Stability and Instability of a Two-Station Queueing Network

J. G. Dai[1]
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[2]
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

John H. Vande Vate[3]
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205, USA
john.vandevate@isye.gatech.edu

Abstract

This paper proves that the stability region of a 2-station, 5-class reentrant 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, under a particular policy. 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. We show that the same network with all interarrival and service times being deterministic is stable. When all distributions are uniform with a given range, our simulation studies show that the stability of the network depends on the width of the uniform distribution. Finally, we show that the same network, with deterministic interarrival and service times, is unstable when it is operated under the *preemptive* version of the static buffer priority service policy. Thus, our examples also demonstrate that the stability region depends on the preemption mechanism used.

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

Short Title: A Two-station Queueing Network

Revision: July 25, 2002

1 Introduction

This paper 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 [28], Dai [8], Stolyar [30], Dai and Meyn [11], Chen [6], Meyn [25], Dai [10], Bramson [4, 5] and Pulhaskii

[1]Research supported in part by NSF grants DMI-94-57336 and DMI-9813345 and by the Logistics Institute - Asia Pacific (TLI-AP), a partnership between the National University of Singapore and the Georgia Institute of Technology.
[2]Research supported in part by NSF grants INT-9971484 and DMI-0132038.
[3]Research supported in part by NSF grant DMI-9813345.

and Rybko [27]. 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, 13, 20, 23]. Recently, Bramson [5] gave an example of a queueing network that is stable, but whose fluid model is not stable.

The queueing network studied in this paper has 2 service stations and 5 job classes. It belongs to a special class of networks called reentrant lines by Kumar [22]. 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. We first consider a model in which all interarrival and service time distributions are assumed to be deterministic. Theorem 2.1 posits that the deterministic network is stable. We next analyze the network under the assumption that all interarrival and service time distributions are exponential. In Theorem 2.2, we prove that such a queueing network is unstable in the sense that, with probability 1, the total number of jobs in the system goes to infinity with time. A consequence of Theorems 2.1 and 2.2 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 [13]. 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 [13], 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 10 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. [12] and [19]) 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 considered in this paper provides a test case for determining the necessity of push start conditions. For the exponential queueing network, Dai and Vande Vate's virtual station condition is satisfied. However, the push start condition is violated. Theorem 2.2 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 holds 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 here will be of use in establishing a more general result of this type.

The discussion so far in this section has assumed that the SBP service policy is non-preemptive. In Theorem 2.3, however, we give a result for the deterministic network operating under the preemptive SBP service policy. In particular, we show that the network is unstable when operated under the preemptive SBP policy, while Theorem 2.1 proves that the same network operating under the non-preemptive SBP policy is stable. Our finding contrasts a common belief that the preemption mechanism should have little effect on system performance measures like throughput and cycle time, at least when the system is heavily loaded [17]. Indeed, Williams [32] shows that the heavy traffic behavior under any HOL service policy is insensitive to the preemption mechanism employed as long as a certain multiplicative state space collapse condition is satisfied. Our example suggests that the state space collapse condition itself may depend on the preemption mechanism used.

To our knowledge, this paper is the first to demonstrate that the stability region of a standard multiclass queueing network operating under an HOL service policy depends 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], [30]) 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 [31] 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. Heretofore, 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 our main results. In Section 3, we introduce the fluid model which corresponds to the queueing network described in Section 2. In this section we also present a stable fluid solution which drains and an unstable fluid solution that diverges to infinity, where both solutions have the same initial state. Section 4 presents the proof of Theorem 2.1, regarding the deterministic network. Sections 5 and 6, which occupy a large portion of the paper, are devoted to the proof of Theorem 2.2. For the casual reader, these two sections can be skipped. In Section 7 we present some simulation results which explore the stability behavior of the uniform network. Section 8 contains the proof of Theorem 2.3 and Section 9 gives a proof outline of a result that is analogous to Theorem 2.2, when the exponential network is operated under the preemptive SBP service policy. Finally, in Section 10, we present a more detailed discussion of virtual station and push start conditions. We

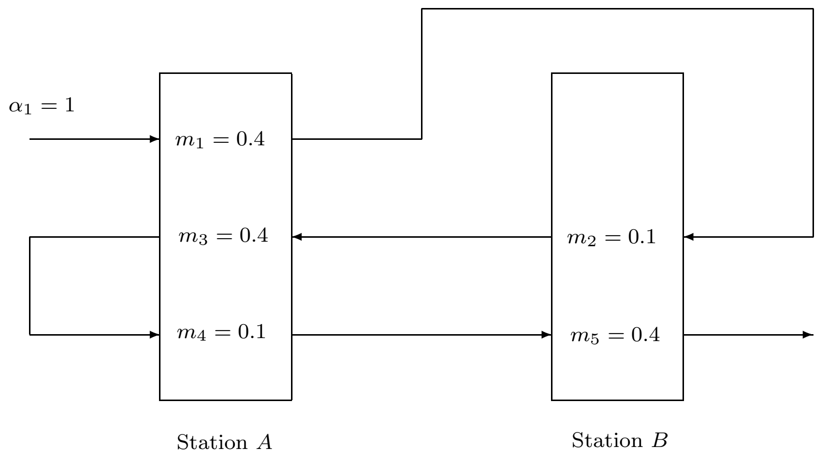

Figure 1: A Push Started Lu-Kumar Network

end the paper with an appendix in which details of some proofs are presented.

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 our main results.

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. We assume that the service times for class k jobs are i.i.d. 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. random variables with mean $1/\alpha_1$. Thus, α_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

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

In this paper, we consider several variations of the general network described above. In all cases, the interarrival time distribution and five service time distributions are assumed to all have the same type of distribution, either deterministic, exponential, or uniform. We will refer to the associated network as either the deterministic network, the exponential network, or the uniform network, respectively. Since we have fixed the mean interarrival and service times, in the deterministic case, the means completely define the respective distributions. In the exponential network, all intearrival and service time distributions are assumed to be exponential with mean values as specified above. For a uniform network, to specify the distributions we need to specify the supports of these uniform distributions. Each uniform distribution is centered at the mean values above, with a width of ϵ. For ease of exposition, we assume that the widths for the interarrival and the five service time distributions are all the same. Thus, for the uniform networks we discuss, we can fully specify the distributions with the parameter ϵ, since the mean values are fixed.

Now we discuss the service policy to be employed in the network under study. 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 high priority and class 2 jobs have low priority. We can consider both non-preemptive and preemptive versions of the policy π. 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. We primarily consider the non-preemptive service policy. However, in Sections 8 and 9 we give some results on the network operating under the preemptive policy.

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. Namely, if one knows $Z(t)$ at time t, the future evolution of the network can be determined from time t on. For the deterministic or uniform networks, the state $Z(t)$ is not sufficient to determine the future evolution of the network. One also needs to know the remaining interarrival and service times at t to completely specify the state of the system. Furthermore, in the non-preemptive case, one must also specify which job is currently in service, no matter what distributional assumption is in effect. In later sections, we will sometimes augment the buffer level state $Z(t)$ with additional information needed to fully specify the network state.

2.2 Main results

Here we present the main theoretical results and also summarize some of our simulation results. The first result shows that, under the non-preemptive SBP service policy the deterministic network is stable (from all initial states). The second result shows that the number of jobs in the exponential network diverges to infinity, from any initial state. Finally, we show that the deterministic network operating under the preemptive SBP policy is unstable from at least one class of states. The first two results show that changing the distribution for the interarrival and service times has a profound effect on network dynamics. The first and third results demonstrate that changing the preemption mechanism also has a dramatic effect. Finally, our simulation results indicate that simply changing the range of the distributions in a uniform network also affects the stability of the network. We now state the main results more precisely.

The first theorem concerns the deterministic network operating under the non-preemptive SBP service policy. In the deterministic network, a fully specified system state should indicate the number of jobs in each class, which classes of jobs are in service, the remaining time until the next job arrives, and the remaining service times for all jobs in service. For notational convenience, we use some subset of this information to define a "special state" $(z_1, z_2, z_3, z_4, z_5; a)$, where z_k is the number of jobs in buffer k and a is the remaining interarrival time until the next job arrives to the network. These special states, observed at service completion times, indeed indicate the complete system state due to the deterministic evolution of the network. With a slight abuse of terminology, such special states are simply called states.

In Lemma 4.1 we verify that for any $0 < a \leq 0.1$ when the deterministic network starts in state

$$(0, 0, 0, 1, 0; a),$$

the network will come back to this same state exactly one minute later. Thus, the trajectory starting from state $(0, 0, 0, 1, 0; a)$ forms an orbit. For a given $0 < a \leq 0.1$, the corresponding orbit is called an a-orbit. With this definition in hand, we now state our first result.

Theorem 2.1. *For the deterministic network operating under the non-preemptive SBP service policy, starting from any state, there exists a finite time at which the network enters an a-orbit, with $0 < a \leq 0.1$.*

As we demonstrate later, while in an a-orbit, the network has at most two jobs. Hence, a consequence of Theorem 2.1 is that the total number of jobs in the deterministic network is at most 2 after some finite time (which depends on the initial state).

The next theorem shows that the number of jobs in the exponential network diverges to infinity.

Theorem 2.2. *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.

One can adopt a number of definitions of stability for a network. We do not adopt any specific definition in this paper, but present some different notions of stability in order to put our results in perspective.

Definition 2.1. The 2-station queueing network is said to be *bounded in probability* if starting from any initial state x,

$$\lim_{M\to\infty} \liminf_{t\to\infty} \mathbb{P}_x \{|Z(t)| \leq M\} = 1.$$

For the exponential network, being stable in the sense of bounded in probability is equivalent to the state process for the network being positive recurrent. See, for example, the proof of Proposition 18.3.1. of Meyn and Tweedie [26] in the discrete time case. (The proof in the continuous time case is analogous.) For the deterministic network, bounded in probability is equivalent to the recurrence property of the trajectory. Hence, if we adopt bounded in probability as the definition for stability, then Theorem 2.1 and Theorem 2.2 show that the exponential network operating under the non-preemptive SBP policy is unstable while the deterministic network operating under the same policy is stable.

Definition 2.2. The 2-station queueing network is said to be *rate stable* if starting from any initial state x,

$$\mathbb{P}_x \left\{ \lim_{t\to\infty} D_5(t)/t = \alpha_1 \right\} = 1,$$

where $D_5(t)$ is the number of jobs which have departed the network in $[0, t]$.

Although it is not proved in this paper, we suspect that Theorem 2.2 can be strengthened to show that, with probability one, the number of jobs in the network grows linearly with time. Such a result would imply that the exponential network is not rate stable, whereas Theorem 2.1 implies that the deterministic network is rate stable. The interested reader should refer to Chen [6] or El-Taha and Stidham [15] for more discussion on rate stability. El-Taha and Stidham use the term *pathwise stability* instead of rate stability.

Since the deterministic network is an extreme case where there is no randomness at all in the system, one may wonder if the fact that the stability of the network depends on the interarrival and service distributions is a robust phenomenon or simply a pathological result which relies on the special deterministic case. In Section 7, we provide some simulation studies which indicate that the uniform network is stable for $\epsilon = 0.001$ and unstable for $\epsilon = 0.1$.

Our final theorem implies that by allowing preemption in our SBP service policy, the deterministic network becomes unstable.

Theorem 2.3. *For the deterministic network operating under the preemptive SBP service policy, the number of jobs in the system grows linearly to infinity with time for any initial state* $Z(0) = (0, 0, 0, n, 0)$ *for* n *sufficiently large.*

This theorem implies that the deterministic network operating under the preemptive policy is unstable, in both the stochastic boundedness and rate stable sense. So this result, along with Theorem 2.1, shows that the stability of the network depends on the preemption mechanism employed. Note that the exponential network is unstable, independent of the preemption mechanism employed (see Theorem 9.1, along with Theorem 2.2 from above).

3 Fluid model solutions

In this section we introduce the fluid model of our queueing network. The fluid model is especially important for our results since fluid model solutions give insight into the behavior of the networks we study. In addition to defining the fluid model below, we present both a stable and unstable fluid solution to the model. It turns out that the stable fluid solution and the stable trajectories of the deterministic network have the same qualitative behavior. Also, the proof of Theorem 2.2, given in Sections 5 and 6, is related to the unstable fluid solution. Essentially, the main idea of the proof is to show that the exponential queueing network dynamics closely follow this unstable fluid model solution.

The fluid model is a deterministic, continuous analog of the queueing network. It is defined through the following set of equations:

$$Z_1(t) = Z_1(0) + \alpha_1 t - \mu_1 T_1(t), \quad t \geq 0, \tag{2}$$

$$Z_k(t) = Z_k(0) + \mu_{k-1} T_{k-1}(t) - \mu_k T_k(t), \quad t \geq 0, \quad k = 2, \ldots, 5, \tag{3}$$

$$Z_k(t) \geq 0, \quad t \geq 0, \quad k = 1, \ldots, 5, \tag{4}$$

$$T_k(t) \text{ is non-decreasing in } t, \quad k = 1, \ldots, 5, \tag{5}$$

$$t - T_k^+(t) \text{ is non-decreasing in } t, \quad k = 1, \ldots, 5, \tag{6}$$

$$\dot{T}_k^+(t) = 1 \text{ for any time } t \text{ with } Z_k^+(t) > 0 \text{ for } k = 1, \ldots, 5, \tag{7}$$

where, $\mu_k = 1/m_k$, $Z_k^+(t)$ is the sum of $Z_\ell(t)$ over all classes ℓ 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, $\dot{f}(t)$ 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.

We construct two different fluid solutions, one which drains and one which diverges to infinity, from the same initial state. These two drastically different fluid solutions may give some insight as to why two queueing networks with the same fluid model may also have drastically different behavior.

In the exposition that follows, it should be kept in mind that at many time points (not just at time zero), fluid solutions may bifurcate. Hence, there are infinitely many stable and unstable solutions from some initial states. We only describe one feasible stable solution which is consistent with the fluid model equations, from a particular initial state. The same point holds for the unstable fluid solution described later in this section, i.e. we only describe one feasible unstable solution from this initial state.

A stable fluid solution. We first construct a fluid solution that drains from the initial state $Z(0) = (0, 0, 0, 1, 0)$, i.e. for this solution we have that $Z(t) = 0$ for all $t > T$, with $T < \infty$. In fact, this initial state is not special, nor is the specific set of network parameters we are using. It can be shown that as long as the *usual traffic conditions*:

$$\rho_A := \alpha_1(m_1 + m_3 + m_4) < 1 \quad \text{and} \quad \rho_B := \alpha_1(m_2 + m_5) < 1, \tag{8}$$

hold, then we can similarly construct a stable fluid solution from any initial state. If one of the usual traffic conditions is violated, then no stable fluid solution exists, from any initial fluid state.

To construct the stable fluid solution, we start the system with initial fluid level $Z(0) = (0, 0, 0, 1, 0)$. The solution is qualitatively divided into two periods. During the first period, the network drains fluid from buffer 4, and does not accumulate fluid in any other buffers. Once the fluid is drained from buffer 4 at time t_1, the network can maintain all buffers empty from t_1 on, which is the second part of the fluid solution.

We use $d_k(t)$ to denote the departure rate $\mu_k \dot{T}_k(t)$ from class k at time t. Note that if we fully specify the departure rates $d_k(t)$, for $k = 1, \ldots, 5$ and for all t, a resulting fluid solution (T, Z) is uniquely defined. One needs to check that the solution satisfies the fluid model equations (2)-(7). In the case of our solution, this is relatively easy to verify.

So, we first consider the time interval $[0, 1)$. For $t \in (0, 1)$ we set

$$d_1(t) = d_2(t) = d_3(t) = 1 \quad \text{and} \quad d_4(t) = d_5(t) = 2.$$

Since, under this set of departure rates, it is clear that only $Z_4(t)$ is positive on $[0, 1)$, we only need to check (7) for $k = 4$. One can check that for $k = 4$, (7) is equivalent to

$$m_1 d_1(t) + m_3 d_3(t) + m_4 d_4(t) = 1, \tag{9}$$

which clearly holds for the departure rates we have specified. To validity of (6) follows from (9) and the fact that

$$m_2 d_2(t) + m_5 d_5(t) \leq 1$$

holds for all $t \in [0, 1)$ for the specified departure rates. The remainder of the fluid model equations are easily verified.

Under the departure rates given, it is clear that buffer 4 will empty at time $t_1 = 1$, with all other buffers remaining empty. So, at t_1, we have $Z(t_1) = (0, 0, 0, 0, 0)$.

Next, on the interval (t_1, ∞) we set $d_k(t) = 1$ for $k = 1, \ldots, 5$, which yields

$$Z(t) = (0, 0, 0, 0, 0) \quad \text{for } t \in [t_1, \infty)$$

Again, it is easy to check that (2)-(7) are satisfied for these departure rates and fluid buffer levels. Hence, we have demonstrated that there exists a stable fluid solution, starting from $Z(0) = (0, 0, 0, 1, 0)$.

An unstable fluid solution. Now we construct a fluid solution that diverges to infinity. Most of the proof of Theorem 2.2 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

$$\rho_{push} := \alpha_1 m_5 + \alpha_1 \frac{m_3}{1 - \alpha_1 m_1} > 1. \tag{10}$$

When $\rho_{push} \leq 1$ and the usual traffic conditions hold, then no divergent fluid solution exists. Condition (10) violates the *push start* condition, first identified in Dai and Vande Vate [13]. The push start condition is a magnification of a *virtual station* phenomenon first observed by Harrison and Nguyen [18] and Dumas [14] and later systematically treated in Dai and Vande Vate [12] and [13]. See Section 10 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 $Z(0) = (0, 0, 0, 1, 0)$. We present a fluid solution in one period that ends when the system state reaches a state $(0, 0, 0, +, 0)$ 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. Again, we describe the fluid solution (T, Z) by specifying the departure rates $d_k(t)$, for $t \geq 0$ and $k = 1, \ldots, 5$. When the time t is clear from the context, we drop the time dependence from the departure rate notation. 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 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 α_1. The initial amount in the pipe at t_2 is $\alpha_1 m_5$. Thus, the pipe must empty in time

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

Hence,

$$Z_4(t_4) = \alpha_1 m_5 + \alpha_1(t_4 - t_2) = \frac{\alpha_1 m_5}{1 - \alpha_1 m_3/(1 - \alpha_1 m_1)} = \frac{(1 - m_1) m_5}{1 - m_1 - m_3} = \frac{6}{5} > 1. \tag{11}$$

Therefore, our top cycle ends at time t_4 with fluid level $(0, 0, 0, 6/5, 0)$. One can check that in (11), $Z_4(t_4) > 1$ is equivalent $\rho_{push} > 1$, for general mean service times. Whenever the usual traffic conditions hold and $\rho_{push} > 1$, our construction always leads to a divergent fluid solution.

4 The Deterministic Network with Non-preemption

In this section, we prove Theorem 2.1. Recall that the network has deterministic interarrival and service times, and is operated under the non-preemptive SBP service policy.

Our first lemma justifies our earlier definition of an a-orbit.

Lemma 4.1. *Starting from an initial state* $(0,0,0,1,0;a)$ *with* $0 < a \leq 0.1$*, the trajectory of the network returns to the same state one minute later.*

Proof. The proof follows from simply examining the sequence of states the network visits in the first minute:

Time	State
0	(0, 0, 0, 1, 0; a)
a	(1, 0, 0, 1, 0; 1)
0.1	(1, 0, 0, 0, 1; a+0.9)
0.5	(0, 1, 0, 0, 0; a+0.5)
0.6	(0, 0, 1, 0, 0; a+0.4)
1.0	(0, 0, 0, 1, 0; a).

□

Theorem 2.1 asserts that the network enters an a-orbit from any initial state. We first prove the theorem for initial states that are regular type 1 or type 2 states, which will be defined shortly. We then prove that the network, starting from any initial state, will eventually reach a regular state that is either of type 1 or type 2, thus proving the main theorem.

Definition 4.1.

1. A *regular state* is a state which is reachable after the network has been in operation for at least one minute.

2. A *type 1 state* is any state of the form $(0,0,0,n,0;a)$ with $n \geq 0$ and $0 < a \leq 1$.

3. A *type 2 state* is any state of the form $(0,1,0,n,0;a)$ with $n \geq 0$ and $0 < a \leq 1$.

Using Definition 4.1, it is easy to check that the following result holds.

Lemma 4.2.

1. *For a regular type 1 state,* $n \geq 1$ *and* $0 < a \leq 0.1$*.*

2. *For a regular type 2 state,* $0 < a \leq 0.6$*.*

Proof. Part 1: Consider a fixed time $t_1 \geq 1$ and suppose $Z(t_1) = (0,0,0,n,0;a)$ with $n \geq 1$ and $0 < a \leq 1$. Note that in the time interval $(t_1 - 1, t_1]$ exactly one job must have arrived from the outside. Furthermore, this job must still be in the system, since the total processing time to complete all steps is 1.4 minutes. Thus, we have $|Z(t_1)| \geq 1$, which implies $n \geq 1$. Next, the job that arrived in $(t_1 - 1, t_1]$ must have completed processing at steps one through three, since there are no jobs present in those buffers. Since these steps require 0.9 minutes, the job must have arrived at or before $t_1 - 0.9$ and after $t_1 - 1$, by assumption. Thus, we have $0 < a \leq 0.1$. *Part 2:* Consider a fixed time $t_2 \geq 1$ and suppose $Z(t_2) = (0,1,0,n,0;a)$ with $n \geq 0$ and $0 < a \leq 1$. Once again, we note that in the time interval $(t_2 - 1, t_2]$ exactly one job must have arrived from the outside. Since there are no jobs in buffer 1, this job must have completed processing at buffer 1. Since processing at buffer 1 requires 0.4 minutes, the job must have arrived at or before $t_2 - 0.4$. Thus, we have $0 < a \leq 0.6$.

□

Without of loss of generality, we assume from now on that all initial states must be regular.

Our first lemma shows that if the network starts from a regular type 1 state it will enter an a-orbit in a very direct manner. Note that if the network starts from any of the intermediate states in the proof of Lemma 4.3, it will enter an a-orbit.

Lemma 4.3. *Starting from a regular type 1 state, the network enters an a-orbit in* $(n-1)$ *minutes.*

Proof. For $n = 1$, the proof follows from our definition of an a-orbit. For $n \geq 2$, the proof follows by observation of the following sequence of states and induction.

Time	State
0	(0, 0, 0, n, 0; a)
a	(1, 0, 0, n, 0; 1)
0.1	(1, 0, 0, n-1, 1; a+0.9)
0.5	(0, 1, 0, n-1, 0; a+0.5)
0.6	(0, 0, 1, n-2, 1; a+0.4)
1.0	(0, 0, 0, n-1, 0; a).

□

The next lemma indicates what occurs when the network starts from a regular type 2 state.

Lemma 4.4. *Starting from any regular type 2 state, the network eventually enters an a-orbit.*

The proof of Lemma 4.4 is left to the Appendix. Our final lemma in this section describes what happens from a general initial state. Note that such an initial state can be assumed to be regular, hence any subsequent state is also regular.

Lemma 4.5. *Starting from any (regular) initial state, the network eventually enters either a regular type 1 or regular type 2 state.*

Proof. First, we set $r \equiv \inf\{t \geq 0 : Z_1(t) = 0\}$, i.e. r is the first time that buffer 1 is empty. We claim that r is finite from any initial state and that $Z_1(t) \leq 1$ for all $t \geq r$. To see this, note that since class 1 jobs have highest priority and class 1 jobs are processed faster than the rate of arriving jobs, there will be some finite time at which $Z_1(t) = 0$. Next, after buffer 1 has drained for the first time, any class 1 arrival will begin processing no more than 0.4 minutes after its arrival (it may be delayed 0.4 minutes to wait for a class 3 job to complete processing). Since arrivals occur every minute, no more than one job can be in buffer 1 after it has drained for the first time.

Now, set $t_1 \equiv \inf\{t \geq r : Z_2(t) + Z_5(t) = 0\}$, i.e. t_1 is the first time that Station B is empty, after buffer 1 has drained for the first time. By Lemma A.1 of the Appendix, t_1 is finite. Furthermore, from our arguments above, we have $Z(t_1) = (1, 0, m, n, 0; a)$ or $Z(t_1) = (0, 0, m, n, 0; a)$, where m and n are arbitrary nonnegative integers and $0 < a \leq 1$. Note that the job in service at Station A may be in the middle of service at t_1.

Next, to complete the proof, we examine the following cases.

Case 1. A class 1 or class 3 job is in service at time t_1. In this case, no class 4 job can be processed until buffers 1 and 3 are both empty, due to our priority scheme.

Let us denote by t_2 the first time after t_1 that a buffer 4 job enters service. Then immediately before t_2, Station A was serving either a class 1 or a class 3 job, both of which require 0.4 minutes of processing time. Note that no buffer 5 jobs are processed during $[t_1, t_2]$, which implies that $Z_2(t) \leq 1$, during this interval. If Station A was serving a class 3 job just before t_2, then $Z_2(t_2) = 0$ since there are no arrivals to buffer 2 during the 0.4 minute processing time of the class 3 job and class 2 jobs require only 0.1 minutes of processing time. Hence, in this case we must have $Z(t_2) = (0, 0, 0, n, 0, a)$, with $n \geq 1$ and $0 < a \leq 0.1$, which is a regular type 1 state. If Station A was serving a class 1 job just before t_2, then we have $Z(t_2) = (0, 1, 0, n, 0; a)$. In order for $Z(t_2)$ to be a regular state, we must have $0 < a \leq 0.6$ at t_2. Thus in Case 1, the network will enter either a regular type 1 or type 2 state at t_2.

Case 2. A class 4 job is in service at t_1 and $Z_1(t_1) = Z_3(t_1) = 0$. First suppose that the class 4 job has just entered service. Then we must have $Z(t_1) = (0, 0, 0, n, 0; a)$ with $n \geq 1$ and $0 < a \leq 0.1$, and we are in a regular type 1 state. If the class 4 job has a partial remaining service time at t_1, then at time t_0, when this job entered service, we have either $Z(t_0) = (0, 0, 0, n, 0; a)$ or $Z(t_0) = (0, 0, 0, n, 1; a)$ again with $n \geq 1$ and $0 < a \leq 0.1$ in both cases. The former state is a regular type 1 state. In the latter case, the buffer 5 job must have less than 0.1 minutes partial remaining service time and one can check that the evolution of the network from such a state is the same as starting from a "pure" regular type 1 state.

Case 3. A class 4 job is in service at t_1 and $Z_1(t_1) = 1$ or $Z_3(t_1) > 0$ (or both). If $Z_1(t_1) = 1$, then at $t_2 < t_1 + 0.1$ the class 4 job completes service and we have $Z(t_2) = (1, 0, m, n - 1, 1; a)$ with $a > 0.9$. At this time the class 1 and class 5 jobs will both initiate service. 0.4 minutes later, the state is $Z(t_2 + 0.4) = (0, 1, m, n - 1, 0; a)$. If $m = 0$ the network is in a (regular) type 2 state. If $m > 0$ then 0.1 minutes later the class 2 job enters buffer 3, leaving Station B empty. In this case, the network is in a state of the form of Case 1, with a buffer 3 job in service. On the other hand, if $Z_1(t_1) = 0$, then at $t_2 < t_1 + 0.1$ we have either $Z(t_2) = (1, 0, m, n - 1, 1; a)$ with $a > 0.9$ or $Z(t_2) = (0, 0, m, n - 1, 1; a)$ with $m > 0$. The former case has already been argued above. In the latter case, at $t_2 + 0.4$ we will either be in a (regular) type 1 state (if there are no arrivals and $m = 1$) or we will be back in Case 1.

Note that the only case we have not covered is $Z(t_1) = (0, 0, m, n, 0; a)$ with $m = n = 0$. This is because the zero state is not a regular state. To see this, we note that $|Z(t)| \geq 1$ for all $t \geq 1$, since arrivals occur exactly once a minute, and the time to complete processing at all buffers is 1.4 minutes.

□

Proof of Theorem 2.1. The proof of Theorem 2.1 now follows from the lemmas given in this section. Taking these lemmas together, we have that the network will enter an a-orbit from any initial state. □

5 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 5.1. *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,*

$$\mathbb{P}\left\{Z_4(T_4) \geq \frac{(1-m_1)m_5}{1-m_1-m_3}\theta\, n\right\} \geq 1-\exp(-\epsilon\sqrt{n}), \tag{12}$$

where

$$\begin{aligned} T_2 &= \inf\{t>0 : Z_3(t)=Z_4(t)=Z_5(t)=0\}, &(13)\\ T_4 &= \inf\{t>T_2 : \textit{ a class 4 job enters service at time } t \\ &\qquad\qquad \textit{and a class 2 job is not in service at } t+\}, &(14)\end{aligned}$$

with $\mathbb{P}\{T_4<\infty\}=1$. *Furthermore, for all sufficiently large n,*

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

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 5.1 and the unstable fluid solution constructed in Section 3 is evident. The magnification factor for the exponential network in (12) is smaller than the one for the fluid model in (11) 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 (12) 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 5.1 is actually general and holds for any service time vector for which $\rho_A<1$, $\rho_B<1$ and $\rho_{push}>1$.

In Section 6, we use Theorem 5.1 to complete the proof of Theorem 2.2. The remainder of this section is devoted to the proof of Theorem 5.1. The actual proof of Theorem 5.1 will be presented in Section 5.5, with the various lemmas presented in Sections 5.1–5.4. In Section 5.1, we show that during the bottom cycle the exponential network closely follows the unstable fluid solution in $[0,t_2]$. In Section 5.3, we show that during the top cycle the exponential network closely follows the unstable fluid solution in $[t_2,t_4]$. Sections 5.2 and 5.4 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.

5.1 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 θ_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 5.2. *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 (13)), with $Z(T_2) = (Z_1(T_2), Z_2(T_2), 0, 0, 0)$ such that for all n sufficiently large,*

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

We now introduce a number of definitions needed for the proof of Theorem 5.2.

5.1.1 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 σ_i and τ_i recursively as follows:

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

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 τ_i, we know that the entire bottom cycle ends at that time. Let

$$r = \inf\{i \geq 0 : Z_4(\tau_i) = 0\}. \tag{15}$$

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

(b) Throughout the interval $[\sigma_i, \tau_i)$, 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 $[\sigma_i, \tau_i)$.

We call the interval $[\sigma_i, \tau_i)$ the ith buffer 5 busy period or simply the ith busy period, and $[\tau_i, \sigma_{i+1})$ the ith impure period, for $i = 1, 2, \ldots$. When $i < r$, the ith 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 5.4. *Suppose that at time t a class 4 job enters service and no class 2 job is in service at t. Let t' be the time at which the current class 4 job completes service. Further, let $t'' \geq t'$ be the first time that buffer 1 is empty, after the class 4 service completion. Let $u_1 = t'' - t$. Then*

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

Furthermore, u_1 is independent of events occurring before t.

Proof. Once the class 4 job has completed service, Station A must serve any class 1 jobs which arrived during this class 4 service, until buffer 1 is empty. Once all class 1 jobs are cleared, Station A is available to process lower priority classes. Hence

$$\mathbb{E}[u_1] = \mathbb{E}[v_1 + s_1],$$

where v_1 is the time it takes to complete the class 1 jobs after the service at buffer 4 and s_1 is the service time at buffer 4. Thus, we have

$$\mathbb{E}[u_1] = \mathbb{E}[v_1] + m_4. \tag{16}$$

We now proceed to derive an expression for $\mathbb{E}[v_1]$. Let N be the number of jobs in buffer 1 after the service completion at buffer 4. Conditioning on s_1, we have:

$$\begin{aligned} \mathbb{E}[N] &= \mathbb{E}(\ \mathbb{E}[N \mid s_1]\ ) \\ &= \mathbb{E}(\alpha_1\, s_1) \\ &= \mathbb{E}(s_1) = m_4. \end{aligned}$$

Using the same procedure after conditioning on N, we have:

$$\begin{aligned} \mathbb{E}[v_1] &= \mathbb{E}(\ \mathbb{E}\,[v_1 \mid N]\ ) \\ &= \mathbb{E}\left[\frac{m_1}{1 - m_1}\, N\right] \\ &= \frac{m_1}{1 - m_1} \mathbb{E}[N] \\ &= \frac{m_1}{1 - m_1} m_4. \end{aligned}$$

The second line is obtained by applying the formula for the mean absorption time to zero from state N, for a birth-death process with constant birth rate 1 and constant death rate $1/m_1$ (see e.g. Karlin and Taylor [21], p. 149).

Plugging the above expression into (16) and doing some algebra yields:

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

One can check that when the usual traffic conditions are satisfied and $\rho_{push} > 1$, that $m_4/(1-m_1) < m_5$.

Finally, note that each time a class 4 job enters service, there must be zero jobs in both buffers 1 and 3. This fact, along with memoryless property of the interarrival times, implies that v_1 is independent of events before t. Using the independence assumptions on the service times for jobs, we have that u_1 is independent of events before t. □

Lemma 5.4 implies 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 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 5.5. *There exists a constant* $0 < c < 1$ *such that for each* $i \geq 1$*,*

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

Proof. We note that

$$\begin{aligned} \mathbb{P}\{r > i\} &= \mathbb{P}\{r > i-1, \tau_i < \infty, Z_4(\tau_i) > 0\} \\ &= \mathbb{P}\{r > i-1\}\mathbb{P}\{\tau_i < \infty, Z_4(\tau_i) > 0 | r > i-1\}, \end{aligned} \tag{17}$$

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

Consider a random walk on $\{0, 1, 2, 3, \cdots\}$ with positive drift. By Lemma 5.4, the number of jobs in buffer 5 during the busy period $[\sigma_i, \tau_i)$ is such a random walk, 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 random walk starts from a state that is bigger than or equal to 1.

Since

$$\{\tau_i < \infty, Z_4(\tau_i) > 0\} \subset \{\text{ the random walk ever reaches state } 0\},$$

and the probability c for the random walk with positive drift to *ever* reach state 0 is strictly less than one, it follows from (17) that $\mathbb{P}\{r > i\} \leq \mathbb{P}\{r > i-1\}\, c$ for each i. From this, and induction, the lemma follows. □

Corollary 5.1. *(a)* $\mathbb{P}\{r < \infty\} = 1$*, and (b)* $\mathbb{P}\{T_2 < \infty\} = 1$*.*

Proof. Part (a) follows from Lemma 5.5. From Part (a), we have $\mathbb{P}\{\sigma_r < \infty\} = 1$. It follows that $\mathbb{P}\{\tau_r < \infty\} = 1$, which implies (b). □

5.1.2 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 5.6.

Theorem 5.6. *For any $0 < \theta_2 < 1$, there exists an $\epsilon_2 > 0$, such that for all sufficiently large n,*

$$\mathbb{P}\{Z_4(\sigma_r) + Z_5(\sigma_r) \geq \theta_2 n\} \geq 1 - \exp(-\epsilon_2 \sqrt{n}).$$

Theorem 5.6 says that by time σ_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 (σ_i, σ_{i+1}), there cannot be too many leaks, when $i < r$.

So, let us fix a period (σ_i, σ_{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.3, 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 5.7. *Let q_i be the number of class 4 jobs that enter service within the ith impure period. There exists a constant c with $0 < c < 1$ such that for $i = 1, \ldots,$*

$$\mathbb{P}\{q_i > 2j)\} \leq c^j \quad \text{for } j = 0, 1, \ldots. \tag{18}$$

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 σ_{i+1}, contradicting the definition of σ_{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 ξ_k be the time that the kth job enters 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 kth job. If A_k occurs, the impure period ends with $k+1$ class 4 jobs having initiated services. Thus,

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

where A_k^c is the complement of A_k.

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 $c = \mathbb{P}\{A_k^c|\xi_k < \sigma_{i+1}\} < 1$. Note that this probability c depends only on the network parameters, i.e., the mean interarrival and service times. Thus, we have

$$\begin{aligned}
\mathbb{P}\{q_i > 2j\} &= \mathbb{P}\{\xi_{2j+1} < \sigma_{i+1}\} \\
&\leq \mathbb{P}\{\xi_{2j-1} < \sigma_{i+1}\} \cdot \mathbb{P}\{A_{2j-1}^c|\xi_{2j-1} < \sigma_{i+1}\} \\
&= \mathbb{P}\{\xi_{2j-1} < \sigma_{i+1}\}c \\
&\dots \\
&\leq \mathbb{P}\{\xi_1 < \sigma_{i+1}\}c^j \\
&\leq c^j,
\end{aligned}$$

proving (18). ☐

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

Lemma 5.8. *There exists an $\epsilon_3 > 0$, such that for all n large enough, for each $i = 1, 2, \dots,$*

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

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. Lemma 5.4 implies that 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 σ_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.3 at the time when a class 5 job is first in service during the busy period, the result follows. ☐

Proof of Theorem 5.6. 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

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

In the second to last line of the proof, the first term follows directly from Lemma 5.5, with an appropriate ϵ_4. The second term in the same line follows from Lemmas 5.7 and 5.8, again with an appropriate ϵ_5. The final inequality is valid for some $\epsilon_2 > 0$ if n is sufficiently large.

☐

5.2 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 5.9. *Let the Markov times σ_r and T_2 be defined as in (15) and (13), respectively. Then for any $0 < \theta_3 < 1$, there exists an $\epsilon_6 > 0$ such that for n large enough*

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

Proof. By the definition of T_2, all jobs present in buffers 4 and 5 at time σ_r will have departed the network by time T_2. By Theorem 5.6, we have that for any $0 < \theta_2 < 1$, buffer 5 must process $\theta_2 n$ jobs, except on an exponentially small set. So, $T_2 - \sigma_r$ is the sum of at least $\theta_2 n$ i.i.d. exponential random variables with mean m_5. Applying Lemma A.2 and using Theorem 5.6 we have, for all $\alpha > 0$ there exists an $\epsilon_7 > 0$ such that for all n large enough,

$$\begin{aligned}
\mathbb{P}\{T_2 - \sigma_r < \theta_2 m_5 n - \alpha n\} &\leq \exp(-\epsilon_7 n) + \exp(-\epsilon_2 \sqrt{n}) \\
\mathbb{P}\{T_2 - \sigma_r < \theta_3 m_5 n\} &\leq \exp(-\epsilon_7 n) + \exp(-\epsilon_2 \sqrt{n}) \\
\mathbb{P}\{T_2 - \sigma_r < \theta_3 m_5 n\} &\leq \exp(-\epsilon_6 \sqrt{n}),
\end{aligned}$$

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

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 5.2.

Proof of Theorem 5.2. Let $E_1(\cdot)$ be the counting process for exogenous arrivals to buffer 1 and let Y_n be the time of the nth arrival during $[\sigma_r, T_2]$. We choose any $\alpha > 0$. By applying Lemma 5.9 we have:

$$\begin{aligned}
\mathbb{P}\{E_1[\sigma_r, T_2] < \theta_3 m_5 n - \alpha n\} \leq\ & \mathbb{P}\{E_1[\sigma_r, T_2] < \theta_3 m_5 n - \alpha n \mid T_2 - \sigma_r \geq \theta_3 m_5 n\} \\
& + \exp(-\epsilon_6 \sqrt{n}).
\end{aligned} \tag{19}$$

Next, we do some rearranging and apply Lemma A.2 in the last inequality:

$$\begin{aligned}
\mathbb{P}\{E_1[\sigma_r, T_2] < \theta_3 m_5 n - \alpha n \mid T_2 - \sigma_r > \theta_3 m_5 n\} &= \mathbb{P}\{Y_{\lfloor \theta_3 m_5 n - \alpha n \rfloor} > T_2 - \sigma_r \mid T_2 - \sigma_r \geq \theta_3 m_5 n\} \\
&\leq \mathbb{P}\{Y_{\lfloor \theta_3 m_5 n - \alpha n \rfloor} > \theta_3 m_5 n\} \\
&\leq \mathbb{P}\{Y_{\lfloor \theta_3 m_5 n - \alpha n \rfloor} > \alpha n + \theta_3 m_5 n - \alpha n\} \\
&\leq \mathbb{P}\{Y_{\lfloor \theta_3 m_5 n - \alpha n \rfloor} > \alpha n + \lfloor \theta_3 m_5 n - \alpha n \rfloor\} \\
&\leq \exp(-\epsilon_8 n).
\end{aligned}$$

Now, plugging the above into (19) and setting $\theta_1 = \theta_3 - \alpha/m_5$, we have

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

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

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

Combining this with our previous inequality yields the theorem. ☐

5.3 The Top Cycle

T_2 is the beginning of what we shall call the top cycle. By virtue of Theorem 5.2, 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 5.1. 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 to 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 $\hat{\tau}_i$:

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

We call the interval $[\hat{\sigma}_i, \hat{\tau}_i)$ the ith buffer 3 busy period or simply the ith busy period, and $[\hat{\tau}_i, \hat{\sigma}_{i+1})$ the ith 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

$$r = \inf\{i : Z_2(\hat{\tau}_i) = 0\}. \tag{20}$$

Note that r is the smallest i such that $\hat{\tau}_i \geq T_3$. Analogous to Lemma 5.5, we have

Lemma 5.10. *There exists a constant c with $0 < c < 1$ such that*

$$\mathbb{P}\{r \geq j\} \leq c^j \quad \textit{for } j = 0, 1, \ldots.$$

The proof of this lemma is actually simpler than that of Lemma 5.5 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 5.4 for this proof. As before, we have as a corollary,

Corollary 5.2. *(a)* $\mathbb{P}\{r < \infty\} = 1$, *and (b)* $\mathbb{P}\{T_3 < \infty\} = 1$.

For $i < r$, the ith busy period is said to be incomplete, and for $i = r$, the ith 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 $[T_2, \hat{\sigma}_r]$.

Again, it is possible to have no leaks, in particular if buffer 3 is always kept busy from the first moment after T_2 at which it begins processing jobs, up until T_3. Intuitively, leaks are jobs which do not contribute to a large build-up of jobs at buffer 4 during the last buffer 3 busy period.

Here is the main theorem for the top cycle:

Theorem 5.11. *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 5.11 depends crucially on the following lemmas, which give bounds on the number of leaks before the last busy period.

Lemma 5.12. *Let q_i be the number of class 2 jobs that have started their services during a single impure period $[\hat{\tau}_i, \hat{\sigma}_{i+1})$. 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 5.7. Since the current lemma has a stronger result, we repeat some of the details here.

Fix an impure period $[\hat{\tau}_i, \hat{\sigma}_{i+1})$. 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 ξ_k be the time that the kth job enters class 2 service within the impure period, and let A_k be the intersection of the corresponding sequence of events (1–6 above) initiated by the kth 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 $\mathbb{P}\{A_k^c | \xi_k < \hat{\sigma}_{i+1}\} = c < 1$. Thus,

$$\begin{aligned} \mathbb{P}\{q_i > j\} &\leq \mathbb{P}\{\xi_j < \hat{\sigma}_{i+1}\} c \\ &\ldots \\ &\leq \mathbb{P}\{\xi_1 < \hat{\sigma}_{i+1}\} c^j \\ &\leq c^j, \end{aligned}$$

where the chain of inequalities is similar to that of the proof of Lemma 5.7. □

Lemma 5.13. *There exists an $\epsilon_{10} > 0$ such that for all n sufficiently large,*

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

Proof. The proof of the lemma is analogous to the proof of Lemma 5.8. However, in the top cycle case, in applying Lemma A.3, 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 5.4. □

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

Lemma 5.14. *Let $0 < \delta < 1$, then there exists an $\epsilon_{11} > 0$ such that for sufficiently large n,*

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

Proof. The proof follows from Lemmas 5.10, 5.12 and 5.13. □

Proof of Theorem 5.11.

$$\begin{aligned}
\mathbb{P}\{Z_1(\hat{\sigma}_r) + Z_2(\hat{\sigma}_r) < \theta_4 m_5 n\} &\leq \mathbb{P}\{Z_1(\hat{\sigma}_r) + Z_2(\hat{\sigma}_r) < \theta_4 m_5 n \mid Z_1(T_2) + Z_2(T_2) \geq \theta_1 m_5 n\} \\
&\quad + \exp(-\epsilon_1\sqrt{n}) \\
&\leq \exp(-\epsilon_{10}\sqrt{n}) + \exp(-\epsilon_1\sqrt{n}) \\
&\leq \exp(-\epsilon_9\sqrt{n}).
\end{aligned}$$

The first inequality follows by conditioning and applying Theorem 5.2. The second follows from Lemma 5.14. The third holds for appropriate ϵ_9 and sufficiently large n. □

5.4 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 5.2. 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 5.3. 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

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

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

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, $R_i \to \infty$ almost surely as $i \to \infty$. Since T_3 is almost surely finite by Corollary 5.2, there are a finite number of R_i before T_3. □

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 5.16. *Suppose that at time t a class 3 job enters service. Let t' be the time at which the current class 3 job completes service. Further, let $t'' \geq t'$ be the first time that buffer 1 is empty, after the class 3 service completion. Let $u_1 = t'' - t$. Then*

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

Furthermore, u_1 is independent of events occurring before t.

Proof. The proof of the lemma is exactly analogous to that of Lemma 5.4. If $\rho_{push} > 1$ and the usual traffic conditions hold, then one can check that $m_3/(1 - m_1) > m_2$. □

Lemma 5.17. *Let θ_5 be any fixed constant with $0 < \theta_5 < 1$. There exists an $\epsilon_{11} > 0$ such that for all s_i sufficiently large*

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

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.2 and 5.16, as in the proof of Lemma 5.9. □

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

Lemma 5.18. *Let θ_6 be any fixed constant with $0 < \theta_6 < 1$. There exists an $\epsilon_{12} > 0$ such that for all s_i sufficiently large*

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

Proof. The proof here is analogous to the proof of Theorem 5.2. □

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 5.19. *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 5.11, we have that, off an exponentially small set, there are at least $s_0 = \theta_4 m_5 n$ jobs in buffers 1 and 2 at time $R_0 = \hat{\sigma}_r$. An application of Lemmas 5.17 and 5.18 yields that for n large enough, off an exponentially small set there will be at least

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

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

$$s_1 = \left(\theta_6 \frac{m_3}{1-m_1}\right) \theta_4 m_5 n$$

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 θ_7 with $0 < \theta_7 < 1$, one can choose θ_4, θ_6 and N with $0 < \theta_i < 1$ such that

$$K \geq \theta_7 \frac{(1-m_1)m_5}{1-m_1-m_3}. \tag{21}$$

Next, for n large enough so that s_i, $i = 1, \ldots, N$, is large enough to apply Lemmas 5.17 and 5.18, 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 $[\hat{\sigma}_r, T_3)$. 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 (21) valid depends only on the problem data m_i, which is fixed, and the θ_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, …, s_N sufficiently large at times $R_1, \ldots, R_N$ to apply Lemmas 5.17 and 5.18. 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 5.17 and 5.18. Thus, we have

$$\begin{aligned}
&\mathbb{P}\{Z_1(R_{i+1}) + Z_2(R_{i+1}) + Z_3(R_{i+1}) \leq s_{i+1}\} \\
&\quad\leq \mathbb{P}\{Z_1(R_i) + Z_2(R_i) + Z_3(R_i) \leq s_i\} \\
&\qquad + \sum_{s=s_i}^{\infty} \mathbb{P}\{Z_1(R_{i+1}) + Z_2(R_{i+1}) + Z_3(R_{i+1}) \leq s_{i+1} | Z_1(R_i) + Z_2(R_i) + Z_3(R_i) = s\} \\
&\qquad \times \mathbb{P}\{Z_1(R_i) + Z_2(R_i) + Z_3(R_i) = s\} \\
&\quad\leq \mathbb{P}\{Z_1(R_i) + Z_2(R_i) + Z_3(R_i) \leq s_i\} + \exp(-\epsilon_{12} s_i) \\
&\qquad \ldots \\
&\quad\leq \mathbb{P}\{Z_1(R_0) + Z_2(R_0) + Z_3(R_0) \leq s_0\} + \sum_{k=0}^{i} \exp(-\epsilon_{12} s_k) \\
&\quad\leq \mathbb{P}\{Z_1(R_0) + Z_2(R_0) + Z_3(R_0) \leq s_0\} + (i+1)\exp(-\epsilon_{12} s_i) \\
&\quad\leq \exp(-\epsilon_9 \sqrt{n}) + N \exp(-\epsilon_{12} s_N) \quad \text{for } i = 0, \ldots, N-1,
\end{aligned}$$

where, in obtaining the last inequality, we have used Theorem 5.11. Now,

$$
\begin{aligned}
&\mathbb{P}\left\{Z_4(T_3) \le \theta_7 n \frac{m_5(1-m_1)}{1-m_1-m_3}\right\} \\
&\le \mathbb{P}\left\{\sum_{k=0}^{N}(Z_1(R_k)+Z_2(R_k)+Z_3(R_k)) \le \theta_7 n \frac{m_5(1-m_1)}{1-m_1-m_3}\right\} \\
&\le \mathbb{P}\left\{\sum_{k=0}^{N}(Z_1(R_k)+Z_2(R_k)+Z_3(R_k)) \le \sum_{k=0}^{N} s_k\right\} \\
&\le \sum_{k=0}^{N}\mathbb{P}\left\{Z_1(R_k)+Z_2(R_k)+Z_3(R_k) \le s_k\right\} \\
&\le (N+1)\left(\exp(-\epsilon_9\sqrt{n}) + N\exp(-\epsilon_{12}s_N)\right) \\
&\le \exp(-\epsilon_{13}\sqrt{n}).
\end{aligned}
$$

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

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

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. ☐

5.5 Proof of Theorem 5.1

Theorem 5.19 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 2.2, we must obtain three additional results.

First, we note that the beginning and ending states in Theorem 5.19 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 5.19. Specifically, we wish to show that at time $T_4 \ge T_3$, 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). We show this, along with the finiteness of T_4, in Theorem 5.1.

Next, we use the results of Section 5 and Theorem 5.1 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 5.20.

Proof of Theorem 5.1. We first prove that T_4 is finite with probability one. Recall that T_2 is finite with probability one, as shown in Corollary 5.1. Let $V_1, V_2, \ldots$ be the sequence of times after T_2 at which a class 4 job enters service. Since both stations empty infinitely often with probability one, each V_i is finite with probability one. Note that if there is no class 2 job in service at V_i, then we set $T_4 := V_i$ by definition. On the other hand, suppose that there is a class 2 job in service at V_i. Then the state of the network must be $Z(V_i) = (0, n, 0, m, r)$, where $n > 0$, $m > 0$ and $r \ge 0$, with a class 2 job in service.

Next, we note that every time the network enters such a state, there is a strictly positive probability that the following sequence of events occurs:

1. The class 4 job completes service and moves to buffer 5. Another class 4 job (if $m > 0$) enters service.

2. The class 2 job completes service and moves to buffer 3. The buffer 5 job enters service.

3. The next class 4 job completes service (again, in the case $m > 0$).

4. The class 3 job completes service and moves to buffer 4.

5. A class 4 job enters service (either the job that just arrived from buffer 3, or another class 4 job).

This sequence of events assumes that steps 3 and 4 occur before the class 5 job completes service and that there are no external arrivals to the system during the entire sequence. Every time a class 4 job enters service, buffers 1 and 3 must be empty. Using this, and the Markov property, we conclude that at each V_i there is a positive probability, which can be bounded away from zero independent of past events, that $V_{i+1} = T_4$. Hence, $T_4 - T_2$ can be bounded above by a proper geometric sum of proper random variables, which implies that $\mathbb{P}\{T_4 < \infty\} = 1$.

Next, we prove the main part of the theorem, which the probabilistic bound on the number of jobs in buffer 4 at T_4. 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 5.1.

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 by $q + 1$, where q is the number of class 4 jobs that have started service in the interval. By Lemma 5.7, 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 5.19 as follows:

$$\begin{aligned} \mathbb{P}\{Z_4(T_4) \leq c\,\theta_7 n - c\,\theta_7\sqrt{n}\} &\leq \mathbb{P}\{Z_4(T_4) \leq c\,\theta_7 n - c\,\theta_7\sqrt{n} \mid Z_4(T_3) \geq c\,\theta_7 n\} \\ &\quad + \mathbb{P}\{Z_4(T_3) < c\,\theta_7 n\} \\ &\leq \exp(-\epsilon_{14}\sqrt{n}) + \exp(-\epsilon_{13}\sqrt{n}) \\ &\leq \exp(-\epsilon_{15}\sqrt{n}), \end{aligned}$$

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 θ_7 and $(1-\sqrt{n}/n)$ can be made arbitrarily close to 1, for n sufficiently large.

□

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 θ_i can be made arbitrarily close to one. In the following arguments, we assume that the θ_i are sufficiently close to one to suit our needs.

Theorem 5.20. *There exists an $\epsilon_{16} > 0$ such that, for sufficiently large n,*

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

Proof. On $[0, \sigma_r]$ the lower bound on $|Z(t)|$ follows directly from the proof of Theorem 5.6, as long as θ_2 is close to unity. Next, let $\tilde{\sigma}_r > \sigma_r$ 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 $\tilde{\sigma}_r$ and Theorem 5.6, the lower bound in fact holds on $[0, \tilde{\sigma}_r]$. Now, we need only show that, off a small set, there will be at least $n/4$ arrivals to the network during $[\sigma_r, \tilde{\sigma}_r]$. If this is so, then the bound holds on $[0, T_2]$.

In $[\sigma_r, \tilde{\sigma}_r]$ buffer 5 must process $(\theta_2 - 1/4)n$ jobs, off an exponentially small set. Now, by arguments analogous to the proof of Lemma 5.9, the amount of time needed to process $(\theta_2 - 1/4)n$ jobs at buffer 5 is $(\theta_9 - 1/4)m_5 n$, off a small set, where $0 < \theta_9 < 1$. By arguments analogous to the proof of Theorem 5.2, the number of exogenous arrivals during $[\sigma_r, \tilde{\sigma}_r]$ is $(\theta_{10} - 1/4)m_5 n$ 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 θ_{10} sufficiently close to one. Now, since all jobs that arrive during $[\sigma_r, \tilde{\sigma}_r]$ 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,

$$\mathbb{P}\{Z_1(\tilde{\sigma}_r) + Z_2(\tilde{\sigma}_r) \geq (\theta_{10} - 1/4)m_5 n\} \geq 1 - \exp(-\epsilon_{17}\sqrt{n}).$$

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 5.3. In particular Theorem 5.11 guarantees that the total number of jobs in the network during the top cycle is at least $\theta_4 m_5 n$ (off a small set), which again is easily bigger than $n/4$ for θ_4 close to one.

Finally, Theorem 5.1 insures that the network does not lose too many jobs during $[T_3, T_4]$, off a small set. In particular Theorem 5.1 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 5.1 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. ☐

6 The Exponential Network – Proof of Theorem 2.2

Finally, we need to use all of our previous results to complete the proof of Theorem 2.2. Specifically, we need to show that Theorem 5.1 implies our main result.

Proof of Theorem 2.2. We are now ready to prove Theorem 2.2. 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 2.2 by contradiction. So, suppose that there exists an initial state z_0 such that

$$\mathbb{P}_{z_0}(\{\omega : Z(t,\omega) \not\to \infty\}) > 0, \tag{22}$$

where $\mathbb{P}_z(\cdot)$ is the probability measure induced when starting in state z. For an integer ℓ, let $A_\ell = \{\text{state } z : |z| \le \ell\}$. Then (22) implies that there exists an $\ell > 0$ such that

$$\mathbb{P}_{z_0}(\cap_{k=1}^{\infty} \cup_{t\in[k,\infty)} \{\omega : Z(t,\omega) \in A_\ell\}) \equiv \delta > 0. \tag{23}$$

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 5.1. Fix $\theta < 1$ such that $c\,\theta > 1$, where

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

By repeatedly applying Theorem 5.1 and the strong Markov property, we have for large enough n,

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

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 5.1 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 5.1 is accessible from the zero state. Since any state can access the zero state, we have

$$\mathbb{P}_z\{\text{ the Markov process } Z \text{ eventually reaches state } z_1\} > 0$$

for any initial state z. Because the set A_ℓ is finite, we have

$$\min_{z\in A_\ell} \mathbb{P}_z\{\text{ the Markov process } Z \text{ eventually reaches state } z_1\} > 0.$$

Now for any ω 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 $k \ge 1$. Each time the process enters A_ℓ, it has a positive probability hitting state z_1. Thus, on the event $\cap_{k=1}^{\infty} \cup_{t\in[k,\infty)} \{\omega : Z(t,\omega) \in A_\ell\}$, the process Z will hit z_1 with probability 1. Therefore, we have

$$\begin{aligned}
\delta &= \mathbb{P}_{z_0}(\cap_{k=1}^{\infty} \cup_{t\in[k,\infty)} \{\omega : Z(t,\omega) \in A_\ell\}) \\
&= \mathbb{P}_{z_0}\left(\cap_{k=1}^{\infty} \cup_{t\in[k,\infty)} \{\omega : Z(t,\omega) \in A_\ell\} \cap \{Z(T) = z_1\}\right) \\
&\le \mathbb{P}_{z_1}(\cap_{k=1}^{\infty} \cup_{t\in[k,\infty)} \{\omega : Z(t,\omega) \in A_\ell\}) \\
&\le \mathbb{P}_{z_1}(\{\omega : Z(t,\omega) < n/4 \quad \text{for some} \quad t \ge 0\}) \\
&\le \delta/2,
\end{aligned}$$

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 (24).

□

7 The Uniform Network

In Sections 5 and 6, we proved that the non-preemptive exponential network is unstable, in the sense that the number of jobs in the system goes to infinity with probability one. In Section 4 we proved that the network operating under the same policy is stable, when interarrival and service times are deterministic. In particular, from any initial state, the deterministic network will reach (and stay in) a set of states with less than two jobs. The interarrival and service time distributions in both of these cases are in some sense extreme. The exponential distribution has unbounded support, and the deterministic "distribution" is degenerate. One may then wonder how the stability or instability of our network depends on the range or variability of the network primitives. In this section, we report the results of simulation studies undertaken to give some insight into this question.

We again investigated the 2-station network of Figure 1. As before, we fix the arrival rate and mean service times to be:

$$\alpha_1 = 1, \quad m_1 = 0.4, \quad m_2 = 0.1, \quad m_3 = 0.4, \quad m_4 = 0.1 \quad \text{and} \quad m_5 = 0.4. \tag{25}$$

Also, the network is once again assumed to be operating under the non-preemptive SBP service policy. However, in our simulations, the service times for class i jobs are set to be uniform random variables with a range of $(m_i - \epsilon/2, m_i - \epsilon/2)$ and the interarrival times are uniform random variables with a range of $(1-\epsilon/2, 1-\epsilon/2)$ where $\epsilon \geq 0$. We will refer to such a network as a uniform(ϵ) network. In particular, Section 4 demonstrates (analytically) that the uniform(0) network is stable. For other values of ϵ, we conducted various simulations to determine stability or instability of a uniform(ϵ) network. It should be noted that we do not provide any proofs or formal statistical results in this section. Our simulation studies simply give an indication of stability or instability. However, we believe that the simulation evidence presents a convincing case for the stability characterization in each case.

		Average Utilization		**Utilization Range**	
ϵ	**Stable**	**Station A**	**Station B**	**Station A**	**Station B**
0.1	No	0.87809	0.45226	(0.86798, 0.88774)	(0.43085, 0.47405)
0.001	Yes	0.90005 $\pm 7.87 \times 10^{-5}$	0.50008 $\pm 1.02 \times 10^{-4}$	(0.89999, 0.90379)	(0.50000, 0.50371)

Table 1: Simulation Study Results

Simulation Studies. All simulations were performed in Arena 3.03. Table 1 provides a short summary of our simulation results. Each network was simulated for 110,000 minutes of simulation time, with 100 independent replications being performed. The statistics we report in the table and our analysis below are based on the utilization at each station. The simulation statistics were only collected for the last 100,000 minutes of simulation time, to reduce initialization bias. Note that using (8) one can calculate the "nominal utilization" levels for Stations A and B, which are $\rho_A = 0.9$ and $\rho_B = 0.5$, respectively. One can show that in any multiclass network, if the average utilization over $[0, t]$ on any sample path does not converge to the nominal utilization as t goes to infinity (at any station), then the number of jobs goes to infinity on this sample path. To be more precise, let $B_i(t)$ be the amount of time station i spent serving jobs in $[0, t]$. Then it can be shown that

$$\limsup_{t\to\infty} \frac{B_i(t,\omega)}{t} < \rho_i \text{ for some station } i \text{ implies } \quad |Z(t,\omega)| \to \infty \quad \text{as } t \to \infty,$$

for a sample path ω. On the other hand, if the average utilization converges to the nominal utilization at each station i for a given sample path, i.e., $\lim_{t\to\infty} B(t,\omega)/t = \rho_i$, then the network is rate stable on this path.

For the case $\epsilon = 0.1$ we initialized the network with varying numbers of jobs in buffer 4 and zero jobs in all other buffers. The smallest value of $Z_4(0)$ which makes all 100 replications unstable is 49. For these runs with $Z_4(0) = 49$, among the 100 replications, we recorded both the maximum average utilization and minimum average utilization (labeled as "Utilization Range" in the table) at Station A and Station B. We also recorded the average utilization at each station, across all 100 runs. From Table 1, we see that no replication in this set of runs achieved the nominal utilization at either station, and that the average utilizations are far below the nominal values. We also did 100 replications starting with $Z_4(0) = 48$, in this case, we found that the utilizations in three replications were close to the nominal values, indicating the network is stable along three sample paths. This seems to suggest that the number of jobs goes to infinity with positive probability, but that this probability is strictly less than one. Presumably, if the network is initialized with more than 49 jobs in buffer 4, most or all similar simulation runs would indicate instability of the uniform(0.1) network.

For the case of $\epsilon = 0.001$, we initialized the network with 100 jobs in every buffer. The network was intentionally initialized in a "worse" (more heavily loaded) state for these runs, as compared to the runs with $\epsilon = 0.1$. For these runs we again recorded the average utilization and in addition calculated 95% confidence intervals for the average utilization. For comparison, we also report the maximum and minimum average utilizations, over the 100 replications. Our simulation results are presented in Table 1. The last row of the table gives the half-lengths of the confidence intervals for average utilization from the uniform(0.001) network simulations. We see that the average utilizations are near the nominal values and that the nominal values are also within the confidence intervals, as expected for a stable network. We also simulated the network from a number of other initial states. All the results from these runs indicate that the uniform(0.001) network is stable from these initial states.

In Figure 2, we provide a WIP profile for one run of the uniform(0.001) network, which seems to be stable, according to our studies. Figure 3 provides a WIP profile for one run of the uniform(0.1) network, which our simulation studies indicate to be unstable. From examining other runs for both networks, these WIP profiles appear to be representative of the typical behavior of these networks.

Our simulation studies seem to indicate that the variability of interarrival and service times have an effect on network stability, i.e. if these times are "close" to deterministic, then the network is stable. Otherwise, if there is sufficient variability in these times, the network is pushed into the instability regime. This indicates that the deterministic and exponential networks studied in earlier sections are not simply pathological cases, but perhaps typical examples of the effect of variability on stability.

8 The Deterministic Network with Preemption

In this section, we again consider the 2-station deterministic network. As in Section 4, the network has deterministic (i.e. constant) interarrival and service times. However, in this section and the next, the network is assumed to be operating under the *preemptive* SBP service policy, as described in Subsection 2.1. We prove Theorem 2.3, which implies that such a network is not rate stable, from many initial states. It should be noted that the preemptive deterministic network is not unstable from *all* initial states. In particular, if the initial state is $(0,0,0,1,0;a)$ with $0.1 \le a < 0.2$, then the network will fall into an orbit like the a-orbit introduced in Subsection 2.2. There may also be

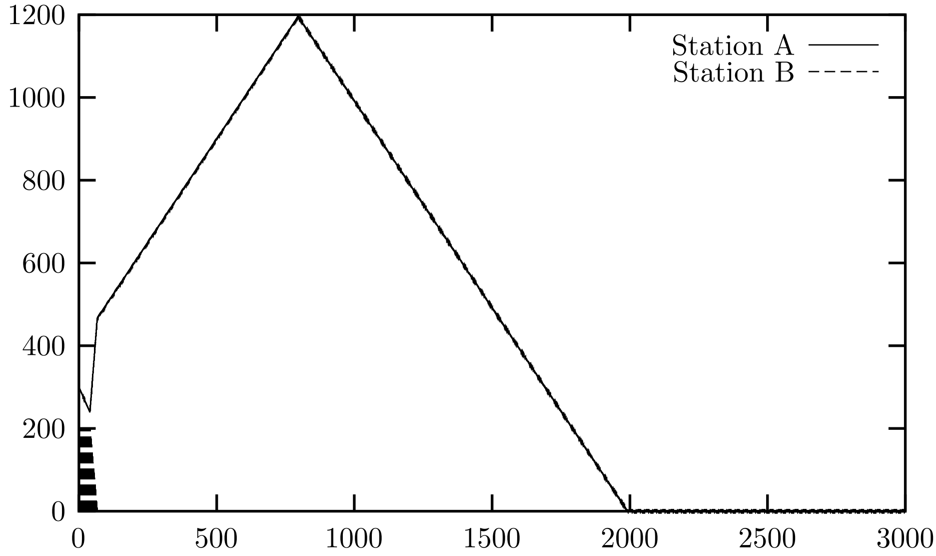

Figure 2: A Stable Sample Path

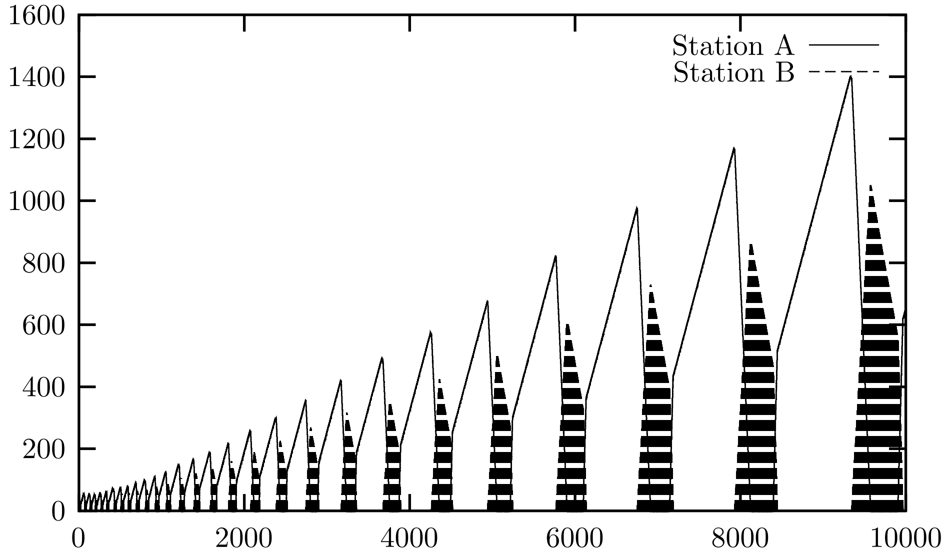

Figure 3: An Unstable Sample Path

many other initial states from which this network is stable.

Theorem 2.3 follows directly from Lemma 8.1 below. We first need the following definition.

Definition 8.1. Let $\mathcal{I}$ be the set of states of the following type:

- $(0, 0, 0, n, 0; a)$ with $0 < a < 1$ and a class 4 job having zero remaining service time,
- $(0, 1, 0, n, 0; a)$ with $0.5 < a < 0.6$ and a class 4 job having zero remaining service time,

where in both cases, we assume $n \geq 1$.

Lemma 8.1. *Consider the deterministic network operating under the preemptive SBP service policy. Let $Z(0) \in \mathcal{I}$ and $Z_4(0) = n$. Then for any constants $k_1 < 1.2$ and $k_2 > 1.2$, for n sufficiently large, there exists a time T such that:*

(a) $Z(T) \in \mathcal{I}$ and $Z_4(T) > k_1 \cdot n$ and

(b) $T \leq k_2 \cdot n$.

Proof. Part a. We assume that $Z(0) \in \mathcal{I}$ and that one class 4 job has zero remaining service time at time 0, i.e. at time 0+ a class 4 job will be passed to buffer 5. If the class 2 job also has zero remaining service time at time 0, we resolve the simultaneous events in the following manner. The class 4 job completes service "first," is passed to buffer 5, and then preempts the zero service time job in buffer 2.

Since class 4 jobs require only 0.1 minutes of service time, whereas class 5 jobs require 0.4 minutes of service time, we claim that buffer 5 will remain positive, independent of the initial state, until after buffer 4 has drained for the first time.

Next, we let $t_2 = \inf\{t > 0 : Z_5(t) = 0\}$. Note that under the SBP policy, no jobs will be processed at buffer 2 in the interval $[0, t_2)$. Furthermore, since $Z_1(0) = 0$ and class 1 jobs have preemptive priority, it is clear that $Z_1(t) \leq 1$ for all $t \geq 0$. Hence, at t_2 the network state will be of the form $Z(t_2) = (0/1, z_2, 0, 0, 0)$, where buffer 1 has either one job or no jobs. Now, since all jobs which arrived during $[0, t_2)$ must be in buffers 1 or 2, we have $z_2 \geq t_2 - 2$. We subtract 2 to account for round-off error in the number of arrivals in $[0, t_2)$ and the possibly one job in buffer 1. Note that in $[0, t_2]$, exactly n jobs were processed at buffer 5. Hence $t_2 = 0.4\,n$. Combining our last two observations, we have $z_2 \geq 0.4\,n - 2$.

Note that for n sufficiently large, we will have $Z_2(t_2) > 0$. Next, let $t_3 = \inf\{t > t_2 : Z_2(t) = 0\}$ and

$$t_4 = \inf\{t > t_2 : \text{a class 4 job departs buffer 4 at } t\}.$$

At t_2 a class 2 job will begin processing and be sent to buffer 3 at $t_2 + 0.1$. Since class 2 jobs are processed faster than class 3 jobs, it is clear that buffer 3 will become overwhelmed with jobs and cannot be empty until after t_3. Now, we let $t_4' = \inf\{t > t_2 + 0.1 : Z_3(t) = 0\}$. Under the SBP policy, we must have $t_4' \leq t_4$. We now wish to provide a lower bound on t_4'.

We have already argued that buffer 1 can contain at most one job at any time. Similarly, since $Z_5(t) = 0$ for $t \in (t_3, t_4)$ buffer 2 can contain at most one job. In fact, for t in this interval we have $Z_1(t) + Z_2(t) \leq 1$. Thus, when buffer 3 becomes empty for the first time at t_4' it must have processed all z_2 jobs which were in buffer 2 at t_2 and all but possibly one of the jobs which arrived during (t_2, t_4'). Furthermore, buffer 1 has processed all of the jobs which arrived in this interval. Combining all of these observations, we have:

$$t_4' - t_2 \geq 0.4\,z_2 + (0.4 + 0.4)[t_4' - t_2 - 2].$$

Rearranging yields $t_4' - t_2 \geq 2\,z_2 - 8$. Finally, the number of jobs present at buffer 4 at time t_4- must include all jobs which arrived in (t_2, t_4), except possibly one job which could be in buffer 2, plus the z_2 jobs present in buffer 2 at t_2. The number of jobs which arrived in (t_2, t_4) is bounded below by the number which arrived in (t_2, t_4') which in turn is bounded below by $t_4' - t_2 - 1$. We now combine these observations with previous inequalities:

$$\begin{aligned} Z_4(t_4) &\geq t_4' - t_2 - 2 + z_2 \\ &\geq 2\,z_2 - 10 + z_2 \\ &\geq 3\,z_2 - 10 \\ &\geq 3(0.4n - 2) - 10 \\ &= 1.2n - 16. \end{aligned}$$

Clearly, for n sufficiently large, we can insure $Z_4(t_4) > k_1 \cdot n$, for any $k_1 < 1.2$.

We now argue that $Z_4(t_4) \in \mathcal{I}$. Since a buffer 4 job is completed at t_4+, we must have $Z_1(t_4) = Z_3(t_4) = 0$, due to the preemptive policy. Furthermore, we know that $Z_5(t_4) = 0$, from our definitions above. If $Z_2(t_4) = 0$, then we automatically have $Z(t_4) \in \mathcal{I}$. If $Z_2(t_4) = 1$, then this job must have been passed to buffer 2 in the interval $(t_4 - 0.1, t_4)$. Since class 1 jobs have highest priority, the same job must have arrived at buffer 1 during $(t_4 - 0.5, t_4 - 0.4)$. Hence the next arrival to the network will occur in $(t_4 + 0.5, t_4 + 0.6)$. In this case we also have $Z(t_4) \in \mathcal{I}$.

Part b. First we recall from above that $t_2 = 0.4\,n$. Next, once buffer 3 becomes empty for the first time at t_4', it is straightforward, but tedious, to show that buffer 4 will complete processing of a job within 2 minutes. We also observe that buffer 1 or buffer 3 must constantly be processing jobs during $(t_2 + 0.1, t_4')$. The amount of processing time devoted to processing such jobs cannot be more than $0.4z_2$ (the time buffer 3 spends on the original buffer 2 jobs) plus $(0.8)(t_4' - t_2)$ (the amount of time buffers 1 and 3 require to process all the jobs arriving in the interval). We also observe that $z_2 \leq t_2 = 0.4\,n$. So, we have:

$$\begin{aligned} t_4' - t_2 - 0.1 &\leq 0.4z_2 + 0.8(t_4' - t_2) \\ &\leq 0.4 \cdot 0.4n + 0.8(t_4' - t_2) \end{aligned}$$

Rearranging, we have $t_4' - t_2 \leq 0.8\,n + 0.5$. Finally, $t_4 \leq t_2 + t_4' - t_2 - 0.1 + 2 \leq 1.2\,n + 2.4$. Hence, for n sufficiently large, we can satisfy part b of the lemma. □

Proof of Theorem 2.3. We note that the initial state in Theorem 2.3 is in the class $\mathcal{I}$ of Lemma 8.1. Hence, we can apply the lemma starting at time 0. Since the lemma indicates that we will always return to a state in the same class as the initial state, we conclude that there exist a sequence of times $s_1, s_2, \ldots$ with

$$Z_4(s_i) > (k_1)^i\,n \qquad i = 1, 2, \ldots.$$

Also, from the lemma we have that

$$\frac{Z_4(s_{i+1}) - Z_4(s_i)}{s_{i+1} - s_i} \geq \frac{k_1 - 1}{k_2}.$$

Since k_1 can be chosen greater than 1 and k_2 can be chosen to be positive, this implies that $Z(t)$ is going to infinity linearly along the sequence of times $\{s_i, i \geq 1\}$.

□

9 The Exponential Network with Preemption

In this section, we consider the 2-station exponential network operating under the preemptive SBP policy. As in Sections 5 and 6, all interarrival and service times are assumed to be i.i.d. exponential random variables. We will refer to this network as the preemptive exponential network. As before, we let $|Z(t)|$ denote the total number of jobs in the network at time t.

For the preemptive exponential network, it turns out that an analog of our main result for the non-preemptive exponential network (Theorem 2.2) holds:

Theorem 9.1. *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 9.1. The full proof of the theorem proceeds in an exactly analogous manner to the proof of Theorem 2.2, 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 9.1. As in the non-preemptive case, the majority of the proof of Theorem 9.1 involves proving a theorem similar to Theorem 5.1. The following theorem can be proven more directly than Theorem 5.1 due to the preemption employed.

Theorem 9.2. *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,*

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

where

$$\begin{aligned} T_2 &= \inf\{t > 0 : Z_3(t) = Z_4(t) = Z_5(t) = 0\}, \\ T_4 &= \inf\{t > T_2 : Z_1(t) = Z_3(t) = Z_5(t) = 0\} \end{aligned}$$

Furthermore, for all sufficiently large n,

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

The main result in the preemptive case, Theorem 9.1 follows directly from Theorem 9.2, as in the non-preemptive case.

Next, we briefly outline the arguments needed to establish Theorem 9.2. 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 5.6. 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 5.6 has been established, the analogous result to Theorem 5.2 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 5.11 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. □

10 Virtual Stations and Push Starts

In this section, we give a more detailed discussion of virtual station and push start conditions. We also attempt to give deeper insight into the results of this paper.

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

$$\rho_{push} = \frac{m_3}{1-m_1} + m_5 \leq 1 \tag{26}$$

is violated. The push start condition is a magnification of a virtual station condition identified in Dai and Vande Vate [13]. The virtual station effect can most easily be seen in the *queueing network* operating the *preemptive* SBP service policy. The following proposition is a special case of Proposition 3.1 of [19]. 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 10.1. *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,*

$$Z_3(t)Z_5(t) = 0 \quad \text{for all } t \geq 0 \tag{27}$$

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

Let us suppose that the network is initially empty. A consequence of equation (27) 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,

$$m_3 + m_5 \leq 1 \tag{28}$$

is necessary for the stability of the queueing network and the corresponding fluid model. Condition (28) 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 [24]. 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 (27) continues to hold in the Lu-Kumar queueing network. Thus, (28) 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 (28) in the Lu-Kumar fluid model leads to the necessity condition (26) in our 5-class fluid model.

As we have seen, the derivation of the push start condition (26) relies on the following factors: (a) the service policy is preemptive so that the virtual station phenomenon (27) 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 (26) 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. The results of Sections 5 and 6 show that, for the exponential network, even under the non-preemptive SBP policy, the push start condition is necessary. Section 8 demonstrates that the deterministic network operating under the non-preemptive SBP policy 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 5.16, 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 (27), to occur. Rather, in this case we have a "partial" virtual station effect. The proof of the following proposition is analogous to the proof of Proposition 3.1 of [19] and is thus omitted.

Proposition 10.2. *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,*

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

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 one job at any time. However, the "partial" virtual station effect in Proposition 10.2 is not sufficient to cause instability, even though the virtual station condition (28) 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 exponential and deterministic networks. 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. 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 2-station multiclass queueing networks. A plausible conjecture is that for 2-station reentrant lines with exponential interarrival and service times, the virtual station and push-start conditions of Dai and Vande Vate [13] 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 [20]. 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 simulation results presented in this paper, along with many more simulations conducted earlier in the research for insight, were conducted by Ki-Seok Choi, a Ph.D. student at the Georgia Institute of Technology. The authors would like to thank Ki-Seok for his invaluable efforts towards this project. The authors would also like to thank Maury Bramson and Ruth Williams for providing valuable input in the course of this research. A portion of this research was performed while Jim Dai was visiting TLI-AP and the Decision Science Department at the National University of Singapore. Also, some of the 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

[1] D. Bertsimas, D. Gamarnik, and J. N. Tsitsiklis. Stability conditions for multiclass fluid queueing networks. *IEEE Transactions on Automatic Control*, 41:1618–1631, 1996. Correction: **42**, 128, 1997.

[2] M. Bramson. Instability of FIFO queueing networks. *Annals of Applied Probability*, 4:414–431, 1994.

[3] M. Bramson. Convergence to equilibria for fluid models of FIFO queueing networks. *Queueing Systems: Theory and Applications*, 22:5–45, 1996.

[4] M. Bramson. Stability of two families of queueing networks and a discussion of fluid limits. *Queueing Systems: Theory and Applications*, 28:7–31, 1998.

[5] M. Bramson. A stable queueing network with unstable fluid model. *Annals of Applied Probability*, 9(3):818–853, 1999.

[6] H. Chen. Fluid approximations and stability of multiclass queueing networks I: Work-conserving disciplines. *Annals of Applied Probability*, 5:637–665, 1995.

[7] H. Chen and H. Zhang. Stability of multiclass queueing networks under FIFO service discipline. *Mathematics of Operations Research*, 22:691–725, 1997.

[8] J. G. Dai. On positive Harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. *Annals of Applied Probability*, 5:49–77, 1995.

[9] J. G. Dai. Stability of open multiclass queueing networks via fluid models. In F. Kelly and R. J. Williams, editors, *Stochastic Networks*, volume 71 of *The IMA volumes in mathematics and its applications*, pages 71–90, New York, 1995. Sringer-Verlag.

[10] J. G. Dai. A fluid-limit model criterion for instability of multiclass queueing networks. *Annals of Applied Probability*, 6:751–757, 1996.

[11] J. G. Dai and S. P. Meyn. Stability and convergence of moments for multiclass queueing networks via fluid limit models. *IEEE Transactions on Automatic Control*, 40:1889–1904, 1995.

[12] J. G. Dai and J. VandeVate. Global stability of two-station queueing networks. In K. S. Paul Glasserman and D. Yao, editors, *Proceedings of Workshop on Stochastic Networks: Stability and Rare Events*, pages 1–26, Columbia University, New York, 1996. Springer-Verlag.

[13] J. G. Dai and J. VandeVate. The stability of two-station multitype fluid networks. *Operations Research*, 48:721–744, 2000.

[14] V. Dumas. A multiclass network with non-linear, non-convex, non-monotonic stability conditions. *Queueing Systems: Theory and Applications*, 25:1–43, 1997.

[15] M. El-Taha and S. Stidham. *Sample-Path Analysis of Queueing Systems.* Kluwer Academic Publishers, New York, 1999.

[16] S. G. Foss and A. Kovalevskii. A stability criterion via fluid limits and its application to a polling system. *Queueing Systems: Theory and Applications*, 32(1–3):131–168, 1999.

[17] J. M. Harrison and V. Nguyen. The QNET method for two-moment analysis of open queueing networks. *Queueing Systems: Theory and Applications*, 6:1–32, 1990.

[18] J. M. Harrison and V. Nguyen. Some badly behaved closed queueing networks. In F. P. Kelly and R. J. Williams, editors, *Stochastic Networks*, volume 71 of *The IMA volumes in mathematics and its applications*, pages 117–124, New York, 1995. Springer-Verlag.

[19] J. J. Hasenbein. Necessary conditions for global stability of multiclass queueing networks. *OR Letters*, 21:87–94, 1997.

[20] J. J. Hasenbein. Stability of fluid networks with proportional routing. *Queueing Systems: Theory and Applications*, 38:319–349, 2001.

[21] S. Karlin and H. M. Taylor. *A first course in stochastic processes, Second edition.* Academic Press, New York, 1975.

[22] P. R. Kumar. Re-entrant lines. *Queueing Systems: Theory and Applications*, 13:87–110, 1993.

[23] S. Kumar and P. R. Kumar. Fluctuation smoothing policies are stable for stochastic reentrant lines. *Discrete Event Dynamical Systems*, 6:361–370, 1996.

[24] S. H. Lu and P. R. Kumar. Distributed scheduling based on due dates and buffer priorities. *IEEE Transactions on Automatic Control*, 36:1406–1416, 1991.

[25] S. P. Meyn. Transience of multiclass queueing networks via fluid limit models. *Annals of Applied Probability*, 5:946–957, 1995.

[26] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability.* Springer-Verlag, New York, 1993.

[27] A. Pulhaskii and A. Rybko. Non-ergodicity of queueing networks when their fluid model is unstable. *Problems of Information Transmission*, 36:26–46, 2000.

[28] A. N. Rybko and A. L. Stolyar. Ergodicity of stochastic processes describing the operation of open queueing networks. *Problems of Information Transmission*, 28:199–220, 1992.

[29] A. Shwartz and A. Weiss. *Large deviations for performance analysis.* Chapman & Hall, New York, 1995.

[30] A. Stolyar. On the stability of multiclass queueing networks: a relaxed sufficient condition via limiting fluid processes. *Markov Processes and Related Fields*, pages 491–512, 1995.

[31] A. L. Stolyar and K. K. Ramakrishnan. The stability of a flow merge point with non-interleaving cut-through scheduling disciplines. *Proceedings of INFOCOM'99*, 1999.

[32] R. J. Williams. Diffusion approximations for open multiclass queueing networks: sufficient conditions involving state space collapse. *Queueing Systems: Theory and Applications*, 30:27–88, 1998.

A Appendix

Proof of Lemma 4.4. The proof essentially follows by examining the sequence of states which occur from the starting states given in the lemma.

We begin by initializing the network in a state $(0, 1, 0, n, 0; a)$ with $n \geq 1$ and $0.1 < a \leq 0.2$. The table below lists the sequence of states from this initial state.

Time	State
0	(0, 1, 0, n, 0; a)
0.1	(0, 0, 1, n-1, 1; a-0.1)
a	(1, 0, 1, n-1, 1; 1)
0.2	(1, 0, 1, n-1, 1; a+0.8)
0.5	(1, 0, 0, n, 0; a+0.5)
0.9	(0, 1, 0, n, 0 ; a+0.1)
1.0	(0, 0, 1, n-1, 1 ; a)
1.2	(1, 0, 1, n-1, 1; a+0.8)
1.4	(1, 0, 0, n, 0; a+0.6)
1.8	(0, 1, 0, n, 0; a+0.2)
1.9	(0, 0, 1, n-1, 1; a+0.1)
2.2	(1, 0, 1, n-1, 1; a+0.8)
2.3	(1, 0, 0, n, 0; a+0.7)
2.7	(0, 1, 0, n, 0; a+0.3)
2.8	(0, 0, 1, n-1, 1; a+0.2)
3.2	(1, 0, 0, n, 0; a+0.8)
3.6	(0, 1, 0, n, 0; a+0.4)
3.7	(0, 0, 1, n-1, 1; a+0.3)
4.1	(0, 0, 0, n, 0; a-0.1)

Table 2: iteration for $n \geq 1$ and $0.1 < a \leq 0.2$

The above sequence of states proves the lemma for initial states of the form $(0, 1, 0, n, 0; a)$ with $n \geq 1$ and $0.1 < a \leq 0.6$, by examination of the state in the table at times 0, 0.9, 1.8, 2.7, and 3.6, since all such states eventually enter the state at time 4.1 in the table, which is of the form in Lemma 4.3. Thus, the lemma is proved for $n \geq 1$ and $0.1 < a \leq 0.6$.

For the case $n = 0$ and $0.1 < a \leq 0.5$, we have the following sequence of states:

Time	State
0	(0, 1, 0, 0, 0; a)
0.1	(0, 0, 1, 0, 0; a-0.1)
a	(1, 0, 1, 0, 0; 1)
0.5	(1, 0, 0, 1, 0; a+0.5)
0.9	(0, 1, 0, 1, 0; a+0.1)

The last state of the table is a type 2 state with $n = 1$ and $0.2 < a \leq 0.6$ and we have already proven the lemma for this case, via Table 2. In the case $n = 0$ and $0.5 < a \leq 0.6$, we note that at $t = 0.5$, the network will enter an a-orbit.

Finally, we need to prove the lemma for the remaining cases, i.e. an initial state of the form $(0, 1, 0, n, 0; a)$ with $n \geq 1$ and $0 < a \leq 0.1$ and an initial state of the form $(0, 1, 0, 0, 0; a)$ with $0 < a \leq 0.1$. For the former case, we iterate the network from such a starting state below.

Time	State
0	(0, 1, 0, n, 0; a)
a	(1, 1, 0, n, 0; 1)
0.1	(1, 0, 1, n-1, 1; a+0.9)
0.5	(0, 1, 1, n-1, 0; a+0.5)
0.6	(0, 0, 2, n-1, 0; a+0.4)
0.9	(0, 0, 1, n, 0; a+0.1)
1.3	(1, 0, 0, n+1, 0; a+0.7)
1.7	(0, 1, 0, n+1, 0; a+0.3)

At time $t = 1.7$ we see that the network has entered a state which is qualitatively the same as time 1.8 in Table 2, hence the network will enter an a-orbit, as the conclusion of the lemma requires, in another 2.3 minutes. For the latter case, we have

Time	State
0	(0, 1, 0, 0, 0; a)
a	(1, 1, 0, 0, 0; 1)
0.1	(1, 0, 1, 0, 0; a+0.9)
a+0.4	(0, 1, 1, 0, 0; 0.6)
a+0.5	(0, 0, 2, 0, 0; 0.5)
a+0.8	(0, 0, 1, 1, 0; 0.2)
a+1.0	(1, 0, 1, 1, 0; 1)
a+1.2	(1, 0, 0, 2, 0; 0.8)
a+1.6	(0, 1, 0, 2, 0; 0.4)

the last state is a state in Table 2 at time 1.8 with $n = 2$ and $a = 0.2$. □

Lemma A.1. *Consider a multiclass queueing network as defined in Section 2 of Dai [10]. Assume that the strong laws of large numbers (2.1)-(2.3) of [10] holds for the external arrival, service and routing processes. Assume further that the traffic intensity at a given station is less than one.*

Then, with probability one, the station empties infinitely often when the network is operated under any non-idling service policy.

Proof. We follow the notation in [10]. Let ω be a sample path on which (2.1)-(2.3) of [10] hold. Consider a fluid limit $(\bar{Z}, \bar{T})$ along the sample path as in [10]. (Here, we use $\bar{Z}$, instead of $\bar{Q}$ in [10], to denote the limiting queue length process.) The fluid limit must satisfy the fluid model equations (4.1)-(4.6) of [10].

Now suppose that we have $\rho_i < 1$ at station i, but station i never empties after some time $t_0 \geq 0$. Since the network is operating under a non-idling service policy, station i is never idle after this finite time t_0. Passing this property to the fluid limit, we have $\sum_{k:\sigma(k)=i} \bar{T}_k(t) = t$ for $t > 0$. (Time t_0 in the queueing network corresponds to time 0 in a fluid limit.) Let $\bar{Q}(t) = (I - P')^{-1}\bar{Z}(t)$. We have

$$\begin{aligned}
\sum_{k:\sigma(k)=i} \frac{1}{\mu_k}\bar{Q}_k(t) &= \rho_i t - \sum_{k:\sigma(k)=i} \bar{T}_k(t) \\
&= \rho_i t - t = (\rho_i - 1)t \\
&< 0
\end{aligned}$$

for $t > 0$, leading to a contradiction.

□

We repeatedly use the following large deviations estimates in the proofs of Section 5. An elementary proof of Lemma A.2 can be found in Shwartz and Weiss [29], Section 1.2.

Lemma A.2. *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)

$$\mathbb{P}\{Y_n < mn - \alpha n\} \leq \exp(-\epsilon n).$$

Lemma A.3. *Consider a $GI/GI/1$ queue with i.i.d. interarrival times $\{u_i\}$ and i.i.d. service times $\{v_i\}$. Assume that $\mathbb{E}[u_1] < \mathbb{E}[v_1]$, the queue is empty at time zero, and that the first arrival occurs at time 0. Assume further that for some $\kappa > 0$, $\mathbb{E}[\exp(\kappa(u_1 + v_1))] < \infty$. Then for $0 < \delta < 1$, there exists a constant $\epsilon > 0$ such that for n sufficiently large (in particular we take $\lfloor \delta\sqrt{n} \rfloor > 0$),*

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

where S_n is the arrival time of the nth job.

Proof.

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

The last inequality follows from Lemma A.2 and the fact that $\mathbb{E}[v_1 - u_1] > 0$. Thus,

$$
\begin{aligned}
&\mathbb{P}\{\text{queue first empties in } [S_{\lfloor \delta\sqrt{n} \rfloor}, S_n]\} \\
&\quad \le \sum_{i=\lfloor \delta\sqrt{n} \rfloor}^{n} \exp(-\epsilon_1 i) \\
&\quad \le n \exp(-\epsilon_1 \lfloor \delta\sqrt{n} \rfloor) \\
&\quad \le \exp(-\epsilon_2 \lfloor \delta\sqrt{n} \rfloor) \qquad \text{(for large enough } n\text{)} \\
&\quad \le \exp(-\epsilon\sqrt{n}).
\end{aligned}
$$

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