMPP and Poisson arrival at $n = 0, 2, 5$ are given in Figure 4. It is clearly shown that the MMPP arrival has significantly longer mean delay than the Poisson arrival.

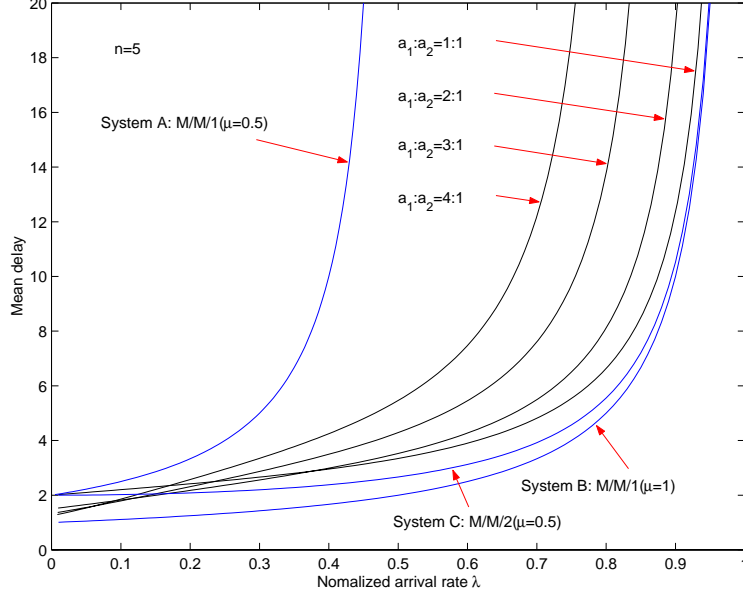


Figure 5: Mean delay \bar{d} versus normalized arrival traffic λ under different service rate ratios when $n = 5$

The third experiment is to analyze mean delays in the resequencing system with heterogeneous servers. In Figure 5, three systems for comparison are the same as the first experiment in Figure 3. Four resequencing systems with $n = 5$ use heterogeneous servers with different service rate ratios at $a_1 : a_2 = 1, 2, 3, 4$ respectively. The higher the ratio is, the longer the mean delay is. The results show that a balanced two server resequencing system achieves the lowest mean delay.

Continuing from heterogenous server experiment, our fourth experiment is on the maximum system throughput calculated from (8). Figure 6 shows the maximum system throughput versus the slow server's service rate a_2 under different n . The maximum system throughput is obtained when two servers are balanced at $a_1 = a_2 = 1/2$. Moreover, the increase of n will not only reduce the mean delay as shown in Figure 3, but also improve the maximum system throughput.

The last experiment is about the server utilization η_1 and η_2 in heterogenous resequencing systems. Figure 7 and Figure 8 show the influence of different service rate ratios on the server utilization in the resequencing systems with $n = 2$ and 20. These results not only provide further supportive results for the above conclusions, but also discover a scenario where the utilization of the

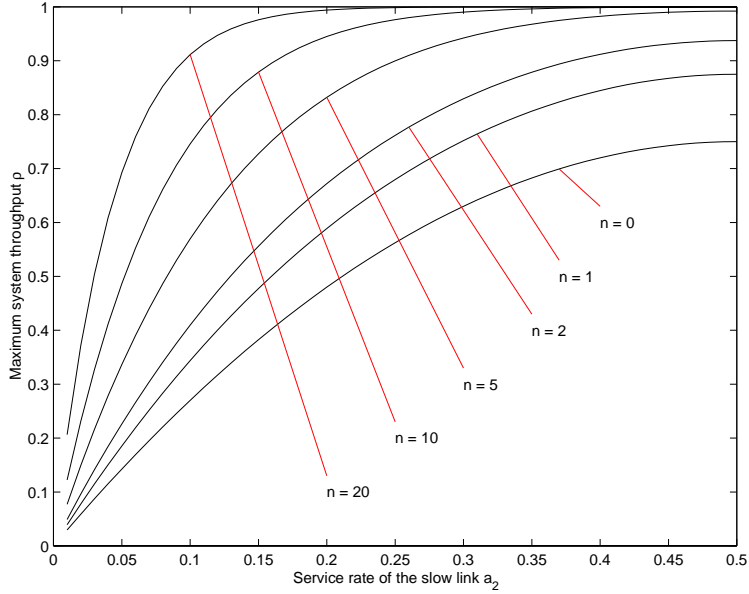


Figure 6: Maximum system throughput ρ versus the service rate of the slow server a_2

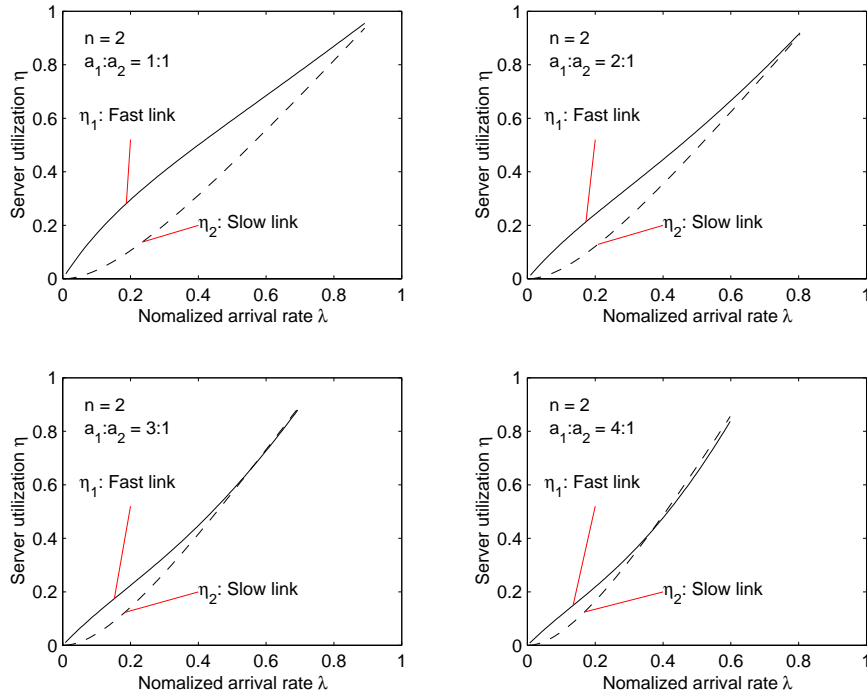


Figure 7: Mean server utilization versus arrival rate λ under different service rate ratios, $n = 2$

fast server is reduced by its slower partner. In both figures, when the service rate ratio increases up to 3, the slow server gets higher utilization under high arrival rate in both systems. This scenario seems to contradict common understanding – when two servers have different service rates, the

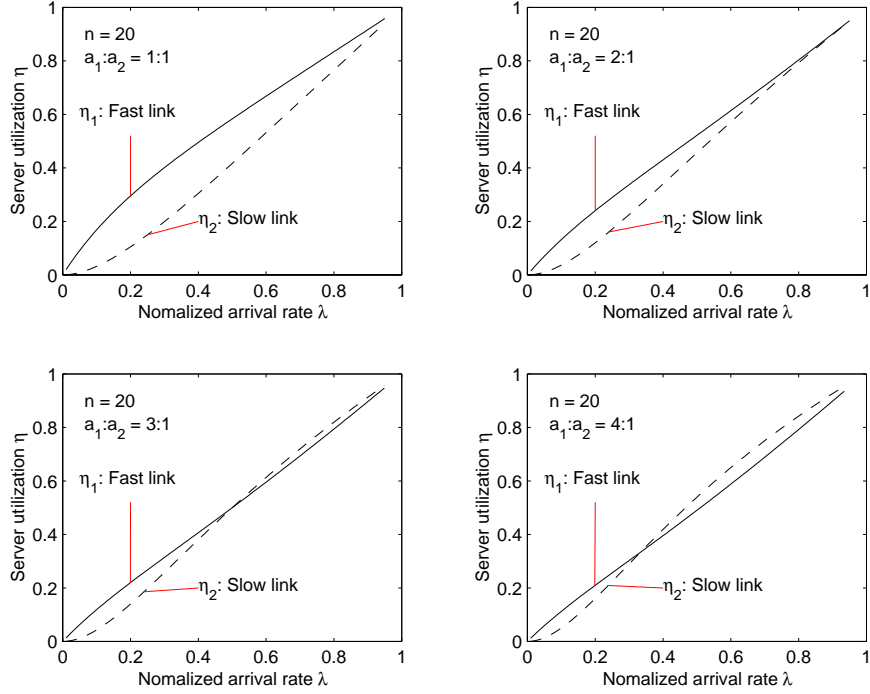


Figure 8: Mean server utilization versus arrival rate λ under different service rate ratios, $n = 20$

fast server should get higher utilization. This scenario can be explained as follows. When the fast server is too fast ($a_1 : a_2 \geq 3$) and the traffic load is high, the slow server will take much longer time to serve an earliest job in the system. This job will block all the other later jobs which have been served in the fast server and waiting in Q2. When the limited resequencing buffer is full, the fast server is blocked and “starved” by its slower partner. This is an important disadvantage of heterogenous resequencing systems.

One approach to solve the above problem is to keep the slow server idle under certain traffic load and resequencing queue occupation. In this way, the job served by the fast server gets minimum resequencing interference from the slow server and can leave the system without waiting in the resequencing buffer. This is similar to the threshold-type job dispatching policy in [3]. However, this solution is based on the case where the service rate ratio is very high ($a_1 : a_2 \geq 3$). It is more realistic to have a balanced multiple server system and it also results in better performance.

5 Conclusion

In this paper, we propose a tandem queue model to analyze the two server resequencing system. The state space and transition matrices are given. Matrix-geometric solution approach is used to find the stationary state probability and the stationary condition of the model, the mean delay of jobs, and the mean server utilization. Based on the results of five numerical experiments, our conclusions about the resequencing system is given in the following.

- The larger the resequencing buffer size, the lower the mean delay and the higher the maximum system throughput.
- The increase of the resequencing buffer size n will be very helpful to reduce the mean delay in the beginning (when the size is small), but these benefits drop down significantly after $n > 20$ and the mean delay converges to its limit at $n = \infty$.
- In heterogeneous resequencing systems, the higher the service rate ratio, the larger the mean delay and the lower the maximum system throughput.
- A balanced two server resequencing system will perform better than a less balanced one.
- In a heterogeneous two-server resequencing system, the slow server might have higher utilization than the fast server. This is a disadvantage because the slow server will block the fast server under heavy traffic load.

Appendix: Proof of Theorem 3.1

From Theorem 1.7.1 in [5], the stationary condition of the tandem queue model satisfies

$$\pi A_0 e < \pi A_2 e, \tag{9}$$

where $\pi = (\pi_1, \pi_2)$, $\pi_k = (\pi_{k,1}, \pi_{k,2}, \dots, \pi_{k,n+2})$, $k = 1, 2$ is the stationary probability vector of the matrix $A = A_0 + A_1 + A_2$ and

$$A = \begin{array}{c} j \\ 1 \\ \vdots \\ \vdots \\ n+1 \\ n+2 \\ n+3 \\ \vdots \\ 2n+3 \\ 2n+4 \end{array} \left(\begin{array}{cccc|cccc} (T + T^0\alpha) \oplus S & I \otimes S_2^0\beta & & 0 & I \otimes S_1^0\beta & 0 & \cdots & 0 \\ & \vdots & \ddots & & \vdots & \vdots & \cdots & \vdots \\ & & & & & & & \\ 0 & & & I \otimes S_2^0\beta & I \otimes S_1^0\beta & 0 & \cdots & 0 \\ 0 & & & (T + T^0\alpha) \oplus S & I \otimes S_1^0\beta & 0 & \cdots & 0 \\ \hline I \otimes S_2^0\beta & 0 & \cdots & 0 & (T + T^0\alpha) \oplus S & I \otimes S_1^0\beta & & 0 \\ & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \\ I \otimes S_2^0\beta & 0 & \cdots & 0 & 0 & & \ddots & I \otimes S_1^0\beta \\ I \otimes S_2^0\beta & 0 & \cdots & 0 & 0 & 0 & & (T + T^0\alpha) \oplus S \end{array} \right)$$

After some derivation, π can be found as:

$$\begin{aligned} \pi_{1,j} &= a_2^j a_1 \theta \otimes \phi, 1 \leq j \leq n+1; \\ \pi_{1,n+2} &= a_2^{n+2} \theta \otimes \phi; \\ \pi_{2,j} &= a_1^j a_2 \theta \otimes \phi, 1 \leq j \leq n+1; \\ \pi_{2,n+2} &= a_1^{n+2} \theta \otimes \phi; \end{aligned} \tag{10}$$

where θ and ϕ are the stationary probability vectors of $T + T^0\alpha$ and $S + S^0\beta$ respectively; n is the length of Q2. Substituting the above (10) into (9) proves Theorem 3.1.

References

- [1] François Baccelli and Armand M. Makowski. Queueing models for systems with synchronization constraints. *Proceedings of IEEE*, 77(1):138–161, January 1989.
- [2] Tak-Shing P. Yum and Tin-Yee Ngai. Resequencing of messages in communication networks. *IEEE Transaction on Communications*, COM-34(2):143–149, February 1986.
- [3] Ilias Iliadis and Luke Y.-C. Lien. Resequencing delay for a queueing system with two heterogeneous servers under a threshold-type scheduling. *IEEE Transaction on Communications*, 36(6):692–702, June 1988.
- [4] S. Varma. A matrix geometric solution to a resequencing problem. *Performance Evaluation*, 12(2):103–114, 1991.

- [5] Marcel F. Neuts. *Matrix-Geometric Solutions in the Stochastic Models: An Algorithmic Approach*. The Johns Hopkins University Press, Baltimore, MD, 1981.
- [6] Don Towsley, C. Gary Rommel, and John A. Stankovic. Analysis of fork-join program response times on multiprocessors. *IEEE Transaction on Parallel and Distributed Systems*, 1(3):286–303, July 1990.
- [7] Alain Jean-Marie and Levent Gün. Parallel queues with resequencing. *Journal of the ACM*, 40(5):1188–1208, 1993.
- [8] Semih Bilgen and Onur Altintas. An approximate solution for the resequencing problem in packet-switching networks. *IEEE Transaction on Communications*, 42(2/3/4):229–232, February/March/April 1994.
- [9] Nitin R. Gogate and Shivendra S. Panwar. Assigning customers to two parallel servers with resequencing. *IEEE Communications Letters*, 3(4):119–121, April 1999.
- [10] Yu Liu. Research on integrated voice and data switching technology. Master’s thesis, , Department of Electronic Engineering, Tsinghua University, Beijing, China, April 1996. (In Chinese).
- [11] Zhisheng Niu, Takehiro Kawai, Yoshiaki Tadokoro, and Haruo Akimaru. A unified solution to mixed loss and delay systems with partial preemptive priority. *Electronics and Communications in Japan, Part 1*, 78(8):57–65, 1995.
- [12] Zhisheng Niu, Yu Liu, Xiaokang Lin. Two-Link Striping System in Packet-Switched Networks. *ACTA Electronica SINICA*, 27(6):83–87, 1999. (In Chinese).