Stability and Instability of a Two-Station Queueing Network: The Exponential Case

J. G. Dai
School of Industrial and Systems Engineering and School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA
dai@isye.gatech.edu

John J. Hasenbein
Graduate Program in Operations Research and Industrial Engineering, Department of Mechanical Engineering, University of Texas at Austin, Austin, TX 78712-1063, USA
jhas@mail.utexas.edu

Abstract

This paper, together with a companion paper [11], proves that the stability region of a 2-station, 5-class re-entrant queueing network, operating under a non-preemptive static buffer priority service policy, depends on the distributions of the interarrival and service times. In particular, our result shows that conditions on the mean interarrival and service times are not enough to determine the stability of a queueing network. In this paper, we prove that when all distributions are exponential, the network is unstable in the sense that, with probability one, the total number of jobs in the network goes to infinity with time. In the companion paper, we show that, among other things, the same network with all interarrival and service times being deterministic is stable.

Keywords: multiclass queueing network, reentrant line, stability, fluid model, virtual station, push start, large deviations estimate

March 8, 2001

1 Introduction

This paper, along with a companion paper [11], is part of an ongoing effort to understand the relationship between the stability of a queueing network and the stability of the corresponding fluid model; see Rybko and Stolyar [26], Dai [8], Stolyar [28], Dai and Meyn [12], Chen [6], Meyn [24], Dai [10], Bramson [4, 5] and Pulhaskii and Rybko [25]. The fluid model is a continuous, deterministic analog of a discrete stochastic queueing network, and is defined through a set of equations. It is known that the stability of a queueing network is implied by the stability of its fluid model; the stability analysis of the latter, though still nontrivial, is often significantly easier than the former; see, for example [1, 3, 7, 14, 19, 22]. Recently, Bramson [5] gave an example of a queueing network that is stable, but whose fluid model is not stable.

The queueing network to be studied in this paper and again in the companion paper [11] has 2 service stations and 5 job classes. It belongs to a special class of networks called re-entrant lines by Kumar [21]. The network is assumed to be operating under a non-preemptive static buffer priority (SBP) service policy, and to have fixed mean interarrival and service times. In this paper, we assume all interarrival and service time distributions are exponential. In Theorem 2.1, we prove that the queueing network is unstable in the sense that, with probability 1, the total number of jobs in the system goes to infinity with time.

1Research supported in part by NSF grants DMI-94-57336 and DMI-9813345
2Research supported in part by NSF grant INT-9971484
In the companion paper [11], we consider a model which is identical to the queueing network studied in this paper, except that all interarrival and service time distributions are assumed to be deterministic. It will be shown that the analogous deterministic network is stable. A consequence of this result and Theorem 2.1 is that the stability region of the 2-station queueing network depends on the distributions, not just the means, of the interarrival and service times. For queueing networks operating under a head-of-line (HOL) service policy, practical fluid models are defined through a set of equations, known as the fluid model equations, which take the mean interarrival and service times as parameters. Hence, a further consequence of our result is that no mean-value based fluid model can determine the stability of the queueing network we study.

For a queueing network operating under a given service policy, each fluid model equation in the corresponding fluid model can be added only when it can be justified by a limiting procedure via fluid limits; see, for example, Section 7 of Dai [8]. Some equations, like the ones balancing the flows among job classes, can be derived and justified easily. Others, in particular those that are specific to a service policy, are more difficult to divine and justify. Generating and verifying such fluid model equations sometimes requires great insight and deep analysis of the queueing network itself; see, for example, the virtual station fluid model equations (14)–(16) in Dai and Vande Vate [14]. Nevertheless, this add hoc way of writing down fluid model equations have been quite successful because it is practical and works well for a number for service policies. For Bramson's example in [5], one may wonder if, by adding additional fluid model equations, a modified fluid model would be stable, thus nullifying the result in [5]. Such a scenario, while unlikely, was not ruled out in Bramson's paper. As noted above, our main result precludes the possibility that adding more fluid equations could ever result in a complete mean-value fluid model for the network we consider.

In Dai and Vande Vate [14], it was shown that 2-station multitype fluid models are globally stable (i.e., stable under any non-idling service policy) if and only if the virtual station and push start conditions are satisfied. (See Section 7 for further discussion of these conditions.) In conjunction with the stability result in Dai [8], this implies that the virtual station and push start conditions are sufficient for global stability of 2-station multitype queueing networks, with general interarrival and service distributions. The next natural question which arises is whether these conditions are also necessary for the stability of such queueing networks. If so, then the fluid model can be used to completely characterize the global stability of this class of queueing networks. A pathwise argument (see, e.g., [13] and [18]) can be used to demonstrate that virtual station conditions are indeed necessary for the global stability of networks with general interarrival and service distributions. The queueing network in this paper and the companion paper provides a test case for determining the necessity of push start conditions. For the exponential queueing network considered in this paper, Dai and Vande Vate's virtual station condition is satisfied. However, the push start condition is violated. Theorem 2.1 of this paper indicates that the push start condition is indeed also necessary for global stability of the exponential queueing network. We believe that it is likely that such a principle should hold for all 2-station multitype queueing networks with exponential distributions, i.e., in such networks the virtual station and push start conditions completely characterize the global stability region. We anticipate that the proof techniques used in the current paper will be of use in establishing a more general result of this type.

Although this paper and the companion paper consider closely related queueing models, they share little in terms of the proof technique. In this paper, we first present a fluid model solution that cycles to infinity. The majority of the paper is then devoted to establishing a series of large deviations type estimates which show that the queueing network dynamics closely follow the unstable fluid model solution with high probability. In the companion paper, we show that the same fluid model has a stable fluid model solution. In that paper, the proof of the main result rests on
a demonstration that, for the deterministic queueing network, the system dynamics closely follow the stable fluid model solution.

To our knowledge, this set of papers is the first to demonstrate that the stability region of a standard multiclass queueing network operating under an HOL service policy may depend on the distributions of interarrival and service times. The first paper which directly showed the gap between the stability of a multiclass queueing network and its fluid model was Bramson [5]. Other researchers have also investigated the relation between the stability of a queueing model and its corresponding fluid model (or the corresponding family of fluid limits). Foss and Kovalevskii [16] consider a polling model and show that the standard definition of stability of a fluid model (see [8], [28]) does not suffice in order to characterize the stability of their stochastic polling model. They present a refined definition of fluid stability which captures the stability behavior of the original system. Stolyar and Ramakrishnan [29] investigate the stability of another type of polling model which falls outside the scope of the standard multiclass queueing network. They also demonstrate that the standard criterion for defining the stability of fluid limits does not properly characterize the stability of the original model. Again, a more refined fluid stability criterion is introduced which characterizes the fluid behavior in a more satisfactory manner. Essentially, both papers propose a more careful examination of the fluid limit model rather than the fluid model. The fluid model is a deterministic, mean-value based model, whereas the fluid limit model considers only the (possibly stochastic) weak limits of the rescaled queue-length process. Although it is possible that investigating fluid limits directly could result in a tight characterization of the stability for a general multiclass network, it is unclear if this approach will be of practical use. The main difficulty is that characterizing the stability of a fluid limit model could be as intractable as characterizing the stability of the original model. Hence, the primary appeal of the fluid model has been the relative simplicity of stability analysis.

The rest of paper is organized as follows. In Section 2, we introduce the 2-station queueing network, and state the main result in Theorem 2.1. In Section 3, we introduce the corresponding fluid model and present an unstable fluid model solution that diverges to infinity. Sections 4 and 5 are devoted to the proof of Theorem 2.1. Section 6 gives a proof outline of a result that is analogous to Theorem 2.1, when the network is operated under the preemptive SBP service policy. In Section 7, we present a more detailed discussion of virtual station and push start conditions. We also discuss some insights that distinguish the results in this paper from those in the companion paper. We end the paper with an appendix in which details of some proofs are presented.

For this set of papers, we would suggest that the more casual reader read Sections 2 and 3 of this paper, skip the proof Sections 4 and 5 and 6, and then move on to Section 7 and to the companion paper.

2 The Queueing Network Model and Main Results

In this section, we first define the queueing network model to be studied in this paper. We then state the main results presented in this paper.

2.1 The queueing network model

In this paper we will only be concerned with the queueing network pictured in Figure 1. The network has 2 service stations, each having a single server. Each job follows the deterministic route indicated in the figure, making a total of 5 visits along the route. Each station may only serve one job at any given time. Jobs that are in service or waiting for the kth step of service are called
class \( k \) jobs. We envision them waiting in buffer \( k \) in front of the station. With a slight abuse of notation, we consider a class \( k \) job that is in service also belongs to buffer \( k \). In this paper, we assume that the service times for class \( k \) jobs are i.i.d. exponential random variables with mean \( m_k \), \( k = 1, \ldots, 5 \). The interarrival times for jobs arriving from the outside are also assumed to be i.i.d. exponential random variables with mean \( 1/\alpha_1 \). Thus, \( \alpha_1 \) is the exogenous arrival rate. We further assume that the sequence of interarrival times and the 5 sequences of service times are mutually independent. Throughout this paper, unless explicitly specified otherwise, we fix the arrival rate and mean service times to be

\[
\alpha_1 = 1, \quad m_1 = 0.4, \quad m_2 = 0.1, \quad m_3 = 0.4, \quad m_4 = 0.1 \quad \text{and} \quad m_5 = 0.4. \quad (1)
\]

When either Station \( A \) or \( B \) completes the service of a job, it must determine which job to pick next for service. A service (or dispatch) policy specifies how each station makes this decision for every possible state of the network. Our network is assumed to be operating under the non-idling static buffer priority (SBP) service policy \( \pi = \{(1,3,4),(5,2)\} \). Under this policy, at Station \( A \) jobs of class 1 have highest priority, class 3 jobs have second highest priority, and class 4 jobs are given lowest priority. At Station \( B \), class 5 jobs have highest priority and class 2 jobs have low priority. We can consider both non-preemptive and preemptive versions of the policy \( \pi \). Under a non-preemptive service policy, once a job is in service, this job must be completed before its server can serve any other jobs. Under a preemptive service policy, a job in service can be preempted by an arriving higher priority job. The preempted job is then served from where it left off when the server completes all higher priority jobs. For concreteness, we primarily consider the non-preemptive service policy in this paper. An analog to our main result actually holds also for a network operating under a preemptive service policy, but we only outline the proof in this case (see Section 6).

We use \( Z_k(t) \) to denote the number of jobs in buffer \( k \) at time \( t \) and \( Z(t) = (Z_1(t), \ldots, Z_5(t)) \) to denote the corresponding vector. We use \( |Z(t)| \) to denote the total number of jobs in the network at time \( t \). For the exponential network the vector \( Z(t) \) completely determines the state of the system in the preemptive case (in the non-preemptive case one must also specify which job class is in service). Namely, if one knows \( Z(t) \) at time \( t \), the future evolution of the network can be determined from time \( t \) on. If service and/or interarrival times are not exponential (e.g., as in
the deterministic network considered in the companion paper), the future evolution of the network cannot be determined from $Z(t)$ alone. One also needs to know the remaining interarrival and service times at time $t$ to completely specify the network state.

\section{Main result}

Our main theorem shows that the number of jobs in the exponential network diverges to infinity.

**Theorem 2.1.** For the exponential network operating under the non preemptive SBP service policy, starting from any initial state,

$$|Z(t)| \to \infty$$

as $t \to \infty$ with probability one.

The proof of Theorem 2.1 will be given in Sections 4 and 5. The main idea is to show that the queueing network dynamics closely follow an unstable fluid model solution. In the next section, we introduce the fluid model which corresponds to the exponential network, and give the fluid model solution which provides the intuitive basis for the instability proof.

\section{A unstable fluid model solution}

As a preliminary to the proof of Theorem 2.1, we now introduce the fluid model of our queueing network. The fluid model is a deterministic, continuous analog of the queueing network. It is defined through the following set of equations:

\begin{align}
Z_1(t) &= Z_1(0) + \alpha_1 t - \mu_1 T_1(t), \quad t \geq 0, \\
Z_k(t) &= Z_k(0) + \mu_{k-1} T_{k-1}(t) - \mu_k T_k(t), \quad t \geq 0, \quad k = 2, \ldots, 5, \\
Z_k(t) &\geq 0, \quad t \geq 0, \quad k = 1, \ldots, 5, \\
T_k(t) &\text{ is non-decreasing in } t, \quad k = 1, \ldots, 5, \\
t - T_k^+(t) &\text{ is non-decreasing in } t, \quad k = 1, \ldots, 5, \\
T_k^+(t) &= 1 \text{ for any time } t \text{ with } Z_k^+(t) > 0 \text{ for } k = 1, \ldots, 5,
\end{align}

where, $\mu_k = 1/m_k$, $Z_k^+(t)$ is the sum of $Z_\ell(t)$ over all classes $\ell$ that have priority at least $k$ and are served at the same station as class $k$. For example, for our network, operating under the priority policy defined in Section 2, we have

$$Z_4^+(t) = Z_1(t) + Z_3(t) + Z_4(t) \quad \text{and} \quad Z_1^+(t) = Z_1(t).$$

The quantity $T_k^+(t)$ is defined in a similar manner. For a function $f(\cdot) : [0, \infty) \to \mathbb{R}^d$ for some integer $d$, $f(t)$ denotes the derivative of $f$ at time $t$.

Each function $(T, Z)$ satisfying (2)-(7) with $T(t) = (T_1(t), \ldots, T_5(t))$ and $Z(t) = (Z_1(t), \ldots, Z_5(t))$ is called a fluid solution to the fluid model. The quantities $Z(t)$ and $T(t)$ have the following interpretation. For each class $k$, $Z_k(t)$ is the fluid level in buffer $k$ at time $t$, and $T_k(t)$ is the amount of time that the class $k$ server has spent serving class $k$ fluid in $[0, t]$. Thus, $\mu_k T_k(t)$ is the cumulative amount of fluid that has departed from buffer $k$ in $[0, t]$. Equations (2) and (3) simply balance the flows in the network. Equation (5) ensures that the amount of time spent on a class is non-decreasing and equation (6) says that the cumulative remaining time for a server, excluding the time spent on classes with priorities of at least $k$, is non-decreasing. Condition (7) follows from...
the SBP policies employed, i.e., when a high priority buffer has a positive amount of fluid, that server should not devote any effort to a lower priority buffer. In the cases $k = 4$ and $k = 2$, (7) simply insures that the fluid network operates under a non-idling policy.

It can be shown that each fluid solution $(T, Z)$ is Lipschitz continuous with respect to $t$; see, for example, Dai [9]. Therefore, each solution is also absolutely continuous and thus has derivatives for almost every $t$. Whenever a derivative like the one in (7) is employed, it is automatically assumed that $(T, Z)$ is differentiable at time $t$.

Now we construct a fluid solution that diverges to infinity. Most of the proof of Theorem 2.1 is devoted to showing that, for the original exponential network, the network dynamics approximately follow this divergent fluid solution. As will be seen shortly, such a fluid solution exists because of the particular choices of the SBP policy and the mean service times employed in our network. It turns out that the divergent fluid solution always exists when the SBP policy is employed and the mean service times satisfy

$$\rho_{\text{push}} := \alpha_1 m_5 + \alpha_1 \frac{m_3}{1 - \alpha_1 m_1} > 1. \quad (8)$$

When $\rho_{\text{push}} \leq 1$ and

$$\rho_A := \alpha_1 (m_1 + m_3 + m_4) \leq 1 \quad \text{and} \quad \rho_B := \alpha_1 (m_2 + m_5) \leq 1, \quad (9)$$

then no divergent fluid solution exists. The conditions in (9) are the so-called usual traffic conditions. Condition (8) violates the push start condition, first identified in Dai and Vande Vate [14]. The push start condition is a magnification of a virtual station phenomenon first observed by Harrison and Nguyen [17] and Dumas [15] and later systematically treated in Dai and Vande Vate [13] and [14]. See Section 7 for more discussion on virtual station and push start conditions.

Now, to construct the divergent fluid solution, we start the system with initial fluid level $Z(0) = (0, 0, 0, 0, 0, 0)$. We present a fluid solution in one period that ends when the system state reaches a state $(0, 0, 0, 0, +, 0)$ with the fluid level in buffer 4 exceeding one unit. (The plus sign indicates the buffer level is positive.) Clearly, such a construction can be extended from period to period to construct a solution which diverges to infinity with time. Within a period, the system evolves in two cycles: the bottom cycle and the top cycle. During the bottom cycle, the initial fluid in buffer 4 drains into buffer 5 and then exits the network. During this draining period, fluid accumulates in buffer 2. The bottom cycle ends when all fluid has drained from buffers 4 and 5, and buffer 2 is the only buffer with a positive amount of fluid. At this point, the top cycle begins. During this cycle, fluid in buffer 2 drains into buffer 3 and accumulates in buffer 4. The cycle ends when all fluid has been drained from buffers 2 and 3 and all fluid in the network resides in buffer 4.

The remainder of this section gives a detailed construction of these two cycles. We use $d_k(t)$ to denote the departure rate $\mu_k T_k(t)$ from buffer $k$ at time $t$. When the time $t$ is clear from the context, we drop the time dependence from the departure rate notation. Note that if we fully specify the departure rates $d_k(t)$, for $t \geq 0$ and $k = 1, \ldots, 5$, a resulting fluid solution $(T, Z)$ is uniquely defined. Of course, one needs to check that the solution satisfies the fluid model equations (2)-(7). This step is routine but tedious, and thus is not provided here.

**Bottom cycle.** As the cycle begins, buffer 1 is initially empty. Since buffer 1 has highest priority, and the arrival rate to buffer 1 is slower than the service rate at buffer 1, buffer 1 will remain empty at all times. However, note that Station A needs to spend $\alpha_1 m_1 = 0.4$ fraction of its time to keep buffer 1 empty. The remaining 60% of the server's capacity may be spent on buffers 3 and 4. If Station A spends all of this 60% remaining capacity on buffer 4, it can process class 4 fluid at a rate of $d_4 = \mu_4 (1 - \alpha_1 m_1) = 6$, which is faster than the maximum service rate $d_5 = \mu_5$ at buffer 5. Hence, at the beginning of the bottom cycle, class 4 fluid is being processed faster than class 5
fluid. So, fluid will accumulate at buffer 5 and furthermore, due to our priority policy, Station B is prevented from serving any class 2 fluid. Therefore, for an initial period of time, buffers 1 and 3 remain empty with buffer 3 having no service activities at all, buffers 2 and 5 accumulate fluid, and buffer 4 drains fluid. Such a state will persist until buffer 4 empties at time \( t_1 \). At this point, the fluid level in the network is \( Z(t_1) = (0,+,0,0,+) \), with a positive amount of fluid in buffers 2 and 5. Since there is no input to buffer 5 immediately after \( t_1 \), buffer 5 will begin draining fluid, and buffer 2 will continue to accumulate fluid. Meanwhile, all other buffers remain empty, with only buffers 1 and 5 processing fluid. This state will continue until buffer 5 empties at time \( t_2 \). Note that during \([0,t_2)\), Station B is spending 100\% of its effort processing class 5 fluid, and that it processes exactly one unit of fluid in this time. Hence, \( t_2 = m_5 \) and at this time, buffer 2 has \( \alpha_1 m_5 \) units of fluid. Thus, the fluid level is given by \( Z(t_2) = (0,\alpha_1 m_5,0,0,0) \). This is the end of the bottom cycle.

**Top cycle.** As soon as buffer 5 is empties, Station B begins processing class 2 fluid at rate \( d_2 = \mu_2 = 10 \). This departure rate from buffer 2 will overwhelm buffer 3, which has a maximum service rate of \( \mu_3 = 2.5 \). Station A must continue to devote 40\% of its time to class 1 fluid. Hence, Station A can only devote 60\% of its capacity to buffer 3, and the departure rate from buffer 3 will be \( d_3 = \mu_3(1 - \alpha_1 m_1) = 1.5 \). Furthermore, Station A cannot devote any processing capacity to class 4 fluid. Thus, in the period immediately after \( t_2 \), buffers 3 and 4 accumulate fluid, buffer 2 drains, and buffers 1 and 5 remain empty. This state will continue until buffer 2 empties at \( t_3 \). From this point on, external fluid flows through buffers 1 and 2 instantaneously to buffer 3. Since this external rate \( \alpha_1 = 1 < d_3 = 1.5 \), in the period immediately after \( t_3 \), class 3 fluid drains into buffer 4, buffer 4 accumulates fluid, and all other buffers remain empty. This state continues until buffer 3 empties at time \( t_4 \). At this time, all buffers are empty except buffer 4. Thus, the fluid level is \( Z(t_4) = (0,0,0,+,0) \). To calculate the amount of fluid in buffer 4, we note that the \( \alpha_1 m_5 \) units of fluid which were present in buffer 2 at time \( t_2 \) have simply moved to buffer 4 at time \( t_4 \). In addition, \( \alpha_1 (t_4 - t_2) \) units of fluid have arrived from the outside during \([t_2,t_4]\) and reside in buffer 4 at \( t_4 \). Thus, \( Z_4(t_4) = \alpha_1 m_5 + \alpha_1 (t_4 - t_2) \). To calculate \( t_4 - t_2 \), we note that during \([t_2,t_4]\), the departure rate from the “pipe” from buffer 1 to buffer 3 is a constant, \( d_3 \). The input rate to this pipe is \( \alpha_1 \). The initial amount in the pipe at \( t_2 \) is \( \alpha_1 m_5 \). Thus, the pipe must empty in time

\[
 t_4 - t_2 = \frac{\alpha_1 m_5}{(d_3 - \alpha_1)} = \frac{\alpha_1 m_5 m_3}{(1 - \alpha_1 m_1 - \alpha_1 m_3)} = \frac{m_5 m_3}{(1 - m_1 - m_3)}.
\]

Thus,

\[
 Z_4(t_4) = \alpha_1 m_5 + \alpha_1 (t_4 - t_2) = \frac{\alpha_1 m_5}{1 - \alpha_1 m_3/(1 - \alpha_1 m_1)} = \frac{(1 - m_1) m_5}{1 - m_1 - m_3} = \frac{6}{5} > 1. \tag{10}
\]

Therefore, our top cycle ends at \( t_4 \) with fluid level \((0,0,0,6/5,0)\). One can check that in (10), \( Z_4(t_4) > 1 \) is equivalent \( \rho_{\text{push}} > 1 \), for general mean service times. Whenever the usual traffic conditions hold and \( \rho_{\text{push}} > 1 \), our construction always leads to a divergent fluid solution.

### 4 The Exponential Network – Preliminary Proofs

The majority of this section is devoted to proving the following theorem. Henceforth, we let \( t^+ \) denote the time immediately after time \( t \).

**Theorem 4.1.** **Consider the exponential network operating under the non-preemptive SBP service policy. Suppose \( Z(0) = (0,z_2,0,n,z_5) \) with a class 4 job entering service at time 0 and a class 2**
job not in service at time $0^+$. Then for any $0 < \theta < 1$, there exists an $\epsilon > 0$ such that for all sufficiently large $n$, \begin{equation}
P\left\{ Z_4(T_4) \geq \frac{(1 - m_1)m_5}{1 - m_1 - m_3} \theta n \right\} \geq 1 - \exp(-\epsilon \sqrt{n}), \end{equation}
where \begin{align}
T_2 &= \inf\{ t > 0 : Z_3(t) = Z_4(t) = Z_5(t) = 0 \}, \\
T_4 &= \inf\{ t > T_2 : \text{a class 4 job enters service at time } t \\
&\quad \text{and a class 2 job is not in service at } t^+ \}.
\end{align}
Furthermore, for all sufficiently large $n$, \[ P \{ |Z(t)| \geq n/4, \forall t \in [0, T_4] \} \geq 1 - \exp(-\epsilon \sqrt{n}). \]

We envision $n$ in the initial state $Z(0)$ as large, with $z_2$ and $z_5$ being relatively small. However, the theorem holds for arbitrary $z_2$ and $z_5$. Such an initial state corresponds to the initial fluid model state $(0, 0, 0, 1, 0)$ used in Section 3. Note that at $T_4$, a class 4 job has just entered service. Thus it is necessarily true that buffers 1 and 3 are empty at $T_4$. Hence, at time $T_4$, the network has returned to a state similar to the initial state, with a magnification factor $\theta(1 - m_1)m_5/(1 - m_1 - m_3)$. We will refer to the time interval $[0, T_4]$ as a cycle, in alignment with the fluid network dynamics. Similarly, the interval $[0, T_2]$ is said to form a bottom cycle and the interval $[T_2, T_4]$ is said to form a top cycle.

The analogy between Theorem 4.1 and the unstable fluid solution constructed in Section 3 is evident. The magnification factor for the exponential network in (11) is smaller than the one for the fluid model in (10) due to randomness in our exponential network. However, since $(1 - m_1)m_5/(1 - m_1 - m_3) > 1$, one can always choose a $\theta < 1$ such that the factor for the stochastic network in (11) is still strictly bigger than one.

Although we have said that our attention will be restricted to the network with a mean service time vector of $m = (0.4, 0.1, 0.4, 0.1, 0.4)$, the proof of Theorem 4.1 is actually general and holds for any service time vector for which $\rho_A < 1, \rho_B < 1$ and $\rho_{push} > 1$.

In Section 5, we use Theorem 4.1 to complete the proof of Theorem 2.1. The remainder of this section is devoted to the proof of Theorem 4.1. This proof, along with numerous supporting lemmas, occupies the bulk of this paper. The actual proof of Theorem 4.1 will be presented in Section 4.5, with the various lemmas presented in Sections 4.1–4.4. In Section 4.1, we show that during the bottom cycle the exponential network closely follows the unstable fluid solution in $[0, t_2]$. In Section 4.3, we show that during the top cycle the exponential network closely follows the unstable fluid solution in $[t_2, t_4]$. Sections 4.2 and 4.4 detail how the exponential network moves from the bottom cycle to the top cycle and from the top cycle to the bottom cycle, respectively. Readers who intend to read the rest of this section seriously should first understand thoroughly the unstable fluid solution constructed in Section 3.

In the following sections, we will introduce a number of positive constants: $\epsilon_1, \epsilon_2, \ldots$. Since the exact values of the constants are not important for our final result, we will not keep track of the values or relationships between the constants.

### 4.1 The Bottom Cycle

At the beginning of what we call the bottom cycle, there are a large number of jobs in buffer 4. We wish to show that once these jobs begin processing, buffer 5 will eventually be overwhelmed.
with jobs, thus preventing buffer 2 jobs from being processed. Hence, once the large number of
original class 4 jobs have completed processing at buffers 4 and 5, there will be a large build-up of
jobs waiting at buffers 1 and 2. The goal of this subsection is show that with high probability, the
behavior described above occurs and that the number of jobs in buffers 1 and 2 at the end of the
bottom cycle is $\theta_1 m_5 n$, where $\theta_1$ is a constant arbitrarily close to 1. These statements are made
more precise in the following theorem, which is the main result for the bottom cycle.

**Theorem 4.2.** Suppose $Z(0) = (0, z_2, 0, n, z_5)$ with a class 4 job entering service at time 0 and a
class 2 job not in service at time 0+. Then for all $0 < \theta_1 < 1$, there exist an $\epsilon_1 > 0$ and a Markov
time $T_2$ (as defined in (12)), with $Z(T_2) = (Z_1(T_2), Z_2(T_2), 0, 0, 0)$ such that for all $n$ sufficiently
large,

$$
P \{Z_1(T_2) + Z_2(T_2) \geq \theta_1 m_5 n \} \geq 1 - \exp(-\epsilon_1 \sqrt{n}).$$

We now introduce a number of definitions needed for the proof of Theorem 4.2.

### 4.1.1 Buffer 5 Busy and Impure Periods

Recall that we have interpreted $T_2$ as being the time a bottom cycle is completed, i.e., the large
number of jobs originally in buffer 4 have been cleared from buffers 4 and 5 and all the jobs in
the network are in buffers 1 and 2. Unlike the unstable fluid model solution in Section 3, buffer 5
may not be always busy during the entire interval $[0, T_2]$ even if buffer 5 initially contains a job.
Although, on average, the processing time of a class 5 job is longer than that of a class 4 job, buffer
5 may be empty from time to time in $(0, T_2)$ due to the randomness in these processing times. Each
time buffer 5 is busy and a class 4 job enters service, buffer 4 has a chance to "overwhelm" buffer 5
entirely until buffer 4 is empty. However, there is also the possibility that buffer 4 will not succeed
in overwhelming buffer 5, if buffer 5 empties prematurely (i.e., before buffer 4 is cleared of all jobs).
Obviously, such emptying times are important for our analysis. We recursively define these times
here. Let $\sigma_1 = 0$ and define

$$
\tau_1 = \inf \{ t \geq \sigma_1 : \text{a class 2 job enters service at time } t \}. 
$$

Next, we define $\sigma_i$ and $\tau_i$ recursively as follows:

$$
\sigma_{i+1} = \inf \{ t \geq \tau_i : \begin{array}{c}
\text{a class 4 job enters service at time } t \\
\text{and a class 2 job is not in service at } t+ 
\end{array} 
$$

$$
\tau_{i+1} = \inf \{ t \geq \sigma_{i+1} : \text{a class 2 job enters service at time } t \}.
$$

Note that at time $\sigma_i < \infty$, it is necessarily true that buffers 1 and 3 are empty. During $[\sigma_i, \tau_i)$,
Station B either serves class 5 jobs or stays idle. Thus, there are no jobs moving from buffer 2 to
buffer 3, and hence buffer 3 remains empty during the period. If buffer 4 happens to be empty at
$\tau_i$, we know that the entire bottom cycle ends at that time. Let

$$
r = \inf \{ i \geq 0 : Z_4(\tau_i) = 0 \}. 
$$

(14)

It is clear that $T_2 \in (\sigma_r, \tau_r)$. Thus, $r$ is also the smallest $i$ such that $\tau_i \geq T_2$. For future purposes,
we summarize some basic properties in the following proposition.

**Proposition 4.3.**

(a) For each $i$, buffer 3 is empty throughout the interval $[\tau_i, \sigma_i)$.

(b) Throughout the interval $[\tau_i, \sigma_i)$, Station B is either working on class 5 jobs or stays idle. In
the latter case, buffer 2 is necessarily empty.
(c) For each \( i < r \), buffer 4 is nonempty throughout the interval \([\tau_i, \sigma_i]\).

We call the interval \([\sigma_i, \tau_i]\) the \( i \)th buffer 5 busy period or simply the \( i \)th busy period, and \([\tau_i, \sigma_{i+1}]\) the \( i \)th impure period, for \( i = 1, 2, \ldots \). When \( i < r \), the \( i \)th busy period is said to be incomplete. When \( i = r \), the busy period is said to be the last busy period. Note that it is possible for there to be only one (initial) busy period and no impure periods in \([0, T_2]\).

It is clear that all these random times depend on the parameter \( n \) or more generally on the initial state \( Z(0) \). To keep our notation simple, we do not explicitly denote such dependence.

Next, we note that while the network is in buffer 5 busy periods, class 4 jobs will be, on average, processed faster than class 5 jobs, even with interruptions to serve the higher priority class 1 jobs. The next lemma makes this statement more precise.

**Lemma 4.4.** Suppose the network is in a buffer 5 busy period and let \( u_i \) be the time between service completions of class 4 jobs, before buffer 4 empties for the first time. Then the \( u_i \) are i.i.d. and

\[
\mathbb{E}[u_1] = \frac{m_4}{1 - m_1} < m_5.
\]

**Proof.** Once a class 4 job has been completed, Station A must serve any class 1 jobs which arrived during the previous class 4 service, until buffer 1 is empty. Once all class 1 jobs are cleared, a class 4 job will enter and complete service uninterrupted, due to the non-preemption assumption. Hence the expected time between service completions is given by

\[
\mathbb{E}[u_i] = \mathbb{E}[v_i + s_{i+1}],
\]

where \( v_i \) is the time it takes to complete the class 1 jobs after the \( i \)th service at buffer 4 and \( s_{i+1} \) is the \( i + 1 \)st service time at buffer 4. Recall that the \( s_i \) were assumed to be i.i.d. Furthermore, each time a class 4 job enters service, there must be zero jobs in buffer 1. This fact, along with memoryless property of the interarrival times, implies that the \( v_i \) are also i.i.d. In particular, we have

\[
\mathbb{E}[u_i] = \mathbb{E}[v_i] + m_4. \tag{15}
\]

We now proceed to derive an expression for \( \mathbb{E}[v_i] \). Let \( N_i \) be the number of jobs in buffer 1 after the \( i \)th service completion at buffer 4. Conditioning on \( s_i \), we have:

\[
\mathbb{E}[N_i] = \mathbb{E}(\mathbb{E}[N_i \mid s_i]) = \mathbb{E}(\alpha_1 s_i) = \mathbb{E}(s_i) = m_4.
\]

Using the same procedure after conditioning on \( N_i \), we have:

\[
\mathbb{E}[v_i] = \mathbb{E}(\mathbb{E}[v_i \mid N_i]) = \mathbb{E}\left[\frac{m_1}{1 - m_1} N_i\right] = \frac{m_1}{1 - m_1} \mathbb{E}[N_i] = \frac{m_1}{1 - m_1} m_4.
\]

The second line is obtained by applying the formula for the mean absorption time to zero from state \( N_i \), for a birth-death with constant birth rate 1 and constant death rate \( 1/m_1 \) (see e.g. Karlin and Taylor [20], p. 149).
Plugging the above expression into (15) and doing some algebra yields:

$$E[u_i] = \frac{m_4}{1 - m_1}.$$  

One can check that when the usual traffic conditions are satisfied and $\rho_{push} > 1$, that $E[u_i] < m_5$. 

Lemma 4.4 states that in a buffer 5 busy period, jobs arrive at buffer 5 faster than they depart from buffer 5, on average. There is a positive probability that buffer 5 empties during a busy period, and before buffer 4 has emptied, which leads to the end of an incomplete busy period. However, such a sequence of events cannot not happen too often, which we demonstrate in the next lemma.

**Lemma 4.5.** There exists a constant $0 < c < 1$ such that for each $i \geq 1$,

$$P\{r \geq i\} \leq c^i.$$  

**Proof.** We note that

$$P\{r > i\} = P\{r > i - 1, \tau_i < \infty, Z_4(\tau_i) > 0\}$$

$$= P\{r > i - 1\} P\{\tau_i < \infty, Z_4(\tau_i) > 0| r > i - 1\},$$  

(16)

Now, on the event that $\{r > i - 1\}$, $\{\sigma_i < \infty\}$ and the network starts a new busy period at $\sigma_i$ with state $Z(\sigma_i)$.

Consider a birth-death process on $\{0, 1, 2, 3, \ldots\}$ with a birth rate of $(1 - m_1)/m_4$ which is greater than the death rate $1/m_5$. By Lemma 4.4, the number of jobs in buffer 5 during the busy period $[\sigma_i, \tau_i)$ is such a birth-death process, assuming that buffer 4 never runs out jobs within the period. At the beginning of the busy period, either a class 5 job is in service or Station B is empty. In the latter case, a job will arrive at buffer 5 when the first job in the period finishes its service at buffer 4. In either case, we assume without loss of generality that the birth-death process starts from a state that is bigger than $\rho$ or equal to 1.

Since

$$\{\tau_i < \infty, Z_4(\tau_i) > 0\} \subseteq \{\text{the birth-death process ever reaches state 0}\},$$

and the probability $c$ for the birth-death process to ever reach state 0 is strictly less than one, it follows from (16) that $P\{r > i\} \leq P\{r > i - 1\} c$ for each $i$. From this, and induction, the lemma follows.  

As a consequence of the lemma, we have the following corollary.

**Corollary 4.1.** (a) $P\{r < \infty\} = 1$, and (b) $P\{T_2 < \infty\} = 1$.

**Proof.** Part (a) follows from Lemma 4.5. From Part (a), we have $P\{\sigma_r < \infty\} = 1$. It follows that $P\{\tau_r < \infty\}$, which implies (b).  

4.1.2 Proof for Bottom Cycle

The goal in this subsection is to provide a probabilistic bound on the number of jobs remaining in buffers 4 and 5 when the last busy period begins. This is the content of Theorem 4.6.

**Theorem 4.6.** For any $0 < \theta_2 < 1$, there exists an $\varepsilon_2 > 0$, such that for all sufficiently large $n$,

$$P\{Z_4(\sigma_r) + Z_5(\sigma_r) \geq \theta_2 n\} \geq 1 - \exp(-\varepsilon_2 \sqrt{n}).$$

11
Theorem 4.6 says that by time $\sigma_r$, the number of jobs that have departed from buffer 5 is a small fraction of $n$ with large probability. The proof of the theorem will be given at the end of the subsection. To aid the proof, we need to examine in detail how jobs depart buffer 5. We call a job a **leak** if it completes processing at buffer 5 during $[0,\sigma_r]$. We are going to show that within a period $(\sigma_i, \sigma_{i+1})$, there cannot be too many leaks, when $i < r$.

So, let us fix a period $(\sigma_i, \sigma_{i+1})$. Recall that the interval $[\sigma_i, \tau_i)$ is called a buffer 5 busy period and the interval $[\tau_i, \sigma_{i+1})$ an impure period. The number of leaks that can happen during the busy period will be shown to be small using Lemma A.2, when $i < r$. We now first control the number leaks during the impure period $[\tau_i, \sigma_{i+1})$.

By definition, buffer 5 must be empty at the beginning of an impure period $[\tau_i, \sigma_{i+1})$. Hence, during any impure period, the number of leaks is bounded above by the number of class 4 service completions during this period. It is possible that the first class 4 job completed during the impure period entered service before the impure period started. However, all subsequent class 4 service completions must be due to jobs which entered service during the impure period. In the next lemma, we derive a bound for such service completions.

**Lemma 4.7.** Let $q_i$ be the number of class 4 jobs that enter service within the $i$th impure period. There exists a constant $c$ with $0 < c < 1$ such that for $i = 1, \ldots$,

$$
\mathbb{P}\{q_i > 2j\} \leq c^j \quad \text{for } j = 0, 1, \ldots.
$$

**Proof.** Fix an impure period $[\tau_i, \sigma_{i+1})$. Each time a class 4 job enters service at time $t \in [\tau_i, \sigma_{i+1})$, a class 2 job must be in service at $t$. Otherwise, the impure period ends at a time $t$ that is strictly less than $\sigma_{i+1}$, contradicting the definition of $\sigma_{i+1}$.

Now consider the following sequence of events starting at time $t$: (Assume, for now, that the next interarrival time to buffer 1 is very long.)

1) The class 4 job completes service before the class 2 job.

2) A second class 4 job enters the service.

3) The class 2 job completes service and becomes a class 3 job.

4) A class 5 job enters service.

5) The second class 4 job completes service.

6) The class 3 job enters service.

7) The class 3 job completes service and becomes a class 4 job. At this moment, the third class 4 job enters service while the class 5 job is still in service, thus ending the impure period.

For the above sequence of events to be possible, it is enough to assume that there are no job arrivals to buffer 1 during the entire impure period.

Let $\xi_k$ be the time that the $k$th job entering class 4 service within the impure period, and let $A_k$ denote the intersection of the corresponding sequence of events (1–7 above) initiated by the $k$th job. If $A_k$ occurs, the impure period ends with $k + 1$ class 4 jobs having initiated services. Thus,

$$
\{q_i > 2j\} = \{\xi_{2j+1} < \sigma_{i+1}\} \subset \{\xi_{2j-1} < \sigma_{i+1}\} \cap A_{2j-1}^c,
$$

where $A_k^c$ is the complement of $A_k$. 

12
By the memoryless property of exponential distributions, the probability $\mathbb{P}\{A_k | \xi_k < \sigma_{i+1}\}$ is strictly bigger than 0. Denoting this non-zero probability by $1-c$, we have $c = \mathbb{P}\{A_k | \xi_k < \sigma_{i+1}\} < 1$. Note that this probability $c$ depends only on the network parameters, i.e., the mean interarrival and service times. Thus, we have

$$
\mathbb{P}\{q_i > 2j\} \leq \mathbb{P}\{\xi_{2j+1} < \sigma_{i+1}\} \leq \mathbb{P}\{\xi_{2j-1} < \sigma_{i+1}\} \cdot \mathbb{P}\{A^c_{2j-1} | \xi_{2j-1} < \sigma_{i+1}\} \\
= \mathbb{P}\{\xi_{2j-1} < \sigma_{i+1}\} c \\
\cdots \\
\leq \mathbb{P}\{\xi_1 < \sigma_{i+1}\} c^j \\
\leq c^j,
$$

proving (17).

Next, we want to control the number of leaks which occur during the $i$th busy period, for $i < r$.

Lemma 4.8. There exists an $\epsilon_3 > 0$, such that for all $n$ large enough, for each $i = 1, 2, \ldots$,

$$
\mathbb{P}\{\text{number of leaks during } [\sigma_i, \tau_i) \text{ exceeds } \sqrt{n}, \ i < r\} \leq \exp(-\epsilon_3 \sqrt{n}).
$$

Proof. Consider the number of jobs in buffer 5 during the busy period $[\sigma_i, \tau_i)$. The buffer 5 queue length process is identical to the queue length process in a $G/G/1$ queue with interarrival times given by the inter-departure times from buffer 4. By Lemma 4.4, the interarrival times are i.i.d. with mean $m_4/(1-m_1)$, which is smaller than the mean service time at buffer 5, $m_5$. At time $\sigma_i$, Station B is either working on a class 5 job or is idle. In the latter case, buffer 2 is necessarily empty, and the first class 4 job to complete service during the busy period will pass to buffer 5 and begin service during the busy period. In either case, applying Lemma A.2 at the time when a class 5 job is first in service during the busy period, the result follows.

Proof of Theorem 4.6. Let $\delta = 1 - \theta_2$. Then $\delta > 0$ and

$$
\mathbb{P}\{Z_4(\sigma_r) + Z_5(\sigma_r) \geq \theta_2 n\} \geq 1 - \mathbb{P}\{\text{more than } \delta n \text{ leaks from buffer 5 in } [0, \sigma_r]\}.
$$

Let $A = \{\text{more than } \delta n \text{ leaks during } [0, \sigma_r]\}$. To estimate the probability of $A$, we have the following

$$
\mathbb{P}(A) = \mathbb{P}(A \cap \{r \geq \sqrt{n}\}) + \mathbb{P}(A \cap \{r < \sqrt{n}\}) \\
\leq \mathbb{P}(\{r \geq \sqrt{n}\}) + \mathbb{P}(\bigcup_{i=1}^{\lfloor \sqrt{n} \rfloor} \{\text{at least } \delta \sqrt{n} \text{ leaks during } [\sigma_i, \sigma_{i+1}), \ i < r\}) \\
\leq \mathbb{P}(\{r \geq \sqrt{n}\}) + \sum_{i=1}^{\lfloor \sqrt{n} \rfloor} \mathbb{P}\{\text{at least } \delta \sqrt{n} \text{ leaks during } [\sigma_i, \sigma_{i+1}), \ i < r\} \\
\leq \exp(-\epsilon_4 \sqrt{n}) + \lfloor \sqrt{n} \rfloor \exp(-\epsilon_5 \sqrt{n}) \\
\leq \exp(-\epsilon_2 \sqrt{n}).
$$

In the second to last line of the proof, the first term follows directly from Lemma 4.5, with an appropriate $\epsilon_4$. The second term in the same line follows from Lemmas 4.7 and 4.8, again with an appropriate $\epsilon_5$. The final inequality is valid for some $\epsilon_2 > 0$ if $n$ is sufficiently large.
4.2 From Bottom to Top

Our primary goal in this subsection is to show that, with high probability, there are roughly $\theta_1 m_5 n$ jobs in buffers 1 and 2 at time $T_2$. In other words, the $n$ original jobs at buffer 4 have now "become" $\theta_1 m_5 n$ jobs in buffers 1 and 2. We begin with some lemmas.

Lemma 4.9. Let the Markov times $\tau_r$ and $T_2$ be defined as in (14) and (12), respectively. Then for any $0 < \theta_3 < 1$, there exists an $\epsilon_6 > 0$ such that for $n$ large enough

$$
P\{T_2 - \tau_r < \theta_3 m_5 n\} \leq \exp(-\epsilon_6 \sqrt{n})
$$

Proof. By the definition of $T_2$, all jobs present in buffers 4 and 5 at time $\tau_r$ will have departed the network by time $T_2$. By Theorem 4.6, we have that for any $0 < \theta_2 < 1$, buffer 5 must process $\theta_2 n$ jobs, except on an exponentially small set. So, $T_2 - \tau_r$ is the sum of at least $\theta_2 n$ i.i.d. exponential random variables with mean $m_5$. Applying Lemma A.1 and using Theorem 4.6 we have, for all $\alpha > 0$ there exists an $\epsilon_7 > 0$ such that for all $n$ large enough,

$$
P\{T_2 - \tau_r < \theta_2 m_5 n - \alpha n\} \leq \exp(-\epsilon_7 n) + \exp(-\epsilon_2 \sqrt{n})
$$

$$
P\{T_2 - \tau_r < \theta_3 m_5 n\} \leq \exp(-\epsilon_7 n) + \exp(-\epsilon_2 \sqrt{n})
$$

$$
P\{T_2 - \tau_r < \theta_3 m_5 n\} \leq \exp(-\epsilon_6 \sqrt{n})
$$

where we have set $\theta_3 = \theta_2 - \alpha / m_5$ to obtain the second expression above. Note that since $\alpha$ can be arbitrarily small, we can obtain the inequality for any $0 < \theta_3 < 1$. \hfill \Box

Now, since we have a lower bound on the time that buffer 5 is busy, we can obtain a lower bound on the number of jobs which must be in buffers 1 and 2 at time $T_2$. This is the main theorem for the bottom cycle, Theorem 4.2.

Proof of Theorem 4.2. Let $E_1(\cdot)$ be the counting process for exogenous arrivals to buffer 1 and let $Y_n$ be the time of the $n$th arrival during $[\tau_r, T_2]$. We choose any $\alpha > 0$. By applying Lemma 4.9 we have:

$$
P\{E_1[\tau_r, T_2] < \theta_3 m_5 n - \alpha n\} \leq P\{E_1[\tau_r, T_2] < \theta_3 m_5 n - \alpha n \mid T_2 - \tau_r \geq \theta_3 m_5 n\} + \exp(-\epsilon_6 \sqrt{n}).
$$

(18)

Next, we do some rearranging and apply Lemma A.1 in the last inequality:

$$
P\{E_1[\tau_r, T_2] < \theta_3 m_5 n - \alpha n \mid T_2 - \tau_r > \theta_3 m_5 n\} = P\{Y_{\theta_3 m_5 n - \alpha n} > T_2 - \tau_r \mid T_2 - \tau_r \geq \theta_3 m_5 n\}
$$

$$
\leq P\{Y_{\theta_3 m_5 n - \alpha n} > \theta_3 m_5 n\}
$$

$$
\leq P\{Y_{\theta_3 m_5 n - \alpha n} > \alpha n + \theta_3 m_5 n - \alpha n\}
$$

$$
\leq P\{Y_{\theta_3 m_5 n - \alpha n} > \alpha n + \theta_3 m_5 n - \alpha n\}
$$

$$
\leq \exp(-\epsilon_8 n).
$$

Now, plugging the above into (18) and setting $\theta_1 = \theta_3 - \alpha / m_5$, we have

$$
P\{E_1[\tau_r, T_2] < \theta_1 m_5 n\} \leq \exp(-\epsilon_8 n) + \exp(-\epsilon_6 \sqrt{n}) < \exp(-\epsilon_1 \sqrt{n}),
$$

for $n$ sufficiently large. Next, since all the exogenous arrivals in the interval $[\tau_r, T_2]$ must still be at buffers 1 and 2 at $T_2$, we have that

$$
Z_1(T_2) + Z_2(T_2) \geq E_1[\tau_r, T_2].
$$

Combining this with our previous inequality yields the theorem. \hfill \Box

14
4.3 The Top Cycle

$T_2$ is the beginning of what we shall call the top cycle. By virtue of Theorem 4.2, at time $T_2$ there are at least $\theta_1 m_5 n$ jobs in buffers 1 and 2, off an exponentially small set. Once buffer 2 begins processing this large number of jobs, we expect buffer 3 to be overwhelmed with jobs, with high probability. However, it is possible for buffer 3 to catch up, which may allow a class 4 job into service, and as in the bottom cycle, we may have buffers 3 and 5 processing jobs simultaneously. In this section, we wish to show that such a state will not persist for long and with high probability that buffer 2 will overwhelm buffer 3.

In order to state our main results, we need several definitions.

**Definition 4.10.** Let $$T_3 = \inf\{t > T_2 : Z_2(t) = Z_3(t) = 0\},$$ i.e., $T_3$ is the time at which we clear the large number of jobs from buffers 2 and 3. Note that $T_3$ is not analogous with the $t_3$ of the fluid iteration.

As in the previous subsection we wish to recursively define buffer 3 busy periods and other times, which we call impure periods. Let $\hat{\sigma}_1 = T_2$ and define $$\hat{\tau}_1 = \inf\{t \geq \hat{\sigma}_1 : \text{a class 4 job enters service at time } t\}.$$ The interval $[\hat{\sigma}_1, \hat{\tau}_1)$ is called the initial buffer 3 busy period. Next, we define $\hat{\sigma}_i$ and $\hat{\tau}_i$:

$$\hat{\sigma}_{i+1} = \inf\{t \geq \hat{\tau}_i : \text{a class 2 job enters service at time } t$$
$$\text{and there is no class 4 job in service at } t\},$$

$$\hat{\tau}_{i+1} = \inf\{t \geq \hat{\sigma}_{i+1} : \text{a class 4 job enters service at time } t\}.$$  

We call the interval $[\hat{\sigma}_i, \hat{\tau}_i)$ the $i$th buffer 3 busy period or simply the $i$th busy period, and $[\hat{\tau}_i, \hat{\sigma}_{i+1})$ the $i$th impure period, for $i = 1, 2, \ldots$. Note that it is possible for there to be only one busy period and no impure periods in $[T_2, T_3)$.

Let $$r = \inf\{i : Z_2(\hat{\tau}_i) > 0\}. \quad (19)$$ Note that $r$ is the smallest $i$ such that $\hat{\tau}_i \geq T_2$. Analogous to Lemma 4.5, we have

**Lemma 4.11.** There exists a constant $c$ with $0 < c < 1$ such that $$\mathbb{P}\{r \geq j\} \leq c^j \quad \text{for } j = 0, 1, \ldots.$$  

The proof of this lemma is actually simpler than that of Lemma 4.4 because there are no analogous external job arrivals to interfere with class 2 services. Thus, we do not need an additional lemma that is analogous to Lemma 4.4 for this proof. As before, we have as a corollary,

**Corollary 4.2.** (a) $\mathbb{P}\{r < \infty\} = 1$, and (b) $\mathbb{P}\{T_3 < \infty\} = 1$.

For $i < r$, the $i$th busy period is said to be incomplete, and for $i = r$, the $i$th busy period is said to be the last busy period. Thus, we call $[\hat{\sigma}_r, T_3)$ the last buffer 3 busy period. In this interval, buffer 3 never "catches-up" with buffer 2. Again, it is possible for $\hat{\sigma}_1 = \hat{\sigma}_r = T_2$, in which case the initial buffer 3 busy period and last buffer 3 busy period coincide.

As in the case for the bottom cycle, we want to control the number of jobs which "leak" during $[T_2, \hat{\sigma}_r]$. Within the period $[T_2, T_3)$, a job is called a leak if it is processed by buffer 2 during $[T_2, \hat{\sigma}_r]$. 

\[15\]
Theorem 4.12. For any $0 < \theta_4 < 1$, there exists an $\epsilon_9 > 0$, such that for all sufficiently large $n$,

$$\mathbb{P}\{Z_1(\hat{\sigma}_r) + Z_2(\hat{\sigma}_r) \geq \theta_4 m_5 n\} \geq 1 - \exp(-\epsilon_9 \sqrt{n}).$$

The proof of Theorem 4.12 depends crucially on the following lemmas, which give bounds on the number of leaks before the last busy period.

Lemma 4.13. Let $q_i$ be the number of class 2 jobs that have started their services during a single impure period $[\hat{r}_i, \hat{\sigma}_{i+1})$. Then there exists a constant $c$ with $0 < c < 1$ such that for $i = 1, \ldots,$

$$\mathbb{P}\{q_i > j\} \leq c^j \quad \text{for } j \geq 0.$$ 

Proof. The proof of the lemma is analogous to the proof of Lemma 4.7. Since the current lemma has a stronger result, we repeat some of the details here.

Fix an impure period $[\hat{r}_i, \hat{\sigma}_{i+1})$. Each time a class 2 job enters service at time $t$ within the period, a class 4 job must be in service at $t$. Otherwise, the impure period ends at a time $t$ that is strictly less than $\hat{\sigma}_{i+1}$, contradicting the definition of $\hat{\sigma}_{i+1}$.

Now consider the following sequence of events starting at time $t$:

1) An external arrival occurs before the class 4 job completes service.

2) The class 4 job completes service and becomes a class 5 job.

3) The class 1 job enters the service.

4) The class 2 job completes its service and becomes a class 3 job.

5) The class 5 job enters service.

6) The class 5 job completes its service before the class 1 job. At this moment, the second class 2 job enters service and ends the impure period.

Let $\xi_k$ be the time that the $k$th job entering class 2 service within the impure period, and let $A_k$ be the intersection of the corresponding sequence of events (1–7 above) initiated by the $k$th job. If the event $A_k$ occurs, the impure period ends with $k$ jobs having initiated services at buffer 2. Thus,

$$\{q_i > j\} = \{\xi_{j+1} < \hat{\sigma}_{i+1}\} \subset \{\xi_j < \hat{\sigma}_{i+1}\} \cap A_j^c.$$ 

Since the probability $\mathbb{P}\{A_k|\xi_k < \hat{\sigma}_{i+1}\}$ is strictly positive, depending only on the network parameters, the mean interarrival and service times, we have $\mathbb{P}\{A_k|\xi_k < \hat{\sigma}_{i+1}\} = c < 1$. Thus,

$$\mathbb{P}\{q_i > j\} \leq \mathbb{P}\{\xi_j < \hat{\sigma}_{i+1}\} c^j \leq c^j,$$

where the chain of inequalities is similar to that of the proof of Lemma 4.7. \qed
Lemma 4.14. There exists an $\varepsilon_{10} > 0$ such that for all $n$ sufficiently large,

$$\mathbb{P}\{\text{number of leaks from buffer 2 during } [\hat{\sigma}_i, \hat{\tau}_i) \text{ exceeds } \sqrt{n}, i < r \} \leq \exp(-\varepsilon_{10}\sqrt{n}).$$

Proof. The proof of the lemma is analogous to the proof of Lemma 4.8. However, in the top cycle case, in applying Lemma A.2, the interarrival times are determined by the service times of class 2 jobs. Thus, there is no need to use a lemma that is analogous to Lemma 4.4. \hfill \Box

With this lemma in hand, we have a result similar to the bottom cycle case.

Lemma 4.15. Let $0 < \delta < 1$, then there exists an $\varepsilon_{11} > 0$ such that for sufficiently large $n$,

$$\mathbb{P}\{\text{more than } \delta n \text{ leaks from buffer 2 in } [T_2, \hat{\sigma}_r] \} \leq \exp(-\varepsilon_{10}\sqrt{n}).$$

Proof. The proof follows from Lemmas 4.11, 4.13 and 4.14. \hfill \Box

Proof of Theorem 4.12.

$$\mathbb{P}\{Z_1(\hat{\sigma}_r) + Z_2(\hat{\sigma}_r) < \theta_4 m_5 n\} \leq \mathbb{P}\{Z_1(\hat{\sigma}_r) + Z_2(\hat{\sigma}_r) < \theta_4 m_5 n \mid Z_1(T_2) + Z_2(T_2) \geq \theta_1 m_5 n\}$$

$$\quad \leq \exp(-\varepsilon_{10}\sqrt{n}) + \exp(-\varepsilon_1\sqrt{n})$$

$$\leq \exp(-\varepsilon_{10}\sqrt{n}).$$

The first inequality follows by conditioning and applying Theorem 4.2. The second follows from Lemma 4.15. The third holds for appropriate $\varepsilon_9$ and sufficiently large $n$. \hfill \Box

4.4 From Top to Bottom

Next, we need to consider in detail what occurs during the interval $[\hat{\sigma}_r, T_3]$. Recall that once buffer 3 begins processing its first job after time $\hat{\sigma}_r$, it will remain positive until $T_3$. Thus, no class 4 job will be processed during $[\hat{\sigma}_r, T_3]$, and hence buffer 5 remains empty during $[\hat{\sigma}_r, T_3]$.

We divide $[\hat{\sigma}_r, T_3]$ into subintervals. We need the following definitions. First, set $R_0 = \hat{\sigma}_r$.

Definition 4.16. Consider the jobs which are in buffers 1 through 3 at time $R_0 = \hat{\sigma}_r$. Let $R_1$ be the time at which all these jobs have completed services at buffer 3.

Definition 4.17. Assume that $R_i$ has been defined. Consider all jobs in buffers 1 through 3 at time $R_i$. Let $R_{i+1}$ be the time at which all these jobs that have completed service at buffer 3.

For convenience, we define

$$S_{i+1} = R_{i+1} - R_i \text{ for } i = 0, 1, 2, \ldots,$$

which is the amount of time needed for all jobs in buffers 1 through 3 at time $R_i$ to complete service at buffer 3. Also, let $v = \inf\{i : R_i \geq T_3\}$.

Proposition 4.18. $\mathbb{P}\{v < \infty\} = 1$.

Proof. In every interval $[R_i, R_{i+1})$ the network must process at least one class 1 job. By the strong law of large numbers, $R_i \rightarrow \infty$ almost surely as $i \rightarrow \infty$. Since $T_3$ is almost surely finite by Corollary 4.2, there are a finite number of $R_i$ before $T_3$. \hfill \Box
We now wish to obtain a lower bound on $T_3 - \hat{\sigma}_r$ and the number of jobs which arrive in that time. This will then give a lower bound on the number of jobs in buffer 4 at time $T_3$.

**Lemma 4.19.** Suppose the network is in a buffer 3 busy period and let $u_i$ be the time between service completions of class 3 jobs. Then the $u_i$ are i.i.d. and

$$E[u_1] = \frac{m_3}{1 - m_1} > m_2.$$

**Proof.** The proof of the lemma is exactly analogous to that of Lemma 4.4. If $\rho_{push} > 1$ and the usual traffic conditions hold, then one can check that $E[u_1] > m_2$. □

**Lemma 4.20.** Let $\theta_5$ be any fixed constant with $0 < \theta_5 < 1$. There exists an $\epsilon_{11} > 0$ such that for all $s_i$ sufficiently large

$$\mathbb{P}\left\{S_{i+1} < \theta_5 \frac{m_3}{1 - m_1} \cdot s_i \mid Z_1(R_i) + Z_2(R_i) + Z_3(R_i) = s_i\right\} \leq \exp(-\epsilon_{11}s_i) \quad \text{for } i = 0, 1, \ldots.$$

**Proof.** During $[R_i, R_{i+1}]$, buffer 3 must process all the jobs present in buffers 1, 2 and 3 at time $R_i$. The result then follows from Lemmas A.1 and 4.19, as in the proof of Lemma 4.9. □

In the next lemma, recall that $E_1[s, t]$ denotes the number of external arrivals in the interval $[s, t]$.

**Lemma 4.21.** Let $\theta_6$ be any fixed constant with $0 < \theta_6 < 1$. There exists an $\epsilon_{12} > 0$ such that for all $s_i$ sufficiently large

$$\mathbb{P}\left\{E_1[R_i, R_{i+1}] < \theta_6 \frac{m_3}{1 - m_1} \cdot s_i \mid Z_1(R_i) + Z_2(R_i) + Z_3(R_i) = s_i\right\} \leq \exp(-\epsilon_{12}s_i).$$

**Proof.** The proof here is analogous to the proof of Theorem 4.2. □

We can now put all of the preceding estimates together to get an estimate of the number of jobs that have arrived during $[\hat{\sigma}_r, T_3]$. By definition, all of these jobs must then be in buffer 4 at time $T_3$. This leads to the following result.

**Theorem 4.22.** Suppose $Z(0) = (0, z_2, 0, n, z_5)$ with a class 4 job entering service at time 0 and a class 2 job not in service at time 0+. Then for all $0 < \theta_7 < 1$, there exists an $\epsilon_{13} > 0$ and a Markov time $T_3$, with $Z(T_3) = (Z_1(T_3), 0, 0, Z_4(T_3), 0)$ such that for all $n$ sufficiently large,

$$\mathbb{P}\left\{Z_4(T_3) \geq \frac{(1 - m_1)m_5}{1 - m_1 - m_3} \theta_7 n\right\} \geq 1 - \exp(-\epsilon_{13}\sqrt{n}).$$

**Proof.** By Theorem 4.12, we have that, off an exponentially small set, there are at least $s_0 = \theta_4m_5n$ jobs in buffers 1 and 2 at time $R_0 = \hat{\sigma}_r$. An application of Lemmas 4.20 and 4.21 yields that for $n$ large enough, off an exponentially small set there will be at least

$$\left(\theta_6 \frac{m_3}{1 - m_1}\right) \theta_4m_5n$$

arrivals while we are processing the jobs present at $\hat{\sigma}_r$. So, at time $R_1$, there are at least

$$s_1 = \left(\theta_6 \frac{m_3}{1 - m_1}\right) \theta_4m_5n$$

18
jobs in buffers 1, 2 and 3, off an exponentially small set. Reasoning similarly, again off exponential sets, at time $R_i$ there are at least

$$s_i = \left( \theta_6 \frac{m_3}{1 - m_1} \right)^i \theta_4 m_5 n$$

jobs in buffers 1 through 3.

Next, fix a positive integer $N$. Define

$$K = \sum_{i=0}^{N} \left( \theta_6 \frac{m_3}{1 - m_1} \right)^i \theta_4 m_5.$$

Since

$$\sum_{i=0}^{\infty} \left( \theta_6 \frac{m_3}{1 - m_1} \right)^i \theta_4 m_5 = \frac{\theta_6 (1 - m_1)}{1 - m_1 - \theta_6 m_3} \theta_4 m_5,$$

for any $\theta_7$ with $0 < \theta_7 < 1$, one can choose $\theta_4$, $\theta_6$ and $N$ with $0 < \theta_i < 1$ such that

$$K \geq \theta_7 \frac{(1 - m_1)m_5}{1 - m_1 - m_3}. \quad (20)$$

Next, for $n$ large enough so that $s_i$, $i = 1, \ldots, N$, is large enough to apply Lemmas 4.20 and 4.21, we show that at $R_N$ we will have processed $Kn$ jobs at buffer 3, off an exponentially small probability set. Note that at $T_3$, buffer 4 must contain all of the jobs processed at buffer 3 during $[\sigma_T, T_3)$. In particular, buffer 4 must contain at least $Kn \geq \theta_7(1 - m_1)m_5/(1 - m_1 - m_3)$ jobs at $T_3$, off a small set.

To complete the proof, we must indeed verify that the above claim is true, except off an exponentially small set. To see this, first note that the size of $N$ required to make (20) valid depends only on the problem data $m_i$, which is fixed, and the $\theta_i$, whose necessary closeness to one is also fixed, depending on the problem data $m_i$. Hence the necessary size of $N$ can be fixed once and for all given the problem data. Next, we need to have the $s_1, s_2, \ldots, s_N$ sufficiently large at times $R_1, \ldots, R_N$ to apply Lemmas 4.20 and 4.21. Since $N$ does not depend on the initial number of jobs $n$ in the system, we can make $n$ large enough so that $s_N$, and thus $s_i$ for $i = 0, \ldots, N - 1$, is large enough to apply Lemmas 4.20 and 4.21. Thus, we have

$$\mathbb{P}\{Z_1(R_{i+1}) + Z_2(R_{i+1}) + Z_3(R_{i+1}) \leq s_{i+1}\}$$

$$\leq \mathbb{P}\{Z_1(R_i) + Z_2(R_i) + Z_3(R_i) \leq s_i\}$$

$$+ \sum_{s=s_i}^{\infty} \mathbb{P}\{Z_1(R_{i+1}) + Z_2(R_{i+1}) + Z_3(R_{i+1}) \leq s_{i+1} | Z_1(R_i) + Z_2(R_i) + Z_3(R_i) = s\}$$

$$\times \mathbb{P}\{Z_1(R_i) + Z_2(R_i) + Z_3(R_i) = s\}$$

$$\leq \mathbb{P}\{Z_1(R_i) + Z_2(R_i) + Z_3(R_i) \leq s_i\} + \exp(-\epsilon_{12}s_i)$$

$$\ldots$$

$$\leq \mathbb{P}\{Z_1(R_0) + Z_2(R_0) + Z_3(R_0) \leq s_0\} + \sum_{k=0}^{i} \exp(-\epsilon_{12}s_k)$$

$$\leq \mathbb{P}\{Z_1(R_0) + Z_2(R_0) + Z_3(R_0) \leq s_0\} + (i + 1) \exp(-\epsilon_{12}s_i)$$

$$\leq \exp(-\epsilon_{9}\sqrt{n}) + N \exp(-\epsilon_{12}s_N) \quad \text{for } i = 0, \ldots, N - 1,$$
where, in obtaining the last inequality, we have used Theorem 4.12. Now,

\[
P\left\{ Z_4(T_3) \leq \theta_7 n \frac{m_5(1 - m_1)}{1 - m_1 - m_3} \right\}
\leq \ P\left\{ \sum_{k=0}^{N} (Z_1(R_k) + Z_2(R_k) + Z_3(R_k)) \leq \theta_7 n \frac{m_5(1 - m_1)}{1 - m_1 - m_3} \right\}
\leq \ P\left\{ \sum_{k=0}^{N} (Z_1(R_k) + Z_2(R_k) + Z_3(R_k)) \leq \sum_{k=0}^{N} s_k \right\}
\leq \sum_{k=0}^{N} P\{Z_1(R_k) + Z_2(R_k) + Z_3(R_k) \leq s_k\}
\leq (N + 1) \left( \exp(-\epsilon_9 \sqrt{n}) + N \exp(-\epsilon_{12} s_N) \right)
\leq \exp(-\epsilon_{13} \sqrt{n}).
\]

So we conclude that for any $0 < \theta_7 < 1$, and there exists $\epsilon_{13} > 0$ such that for $n$ sufficiently large

\[
P\left\{ Z_4(T_3) \geq \theta_7 n \frac{m_5(1 - m_1)}{1 - m_1 - m_3} \right\} \geq 1 - \exp(-\epsilon_{13} \sqrt{n}).
\]

By definition we have $Z(T_3) = (Z_1(T_3), 0, 0, Z_4(T_3), 0)$ and a class 3 job was completed at $T_3$. This concludes the proof of the theorem. \(\Box\)

4.5 Proof of Theorem 4.1

Theorem 4.22 essentially shows that if we start with a large number of jobs in buffer 4, then with very high probability there will be a large number of jobs in buffer 4 some time later. To complete the proof of our main result, Theorem 2.1, we must obtain three additional results.

First, we note that the beginning and ending states in Theorem 4.22 are not qualitatively identical. In the theorem, at time 0 a class 4 job enters service. In the conclusion of the theorem, we have that at time $T_3$ a class 1 job may be in service (if $Z_1(T_3) \neq 0$). At time 0, the network enters what we have called a buffer 5 busy period. Our first task is to “complete the loop” in Theorem 4.22. Specifically, we wish to show that some time after $T_3$, the network will once again enter a buffer 5 busy period, without losing too many jobs from buffer 4 (jobs which were present at $T_3$). This is the content of Theorem 4.1.

Next, we use the results of Section 4 and Theorem 4.1 to show that we can put a lower bound on the total number of jobs in the network at any time during a cycle. We demonstrate this in Theorem 4.23.

Proof of Theorem 4.1. If $Z_1(T_3) = 0$, then we set $T_4 := T_3$ and we are done. If not, then after the class 3 job has completed service at time $T_3$ the network is entering a typical impure period for the bottom cycle. When it exits this impure period, it will enter a buffer 5 busy period and the network will be in a state as described by the conclusion of the theorem. We call the time that it exits the impure period $T_4$. Note that this definition is consistent with the definition of $T_4$ given in the statement of Theorem 4.1.

Next, let $N$ be the number of jobs which are leaked from buffer 4 during this impure cycle interval, $[T_3, T_4)$. It is bounded $q + 1$, where $q$ is the number of class 4 jobs that have started service
in the interval. By Lemma 4.7, we have that there exists an $\epsilon_{14} > 0$ such that for all $n$ sufficiently large

$$\mathbb{P}\{N \geq \sqrt{n}\} \leq \exp(-\epsilon_{14}\sqrt{n}).$$

We can now combine this estimate with the result of Theorem 4.22 as follows:

$$\mathbb{P}\{Z_4(T_4) \leq c \theta_7 n - c \theta_7 \sqrt{n}\} \leq \mathbb{P}\{Z_4(T_4) \leq c \theta_7 n - c \theta_7 \sqrt{n} \mid Z_4(T_3) \geq c \theta_7 n\}$$

$$+ \mathbb{P}\{Z_4(T_3) < c \theta_7 n\}$$

$$\leq \exp(-\epsilon_{14}\sqrt{n}) + \exp(-\epsilon_{13}\sqrt{n})$$

$$\leq \exp(-\epsilon_{15}\sqrt{n}),$$

where

$$c = \frac{(1 - m_1)m_5}{1 - m_1 - m_3}.$$

The last inequality holds for appropriate $\epsilon_{15} > 0$ and $n$ large enough. Continuing, we have

$$\mathbb{P}\{Z_4(T_4) \leq c \theta_7 n (1 - \sqrt{n}/n)\} \leq \exp(-\epsilon_{15}\sqrt{n})$$

which implies

$$\mathbb{P}\{Z_4(T_4) \leq c \theta_8 n\} \leq \exp(-\epsilon_{15}\sqrt{n})$$

for any $0 < \theta_8 < 1$, since both $\theta_7$ and $(1 - \sqrt{n}/n)$ can be made arbitrarily close to 1, for $n$ sufficiently large.

$\square$

**A Lower Bound on the Total Jobs in the Network.** We now wish to show that, off a set of small probability, there will be at least $n/4$ jobs in the network in $[0, T_4]$. In all previous results above, recall that all $\theta_i$ can be made arbitrarily close to one. In the following arguments, we assume that the $\theta_i$ are sufficiently close to one to suit our needs.

**Theorem 4.23.** There exists an $\epsilon_{16} > 0$ such that, for sufficiently large $n$,

$$\mathbb{P}\{|Z(t)| \geq n/4, \forall t \in [0, T_4]\} \geq 1 - \exp(-\epsilon_{16}\sqrt{n}).$$

**Proof.** On $[0, \sigma_r]$ the lower bound on $|Z(t)|$ follows directly from the proof of Theorem 4.6, as long as $\theta_2$ is close to unity. Next, let $\bar{\sigma}_r > \sigma_r$ be the time at which only $n/4$ of the original jobs from buffer 4 remain in buffers 4 and 5. Then by our definition of $\bar{\sigma}_r$ and Theorem 4.6, the lower bound in fact holds on $[0, \bar{\sigma}_r]$. Now, we need only show that, off a small set, there will be at least $n/4$ arrivals to the network during $[\sigma_r, \bar{\sigma}_r]$. If this is so, then the bound holds on $[0, T_2]$.

In $[\sigma_r, \bar{\sigma}_r]$ buffer 5 must process $(\theta_2 - 1/4)n$ jobs, off an exponentially small set. Now, by arguments analogous to the proof of Lemma 4.9, the amount of time needed to process $(\theta_2 - 1/4)n$ jobs at buffer 5 is $(\theta_9 - 1/4)m_5n$, off a small set, where $0 < \theta_9 < 1$. By arguments analogous to the proof of Theorem 4.2, the number of exogenous arrivals during $[\sigma_r, \bar{\sigma}_r]$ is $(\theta_{10} - 1/4)m_5n$ off an exponentially small set, for $0 < \theta_{10} < 1$. Note that $(0.75)m_5 = 0.3$, hence it is easy to make the expression above close to $n/4$ for $\theta_{10}$ sufficiently close to one. Now, since all jobs that arrive during $[\sigma_r, \bar{\sigma}_r]$ are in buffers 1 or 2, we have established that, for any $0 < \theta_{10} < 1$ there exists an $\epsilon_{17} > 0$ such that for all sufficiently large $n$,

$$\mathbb{P}\{Z_1(\bar{\sigma}_r) + Z_2(\bar{\sigma}_r) \geq (\theta_{10} - 1/4)m_5n\} \geq 1 - \exp(-\epsilon_{17}\sqrt{n}).$$
Thus, the lower bound on $|Z(t)|$ of the theorem holds on $[0, T_2]$. Now, we need to show that the bound holds on $[T_2, T_3]$. On this interval, which is the top cycle, the bound follows automatically from the arguments in Section 4.3. In particular Theorem 4.12 guarantees that the total number of jobs in the network during the top cycle is at least $\theta_1 m_5 n$ (off a small set), which again is easily bigger than $n/4$ for $\theta_4$ close to one.

Finally, Theorem 4.1 insures that the network does not lose too many jobs during $[T_3, T_4]$, off a small set. In particular Theorem 4.1 implies that with high probability, there are at least $\theta m_5 n$ jobs in the network during $[T_3, T_4]$. (Note that the constant $c$ which appears in the proof of Theorem 4.1 is larger than $m_5$).

We obtain the probabilistic lower bound on $|Z(t)|$ for all $t \in [0, T_4]$ by taking all of the exponential bounds together. \hfill \Box

## 5 The Exponential Network – Proof of Theorem 2.1

Finally, we need to use all of our previous results to complete the proof of Theorem 2.1. Specifically, we need to show that Theorem 4.1 implies our main result.

**Proof of Theorem 2.1.** We are now ready to prove Theorem 2.1. Our proof is similar to the proof of instability given in Bramson [2], although we provide some extra details of the method. We let $Z = \{Z(t), t \geq 0\}$ be the queue-length process for our network. In the case when there is more than one job class present at a station, we assume that $Z(t)$ has the information of which job is in service appended to it. In the non-preemptive case, this information is required to make $Z$ a Markov process. So, $Z$ is then a discrete state, continuous-time Markov process. For clarity, we will sometimes explicitly denote the dependence of $Z(t)$ on the sample path by writing $Z(t, \omega)$.

We will now prove Theorem 2.1 by contradiction. So, suppose that there exists an initial state $z_0$ such that

$$\mathbb{P}_{z_0}(\{\omega : Z(t, \omega) \not\to \infty\}) > 0,$$

where $\mathbb{P}_z(\cdot)$ is the probability measure induced when starting in state $z$. For an integer $\ell$, let $A_\ell = \{\text{state } z : |z| \leq \ell\}$. Then (21) implies that there exists an $\ell > 0$ such that

$$\mathbb{P}_{z_0}(\cap_{k=1}^{\infty} \cup_{t \in [k, \infty)} \{\omega : Z(t, \omega) \in A_\ell\}) = \delta > 0.$$  \hfill (22)

Now suppose we begin in an initial state $z_1 = (0, 0, 0, n, 0)$. Note that this state is a special case of the initial state as given in Theorem 4.1. Fix $\theta < 1$ such that $c \theta > 1$, where

$$c = \frac{(1 - m_1) m_5}{1 - m_1 - m_3}.$$  \hfill (23)

By repeatedly applying Theorem 4.1 and the strong Markov property, we have for large enough $n$,

$$\mathbb{P}_{z_1}(\{\omega : Z(t, \omega) < n/4 \text{ for some } t \geq 0\}) \leq 2 \sum_{i=0}^{\infty} \exp[-\epsilon \sqrt{n}(c \theta)^t].$$  \hfill (24)

Note in particular that the right-hand side of (24) approaches zero as $n$ goes to infinity, hence the probability on the left can be made as small as desired. Choose an $n > 4\ell$ which is large enough to satisfy Theorem 4.1 and such that the left-hand side of (24) is smaller than $\delta/2$.

One can check that any initial state of the form given in Theorem 4.1 is accessible from the zero state. Since any state can access the zero state, we have

$$\mathbb{P}_z\{\text{the Markov process } Z \text{ eventually reaches state } z_1\} > 0$$  \hfill (25)
for any initial state $z$. Because the set $A_\ell$ is finite, we have
\[
\min_{z \in A_\ell} \mathbb{P}_z \{ \text{the Markov process $Z$ eventually reaches state } z_1 \} > 0.
\]

Now for any $\omega$ in $\cap_{k=1}^\infty \cup_{t \in [k,\infty)} \{ \omega : Z(t,\omega) \in A_\ell \}$, there exists a sequence $\{t_k\}$ such that $t_k > k$ and $Z(t_k,\omega) \in A_\ell$ for $k \geq 1$. Each time the process enters $A_\ell$, it has a positive probability hitting state $z_1$. Thus, on the event $\cap_{k=1}^\infty \cup_{t \in [k,\infty)} \{ \omega : Z(t,\omega) \in A_\ell \} \cap \{ Z(T) = z_1 \}$, the process $Z$ will hit $z_1$ with probability 1. Therefore, we have
\[
\delta = \mathbb{P}_{z_0} (\cap_{k=1}^\infty \cup_{t \in [k,\infty)} \{ \omega : Z(t,\omega) \in A_\ell \}) \\
= \mathbb{P}_{z_0} (\cap_{k=1}^\infty \cup_{t \in [k,\infty)} \{ \omega : Z(t,\omega) \in A_\ell \} \cap \{ Z(T) = z_1 \}) \\
\leq \mathbb{P}_{z_1} (\cap_{k=1}^\infty \cup_{t \in [k,\infty)} \{ \omega : Z(t,\omega) \in A_\ell \}) \\
\leq \mathbb{P}_{z_1} (\{ \omega : Z(t,\omega) < n/4 \text{ for some } t \geq 0 \}) \\
\leq \delta/2,
\]
yielding a contradiction. In the second display, $T$ is the first hitting time to the state $z_1$. We obtain the first inequality via the strong Markov property. The last inequality follows from (23).

\[\Box\]

6 The Preemptive Exponential Network

In this section only, we consider the exponential network operating under the static buffer priority policy which is preemptive. Specifically, the network has two single-server processing stations and five job classes, as depicted in Figure 1. All interarrival and service times are again assumed to be i.i.d. exponential random variables. The network has the same arrival rate and mean service times as before:

\[\alpha_1 = 1, \quad m_1 = 0.4, \quad m_2 = 0.1, \quad m_3 = 0.4, \quad m_4 = 0.1 \quad \text{and} \quad m_5 = 0.4.\]

Also, the network operates under the same non-idling static buffer priority service policy $\pi = \{(1,3,4),(5,2)\}$. However, we now assume that the service policy is preemptive resume, i.e., if a higher priority job arrives during the service of a lower priority job, then the lower priority job is preempted and the higher priority job immediately receives service. Hence the highest priority jobs, class 1 jobs at Station $A_1$ and class 5 jobs at Station $B$, will never experience a service delay due to jobs of any other class. Jobs which are preempted complete service where they left off, after all higher priority jobs have been served. Again, we let $|Z(t)|$ denote the total number of jobs in the network at time $t$. We will refer to this network as the preemptive exponential network.

For the preemptive exponential network, it turns out that an analog of our main result for the non-preemptive network (Theorem 2.1) holds:

**Theorem 6.1.** For the exponential network operating under the preemptive SBP service policy, starting from any initial state,

\[|Z(t)| \to \infty\]

as $t \to \infty$ with probability one.

In this section, we only provide an outline for the proof of Theorem 6.1. The full proof of the theorem proceeds in an exactly analogous manner to the proof of Theorem 2.1, but in fact can be
considerably simplified due to the preemptive assumption. However, the basic idea is the same: use simple large deviations estimates to show that the queueing network will roughly follow the unstable fluid behavior (outlined in Section 3), with high probability.

**Proof Outline for Theorem 6.1.** As in the non-preemptive case, the majority of the proof of Theorem 6.1 involves proving a theorem similar to Theorem 4.1. The following theorem can be proven more directly than Theorem 4.1 due to the preemption employed.

**Theorem 6.2.** Consider the exponential network operating under the preemptive SBP service policy. Suppose $Z(0) = (0, z_2, 0, n, 0)$. Then for any $0 < \theta < 1$, there exist an $\epsilon > 0$ and a Markov time $T_4$ with $Z(T_4) = (0, Z_2(T_4), 0, Z_4(T_4), 0)$ such that for all sufficiently large $n$,

$$
P \left\{ Z_4(T_4) \geq \frac{(1 - m_1)m_5}{1 - m_1 - m_3} \theta n \right\} \geq 1 - \exp(-\epsilon \sqrt{n}).$$

where

$$
T_2 = \inf \{ t > 0 : Z_3(t) = Z_4(t) = Z_5(t) = 0 \},
$$

$$
T_4 = \inf \{ t > T_2 : Z_1(t) = Z_3(t) = Z_5(t) = 0 \}.
$$

Furthermore, for all sufficiently large $n$,

$$
P \{ |Z(t)| \geq n/4, \quad \forall \ t \in [0, T_4] \} \geq 1 - \exp(-\epsilon \sqrt{n}).$$

The main result in the preemptive case, Theorem 6.1 follows directly from Theorem 6.2, as in the non-preemptive case.

Next, we briefly outline the arguments needed to establish Theorem 6.2. As before, one needs to employ basic large deviations estimates to show that, with high probability, very few jobs will “leak” from buffer 4, before the network enters the last busy period in the bottom cycle. This is analogous to the result of Theorem 4.6. However, in the preemptive case the arguments can be simplified considerably. During the analogous impure periods in the preemptive case, jobs cannot be processed or “leaked” from buffer 5. Hence, it is not necessary to derive bounds for such impure periods or jobs leaked during these periods. The main reason for the simplification is the fact the jobs can never be processed simultaneously at buffers 3 and 5 in the preemptive network. Now, once a modified Theorem 4.6 has been established, the analogous result to Theorem 4.2 follows exactly as in the non-preemptive case.

For the second half of the proof (the top cycle), the preemptive case similar, yet simpler. Once again, establishing a result like Theorem 4.12 is easier because leaks from buffer 2 cannot occur during top cycle impure periods in the preemptive case. Otherwise, the arguments for the top cycle in the preemptive case are exactly analogous to the non-preemptive case. □

### 7 Virtual Station, Push Start and the Companion Paper

In this section, we give a more detailed discussion of virtual station and push start conditions. We will also provide some insights as to why the results in this and the companion paper hold.

As was mentioned in Section 3, an unstable fluid model solution exists because the push start condition

$$
\rho_{push} = \frac{m_3}{1 - m_1} + m_5 \leq 1
$$

is violated. The push start condition is a magnification of a virtual station condition identified in Dai and Vande Vate [14]. The virtual station effect can most easily be seen in the queuing
network operating the preemptive SBP service policy. The following proposition is a special case of Proposition 3.1 of [18]. Note that the result holds even if we allow the occurrence of simultaneous events, as long as they are processed one at a time, consistent with the given preemptive SBP service policy.

**Proposition 7.1**. For the 2 station, 5 class queueing network under any distributional assumptions on interarrival and service times, assume the preemptive SBP service policy is employed. Then, for every sample path,

$$Z_3(t)Z_5(t) = 0 \quad \text{for all } t \geq 0$$

(25)

as long as $Z_3(0)Z_5(0) = 0$.

Let us suppose that the network is initially empty. A consequence of equation (25) is that jobs in classes 3 and 5 can never be processed simultaneously. Thus, one can envision that these two classes constitute a virtual station, with at most one class being served at a time. Therefore,

$$m_3 + m_5 \leq 1$$

(26)

is necessary for the stability of the queueing network and the corresponding fluid model. Condition (26) is called a virtual station condition.

To describe the push start condition, we consider a 2-station, 4-class queueing network obtained by deleting class 1 from the 5-class network in Figure 1. We retain the priority scheme from the 5-class network. The resulting 4-class network is the well known Lu-Kumar network [23]. For simplicity, in the 4-class network, we retain the class designations from the 5-class network. For example, jobs in the first step of processing in the Lu-Kumar network are labeled as class 2 jobs. One can show that (25) continues to hold in the Lu-Kumar queueing network. Thus, (26) is necessary for the stability of both the Lu-Kumar queueing network and the Lu-Kumar fluid model.

Now let us consider the fluid model of our 5-class network. Since class 1 has highest priority, and $\alpha_1 = 1 < \mu_1$, buffer 1 will empty in finite time and will remain empty thereafter. In keeping buffer 1 empty, server A spends $\alpha_1 m_1 = 40\%$ of its effort on class 1 fluid. The remaining $1 - m_1$ of its effort can be spent on fluid in classes 3 and 4. Since buffer 1 will remain empty, one would like to delete the buffer from our analysis in the 5-class fluid model. The resulting fluid model, after the deletion, is identical to the Lu-Kumar fluid model, except that mean processing time at classes 3 and 4 need to be expanded by a factor of $1/(1 - m_1)$. The necessity condition (26) in the Lu-Kumar fluid model leads to the necessity condition (24) in our 5-class fluid model.

As we have seen, the derivation of the push start condition (24) relies on the following factors: (a) the service policy is preemptive so that the virtual station phenomenon (25) occurs; (b) fluid model analysis is used to fully exploit the push start effect. For the queueing network operating under the SBP service policy, the push start condition (24) may not be necessary for its stability, either because a non-preemptive policy is used or the queueing network cannot realize the push start effect as demonstrated in the fluid model. In this paper, we show that, for the exponential network, even under the non-preemptive SBP policy, the push start condition is necessary. In the companion paper, we show that, for a deterministic network operating under the non-preemptive SBP policy, the network is stable even the push start condition is violated.

In the queueing network, in order for the push start phenomenon to have a full effect as in the fluid model, there must be some independence (loosely defined) between arrival times to buffer 1 and times when buffer 3 is positive. Although it is somewhat hidden in our analysis and proofs, the basic reason that this push start effect holds in the exponential network is that arrivals to the network are Poisson. The push start effect is precisely demonstrated in Lemma 4.19, for the
non-preemptive case. A similar lemma can be proven, using slightly different techniques, in the preemptive case. If the Poisson arrival assumption is removed, then the push start effect need not hold, or at least the magnifying factor may not be the same.

Under the non-preemptive SBP policy, one cannot expect the full virtual station phenomenon, as in (25), to occur. Rather, in this case we have a “partial” virtual station effect.

**Proposition 7.2.** For the 2-station, 5-class queueing network under any distributional assumptions on interarrival and service times, assume the non-preemptive SBP service policy is employed. Then, for every sample path,

\[ [Z_3(t) - 1]^+ \cdot [Z_5(t) - 1]^+ = 0 \quad \text{for all } t \geq 0 \]

as long as \([Z_3(0) - 1]^+ \cdot [Z_5(0) - 1]^+ = 0\). Here \(x^+ = \max\{x, 0\}\) for a real number \(x\).

The proposition asserts that, under the non-preemptive SBP policy, only one of the two buffers can have more than 1 job at any time. However, the “partial” virtual station effect in Proposition 7.2 is not sufficient to cause instability, even though the virtual station condition (26) is violated. In particular, if buffers 3 and 5 are both processing jobs a large portion of the time, the network will in fact be stable, given that the usual traffic conditions hold. This is the crucial difference between the networks considered in this paper and the companion paper. The key feature, demonstrated in the instability proof for the non-preemptive exponential network, is that although buffers 3 and 5 may sometimes process jobs simultaneously, *they will only do so a small percentage of the time*. This key feature is missing in the deterministic analog considered in the companion paper. In the deterministic non-preemptive case, the network will always eventually reach a state where buffers 3 and 5 are processing jobs simultaneously a large percentage of the time. At this point, we do not know the exact source of the non-necessity of push start condition for the deterministic network. The non-preemption makes the virtual station less tight, but it could also make the push start factor smaller. Identifying the exact source of non-necessity is a future research topic.

A major question which arises from the instability results we have demonstrated in the exponential case, is if the result can be generalized to any larger class of networks. We conjecture that the results can indeed be extended, at least for two station multiclass queueing networks. A plausible conjecture is that for two-station reentrant lines with exponential interarrival and service times, the virtual station and push-start conditions of Dai and Vande Vate [14] are necessary and sufficient for global stability. In fact, it is likely that such a result holds for the broader class of networks considered in Hasenbein [19]. In light of the results in Bramson [5], it is unclear if such a principle holds for networks with more than two stations.

**Acknowledgments:** The authors would like to thank Maury Bramson for providing valuable input in the course of this research. A portion of this research was performed while John Hasenbein was visiting the Centro de Investigación en Matemáticas (CIMAT) in Guanajuato, Mexico. He would like to thank CIMAT and the National Science Foundation, who jointly provided an International Research Fellowship which made this visit possible.

**References**


A Appendix

A.1 Large Deviations Estimates

We repeatedly use the following large deviations estimates in the proofs of Section 4. An elementary proof of the lemma can be found in Shwartz and Weiss [27], Section 1.2.

Lemma A.1. Let \( X_1, X_2, \ldots \) be an i.i.d. sequence of non-negative random variables with mean \( \mathbb{E}(X_1) = m \). Set \( Y_n = X_1 + \cdots + X_n \). Suppose further that the \( X_i \) possess exponential moments, i.e., there exists a constant \( \kappa > 0 \) such that

\[
\mathbb{E} \left[ \exp(\kappa X_1) \right] < \infty.
\]

Then for every \( \alpha > 0 \), there exists an \( \epsilon > 0 \), so that for all \( n \geq 1 \),

(i) \[
\mathbb{P} \{ Y_n > mn + \alpha n \} \leq \exp(-\epsilon n).
\]

(ii) \[
\mathbb{P} \{ Y_n < mn - \alpha n \} \leq \exp(-\epsilon n).
\]
A.2 An Estimate for $G/G/1$ Queue

Lemma A.2. Consider a $GI/GI/1$ queue with i.i.d. interarrival times $\{u_i\}$ and i.i.d. service times $\{v_i\}$. Assume that $E[u_1] < E[v_1]$, the queue is empty at time zero, and that the first arrival occurs at time 0. Assume further that for some $\kappa > 0$, $E[\exp(\kappa(u_1 + v_1))] < \infty$. Then for $0 < \delta < 1$, there exists a constant $\epsilon > 0$ such that for $n$ sufficiently large (in particular we take $\lceil \delta \sqrt{n} \rceil > 0$),

$$
\mathbb{P}\{\text{queue first empties in } [S_{\lceil \delta \sqrt{n} \rceil}, S_n]\} \leq \exp(-\epsilon \sqrt{n}),
$$

where $S_n$ is the arrival time of the $n$th job.

Proof.

\[
\begin{align*}
\mathbb{P}\{\text{queue first empties between the } i\text{th and } (i+1)\text{st arrival}\} &= \mathbb{P}\{u_2 < v_1, u_2 + u_3 < v_1 + v_2, \ldots, u_2 + \ldots + u_i < v_1 + \ldots + v_{i-1}, \\
&\quad u_2 + \ldots + u_{i+1} > v_1 + \ldots + v_i\} \\
&\leq \mathbb{P}\{u_2 + \ldots + u_{i+1} > v_1 + \ldots + v_i\} \\
&= \mathbb{P}\{(v_1 - u_1) + \ldots + (v_i - u_i) < 0\} \\
&\leq \exp(- \epsilon_1 n) \quad \text{for all } i \text{ for some } \epsilon_1 > 0.
\end{align*}
\]

The last inequality follows from Lemma A.1 and the fact that $E[v_1 - u_1] > 0$. Thus,

\[
\mathbb{P}\{\text{queue first empties in } [S_{\lceil \delta \sqrt{n} \rceil}, S_n]\} \leq \sum_{i=\lceil \delta \sqrt{n} \rceil}^{n} \exp(-\epsilon_1 i) \\
\leq n \exp(-\epsilon_1 \lceil \delta \sqrt{n} \rceil) \\
\leq \exp(-\epsilon_2 \lceil \delta \sqrt{n} \rceil) \quad \text{for large enough } n \\
\leq \exp(-\epsilon \sqrt{n}).
\]

The last inequality is possible because for large $n$, $\lceil \delta \sqrt{n} \rceil / \sqrt{n}$ can be bounded away from zero, allowing us to pick an $\epsilon$ which satisfies the inequality for all large $n$. \qed