EURANDOM PREPRINT SERIES
2011-032

Separation of timescales in a two-layered network

Maria Vlasiou, Jiheng Zhang, Bert Zwart, Rob van der Mei
ISSN 1389-2355

Separation of timescales in a two-layered network

Maria Vlasiou*, Jiheng Zhang†, Bert Zwart‡, Rob van der Mei‡
*Department of Mathematics and Computer Science
Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
Email: m.vlasiou@tue.nl
†Department of Industrial Engineering and Logistics Management
Hong Kong University of Science and Technology, Hong Kong, S.A.R, China
Email: j.zhang@ust.hk
‡Centrum Wiskunde & Informatica
Science Park 123,Amsterdam, The Netherlands
Email: (bert.zwart, mei)@cwi.nl

***Abstract*—We investigate a network consisting of two layers occurring in, for example, application servers, and model the first layer as a many-server Jackson network. Active servers acts as customers at the second layer, where they are served by a common CPU. This system provides a benchmark example of a layered system. Our main result shows a separation of time scales in heavy traffic: the main source of randomness occurs at the (aggregate) CPU level; the interactions between different types of nodes at the other level is shown to converge to a fixed point at a faster time scale; this also yields a state-space collapse property. Apart from these fundamental insights, we also obtain an explicit approximation for the joint law of the system which is provably accurate for heavily loaded systems, and performs numerically well for moderately loaded systems. The obtained results for the model under consideration can be applied to thread-pool dimensioning in application servers.**

I. Introduction

Communication networks need to support a growing diversity and heterogeneity in applications. Examples are web-based multi-tiered system architectures, with a client tier to provide an interface to end users, a business logic tier to coordinate information retrieval and processing, and a data tier with legacy systems to store and access customer data. In such environments, different applications compete for access to shared infrastructure resources, both at the software level (e.g., mutex and database locks, thread-pools) and at the hardware level (e.g., bandwidth, processing power, disk access). Thus, the performance of such applications is determined by the interplay of software and hardware contention. For background, see [1], [2].

In particular, in situations where web pages are created on-the-fly (think of making a reservation online), the benefits of caching are limited and sizes of web pages are unknown, and there is usually ample core network bandwidth available at reasonable prices. Consequently, the bottleneck in user-level performance can shift from the network interface to the application server, and implementing size-based scheduling policies becomes hard, contrary to the situation considered in [3], [4].

Application servers usually implement a number of thread-pools; a thread is software that can perform a specific type of sub-transaction. Consider for example the web-server performance model proposed in [1]. Each HTTP request that requires server-side scripting (e.g., CGI or ASP scripts, or Java servlets) consists of two subsequent phases: a document-retrieval phase, and a script processing phase. To this end, the web server implements two thread-pools, one performing the first phase of processing, and the other performing the second phase of processing. The model consists of a tandem of two multi-server queues, where servers at queue 1 represent the phase-1 threads, and the servers at queue 2 represent phase-2 threads. A particular feature of this model is that at all times the active threads share a common Central Processing Unit (CPU) hardware in a Processor-Sharing (PS) fashion; c.f. [5], [6]. Alternatively, one can think of scheduling jobs in data centers, where different parts of a job are taken care of by a different thread-pool.

Motivated by this, we study a relatively simple, but non-trivial two-layered network. An informal model description is as follows. The first layer looks like a (generalized) Jackson network consisting of many-server queues. The servers in this network act as customers in a second layer, in the sense that they are served by a single CPU in a PS fashion. A detailed model description is provided in Section II.

Variations of the above model have been investigated in several papers in the literature, but apart from stability analysis [5], a rigorous analysis of this layered network has been lacking. The same can be said about other literature on layered networks. Only a limited number of papers focus on the performance of multi-layered queuing networks. A fundamental paper is Rolia and Sevcik [7], who propose the Method of Layers, i.e., a closed queuing-network based model for the responsiveness of client-server applications, explicitly taking

The research of Maria Vlasiou and Jiheng Zhang is partly supported by two grants from the 'Joint Research Scheme' program, sponsored by the Netherlands Organization of Scientific Research (NWO) and the Research Grants Council of Hong Kong (RGC) through projects D-HK007/11T and 600.649.000.10N006. The research of Bert Zwart is partly supported by an NWO VIDI grant and an IBM faculty award.

into account both software and hardware contention. Another fundamental contribution is presented by Woodside et al. [8], who propose to use the so-called Stochastic Rendezvous Network model to analyze the performance of application software with client-server synchronization. The contributions presented in [7] and [8] are often referred to as Layered Queuing Models. A common drawback of multi-layered queuing models is that exact analysis is primarily restricted to special cases, and numerical algorithms are typically required to obtain performance measures of interest (see for example [8]). Although such methods are important, it is also valuable to look at layered systems from a more qualitative point of view, which we do in this paper by considering the system under critical load.

The most simple example of the layered systems we consider is the case where the first layer consists of a single node. In this case, the model reduces to the so-called limited processor sharing (LPS) queue. Recently, there has been considerable interest in the analysis of LPS systems. Avi-Itzhak and Halfin [9] propose an approximation for the mean response time. A computational analysis based on matrix geometric methods is performed in Zhang and Lipsky [10], [11]. Some stochastic ordering results are derived in Nuyens and van de Weij [12]. Large deviation results are presented in Nair *et al.* [13], and these results are also applied to show that LPS provides robust performance across a range of both heavy-tailed and light-tailed job sizes, as it combines the attractive properties of a guaranteed service rate of FIFO and the possibility of overtaking offered by PS.

The work on LPS that is most relevant for this study is the work of Zhang, Dai and Zwart [14]–[16] who study the stochastic processes that underlie the LPS queue in the heavy traffic regime, i.e. an asymptotic regime where the traffic intensity converges to 1. The setting is rather general, allowing the inter-arrival and service times to have general distributions. Fluid and diffusion limits are derived, leading to a heavy traffic analysis of the steady-state distribution of LPS, showing that the approximation by Avi-Itzhak and Halfin [9] is asymptotically accurate in heavy traffic.

In the present paper, we perform an analysis similar to the one performed in [14]–[16], generalizing from a single node to networks, although our mathematical results are derived under the assumption that job sizes are exponential; however, we do propose an extension based on these mathematical results. We analyze the system as it approaches heavy traffic. Under the assumption that there is a single bottleneck (an exact definition of bottleneck is given later), we derive explicit results for the joint distribution of the number of jobs in the system, by proving a diffusion limit theorem. This limit theorem does not only yield explicit approximations but yields also useful insights: if we look at the system from the CPU layer, we can aggregate the whole system since the total workload acts as if we were dealing with a single server queue. However, the interaction of several types of customers at the other layer would then be lost. It turns out, nonetheless, that those interactions take place at a much faster time scale in heavy traffic, and that the number of users of all types converge instantaneously to a piece-wise linear function of the number of users at the bottleneck. This separation of time scales property is shown to imply that in heavy traffic, the joint queue length vector can be written as a deterministic function of the total workload as seen from the CPU layer. Such a property is known as *state-space collapse* (SSC) in the stochastic network literature.

Thus, our methodological contribution is that it is possible to rigorously establish a separation of time scales property in heavy traffic in an important class of layered networks, which makes these layered networks tractable. Although we focus on the Markovian case, we believe that such properties hold more generally as well; we provide some physical and numerical arguments to support this claim. The result on separation of time scales result essentially implies that the main source of randomness in heavy traffic can be observed at the CPU layer, thus making performance analysis much more tractable. Apart from supporting these claims by theorems, some numerical experiments suggest that the resulting approximations perform well. The results in our paper may be useful to create design rules, for example to dimension thread-pools. Some first efforts using heuristic approximations were proposed in [6].

The paper is organized as follows. We provide a detailed model description in Section II. In Section III we propose a fluid model for our two-layered system. We use this fluid model to analyze how users of different types interact if the system is in heavy traffic. In doing so, we construct a Lyapounov function which we use to show that the user population converges uniformly to a fixed point that is uniquely defined through the total workload. The fluid model also helps understand which stations will be bottlenecks. Section IV contains our main results, namely a process limit theorem for the customer population process. A heavy traffic approximation of the steady-state distribution is proposed in Section V. Section VI presents an extension to general service times based on physical arguments, and some numerical validation by comparing the proposed approximations with simulation results. Concluding remarks can be found in Section VII. Additional proofs can be found in the appendix.

II. Model description

The purpose of this section is to give a formal model description. We adopt the convention that all vectors are column vectors, and use a^T to denote the transpose of a vector or matrix. For vectors x, y we denote xy to be the vector consisting of elements $x_i y_i$. Furthermore, I is the identity matrix, e is the vector consisting of 1's, and e_i is the vector whose ith element is 1 and the rest are all 0.

We consider a network with J nodes. Jobs arrive at node $i \in \{1, ..., J\}$ according to a Poisson process of rate λ_i, and are served at rate μ_i at that node. Each node has K_i servers. Customers move between queues according to a substochastic routing matrix P of dimension J. As in the case of regular queuing networks, we need to introduce the actual arrival rates of jobs to station i, which are denoted by γ_i and are a solution

to the traffic equation

$$\gamma = \lambda + P^T\gamma.$$

Throughout the paper, we need to assume that $I - P$ is invertible as is usual for open Jackson networks, so we have that $\gamma = (I - P^T)^{-1}\lambda$. All active servers interact since they acquire their capacity from a CPU working at rate 1.

It can be useful to view the system from the CPU layer, since there is a connection with an $M/PH/1$ queue which we now describe: users arrive at rate $\lambda = \sum_i \lambda_i$ and start their service at node i with probability $a_i = \lambda_i/\lambda$. Define $a_0 = 0$, $p_{00} = 1$, and for $i \geq 1$, $p_{0i} = 0$ and $p_{i0} = 1 - \sum_j p_{ij}$. Observe that the total service requirement of a job is the time to absorption in state 0 of a continuous-time Markov chain with initial distribution (a_i), where the sojourn time in state i is exponentially distributed with rate μ_i, after which one jumps to state j with probability p_{ij}. Thus, the total service requirement S of an arbitrary customer has a phase-type distribution with parameters (a, μ, P). We also denote by $\beta_i = 1/\mu_i$ and $\beta_i^{(2)} = 2/\mu_i^2$ the first and second moment of service requirements at node i. The corresponding vectors are denoted by β and $\beta^{(2)}$.

It is possible to compute the first two moments of this distribution by using standard methods (see for e.g. [17] and references therein). Let T_i be the total service requirement of users waiting to be served at node i. This includes their immediate service at node i and all the future services due to routing. Denote by τ_i and $\tau_i^{(2)}$ the first and second moment of T_i. Then $\tau = (1 - P)^{-1}\beta$ and

$$\tau_i^{(2)} = \beta_i^{(2)} + \sum_j p_{ij}(2\beta_i\tau_j + \tau_j^{(2)}).$$

In vector notation, this becomes

$$\tau^{(2)} = (I - P)^{-1}\left(\beta^{(2)} + 2\beta(P\tau)\right).$$

Notice that the final formulae are still valid if the service requirement of a user at node i is not exponential, but is generally distributed. In that case, the total service requirement is simply the time to absorption of a semi-Markov process. We need this interpretation in Section VI. Of course, in that case, it no longer holds that $\beta_i^{(2)} = 2/\mu_i^2$.

We can compute the first and the second moment of the total service requirement S, obtaining

$$E[S] = a^T\tau \text{ and } E[S^2] = a^T\tau^{(2)}.$$

It is also clear from the $M/PH/1$ interpretation that the global stability condition of the system is $E[S]\sum_i \lambda_i < 1$, or equivalently

$$\rho := \lambda^T(I - P)^{-1}\beta = \beta^T\gamma < 1.$$

We also define $\rho_i = \beta_i\gamma_i = \gamma_i/\mu_i$. Observe that $\rho = \sum_i \rho_i$.

Example: We are particularly interested in the simple two-node tandem case ($J = 2$), where all users first enter station 1 ($\lambda_2 = 0$), then move from station 1 to station 2 ($p_{12} = 1$) and then leave ($p_{20} = 1$). In this case $\gamma_1 = \gamma_2 = \lambda_1$, $E[S] = 1/\mu_1 + 1/\mu_2$, and

$$E[S^2] = 2/\mu_1^2 + 2/(\mu_1\mu_2) + 2/\mu_2^2.$$

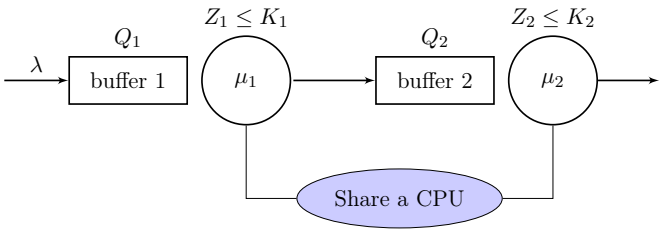

Fig. 1. LPS queues in tandem with a shared CPU

We now investigate the system under critical load, i.e. when ρ is (close to) 1. To this end, we first develop and analyze a critical fluid model in the next section.

III. Fluid analysis and invariant points

In this section we propose a fluid model for our layered system under the assumption of critical loading, i.e. $\rho = 1$, or equivalently,

$$\sum_i \frac{\gamma_i}{\mu_i} = 1. \tag{1}$$

In the sequel, we establish that the workload will stay constant, and that the queue length vector will converge to an invariant point. We also characterize the set of invariant points, and show this set is one-dimensional under the assumption that there is a unique bottleneck.

Our fluid model is defined by the following ordinary differential equation (ODE):

$$\bar{X}_i'(t) = \lambda_i - \mu_i R_i(\bar{X}(t)) + \sum_{j=1}^{J} p_{j,i}\mu_j R_j(\bar{X}(t)).$$

We can write this into vector form

$$\bar{X}' = \Psi(\bar{X}), \tag{2}$$

where $\Psi : [0, \infty)^J \to \mathbb{R}^J$ can be represented as

$$\Psi(x) = \lambda - \mu R(x) + P^T(\mu R(x)), \tag{3}$$

where $\mu R(x)$ means component-wise product.

Theorem 1 (Existence and uniqueness)**.** *For any $\bar{X}(0) = x \in \mathbb{R}_+^J$, there exist a unique solution to the ODE* (2)*.*

Proof: It is clear that each $R_i(x)$ is Lipschitz continuous on $\mathbb{R}_+^J$. So is the linear combination $\Psi(x)$. The result follows from Theorem VI in Chapter 10 of [18]. ■

Recall that the system is a work-conserving single-server queue when considered at the CPU layer. We now show that this is also the case for our fluid model. We define the workload for the fluid model as follows:

$$\bar{W}(t) = \beta^T(1 - P^T)^{-1}\bar{X}(t). \tag{4}$$

Proposition 1. *For each solution of* (2)*,* $\bar{W}(t) = \bar{W}(0)$.

Proof: The proof is quite straightforward. From (2),

$$\begin{aligned}\bar{W}'(t) &= \beta^T(I-P^T)^{-1}\bar{X}'(t)\\ &= \beta^T(I-P^T)^{-1}\left(\lambda - \mu R(x) + P^T(\mu R(x))\right)\\ &= \beta^T\gamma - \beta^T(I-P^T)^{-1}(I-P^T)\mu R(\bar{X}(t))\\ &= 1 - \beta^T\mu R(\bar{X}(t)) = 1-1=0,\end{aligned}$$

where $\beta^T\gamma = 1$ is due to critical loading and $\beta^T\mu R(x) = \sum_{i=1}^J R_i(x)$ which equals 1 by the definition of $R(x)$. ■

We now characterize the invariant manifold of the ODE, which is the set of invariant points. A point x is invariant if

$$\mu_i R_i(x) = \gamma_i, \quad i = 1,\dots,J. \tag{5}$$

A crucial notion in the study of invariant points is the notion of bottleneck. It turns out that the following definition is appropriate:

Definition 1 (Bottleneck)**.** *Node* i *is the bottleneck if* $i = \arg\min_j \frac{\mu_j K_j}{\gamma_j}$.

In this paper, we focus on the case where there is a unique bottleneck. Without loss of generality, we take node 1 as the bottleneck when we investigate the case of a general network; in the two-node tandem case we sometimes take node 2 as the bottleneck if we do numerical experiments.

There are two cases: if $x_1 \le K_1$ then it follows from (5) that

$$\mu_i x_i = \gamma_i \sum_j x_j.$$

Thus, $\sum_j x_j = \mu_1 x_1/\gamma_1$, so that $\mu_i x_i = \gamma_i \frac{\mu_1 x_1}{\gamma_1}$. If $x_1 \ge K_1$ then we can write $\mu_i x_i = \gamma_i \dfrac{\mu_1 K_1}{\gamma_1}$. In general, for $i = 2,\dots,J$, we have $x_i = \dfrac{\gamma_i}{\mu_i}\dfrac{\mu_1(x_1 \wedge K_1)}{\gamma_1}$. Thus, the set of invariant points, called the invariant manifold, is the following:

$$\mathcal{I} = \left\{ x \in \mathbb{R}_+^J : \frac{\mu_i x_i}{\gamma_i} = \frac{\mu_1(x_1 \wedge K_1)}{\gamma_1}, i = 2,\dots,J \right\}.$$

The invariant manifold is illustrated in the following picture for the two dimensional case.

We now conclude by formally showing that our notion of invariant points makes indeed sense.

Proposition 2. $\bar{X}(t) = \bar{X}(0)$ *for all* $t \ge 0$ *if and only if* $\bar{X}(0) \in \mathcal{I}$.

Proof: The necessity part follows from the above discussion. For sufficiency, it suffices to show that for any $x \in \mathcal{I}$, $\Psi(x) = 0$. Note that for any $x \in \mathcal{I}$, let $c = \frac{\mu_i x_i}{\gamma_i}$, $i = 2,\dots,J$. By (1), $x_1 \wedge K_1 + \sum_{i=2}^J x_i = \sum_{i=1}^J c\frac{\gamma_i}{\mu_i} = c$. So $R_i(x) = \frac{x_i}{c}$, thus $\mu_i R_i(x) = \gamma_i$. We have

$$\begin{aligned}\Psi(x) &= \lambda - \mu R(x) + P^T(\mu R(x))\\ &= (I-P^T)(\gamma - \mu R(x)) = 0.\end{aligned}$$

■

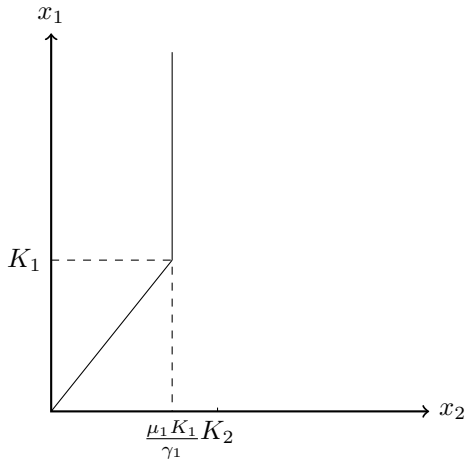

Fig. 2. Invariant manifold for the 2-dimensional tandem case.

A. Convergence to invariant points

We now define a Lyapunov function to show that the solution of the ODE (2) converges to the invariant manifold $\mathcal{I}$. Let x^* be the point in the invariance manifold with $x_1^* = K_1$. We define a critical workload level

$$w^* = \beta^T(1-P^T)^{-1}x^* = \beta^T(1-P^T)^{-1}(\beta\gamma)\frac{\mu_1}{\gamma_1}K_1.$$

(Yet another interpretation is $w^* = \tau^T x^*$.) This gives rise to a "critical hyperplane":

$$\{x : \beta^T(1-P^T)^{-1}x = w^*\}. \tag{6}$$

The idea is simple. The Lyapunov function is constructed based on the distance from the point x to the invariant manifold depending on whether x is above the critical hyperplane or below. For any $w \le w^*$, let for $i = 1,\dots,J$

$$x_i^\dagger(w) = \frac{\gamma_i w}{\mu_i \beta^T(I-P^T)^{-1}(\beta\gamma)} = \frac{\mu_1 K_1}{\gamma_1}\frac{\gamma_i}{\mu_i}\frac{w}{w^*}. \tag{7}$$

Note that

$$R_i(x^\dagger(w)) = \frac{x_i^\dagger(w)}{\sum_i x_i^\dagger(w)} = \frac{\gamma_i/\mu_i}{\sum_j \gamma_i/\mu_i} = \frac{\gamma_i}{\mu_i}.$$

This is an intuitive result, since on the invariant manifold, the R_i's, representing outflow of work at station i, should be equal to the inflow of work at station i. For any $w > w^*$, let

$$x_1^\dagger(w) = K_1 + \frac{(w-w^*)}{\beta^T(I-P^T)^{-1}e_1} = K_1 + \frac{(w-w^*)}{\tau_1} \tag{8}$$

$$x_i^\dagger(w) = \frac{\mu_1 K_1}{\gamma_1}\frac{\gamma_i}{\mu_i}, \quad i = 2,\dots,J. \tag{9}$$

It is clear that $x^\dagger(w)$ is the intersection of the workload hyperplane $\bar{W}(t) = w$ and the invariant manifold. Depending on whether w is larger than w^* or not, the calculation is different. We can define a Lyapunov function:

$$\begin{aligned}&\mathcal{L}(\bar{X}(t))\\ &= \left[\left(\bar{X}(t) - x^\dagger(w)\right)^T (I-P^T)^{-1}\left(\bar{X}(t) - x^\dagger(w)\right)\right].\end{aligned} \tag{10}$$

We now state the following property for the Lyapunov function, which will imply the convergence to the invariant manifold for the fluid model.

Proposition 3. *$\mathcal{L}(x)$ is continuous in x and for any $x \notin \mathcal{I}$, $\mathcal{L}(x) > 0$; for any $x \in \mathcal{I}$, $\mathcal{L}(x) = 0$. When $\bar{X}(t) \notin \mathcal{I}$,*

$$\frac{d}{dt}\mathcal{L}(\bar{X}(t)) < 0.$$

A proof of this result can be found in the appendix. The following proposition is an immediate consequence.

Proposition 4 (Convergence to the invariant manifold)**.** *For any solution $\bar{X}$ to the ODE* (2)*, we have that*

$$\bar{X}(t) \to x^{\dagger}(w), \quad \text{as } t \to \infty,$$

where $x^{\dagger}$ is as defined by (7)–(9) *and $w = \bar{W}(0)$.*

In the following section we will show that, as ρ is close to 1, the fluid model is a good approximation of the queue length on a time scale of $O(1/(1-\rho))$. Since the diffusion time scale is of $O(1/(1-\rho)^2)$ it is tempting to conclude that the only configurations of the customer populations that matter are configurations on the invariant manifold. These configurations depend on the workload w at the CPU, which then is expected to be the driving force of randomness. The goal of the next section is to make this statement rigorous.

IV. State-space collapse in heavy traffic

We are now ready to develop a diffusion approximation for the process describing the number of customers in the system, which we sometimes also refer to as the head-count process. Consider a sequence of such processes indexed by n. As $n \to \infty$,

$$\lambda^n \to \lambda.$$

Let $\gamma^n = (I - P^T)^{-1}\lambda^n$, and

$$\rho^n = (\gamma^n)^T \frac{1}{\mu}.$$

We assume that as $n \to \infty$, for each $i = 1, \ldots, J$,

$$n(1-\rho^n) \to \theta > 0, \text{ and } \frac{K_i^n}{n} \to K_i. \tag{11}$$

We are interested in the limit of the diffusion scaled process

$$\hat{X}^n(t) = \frac{1}{n} X^n(n^2 t)$$

in the heavy traffic regime.

As shown in Williams [19], a key step in obtaining a diffusion limit in heavy traffic is to establish a state-space collapse result. In our setting, SSC means that the diffusion-scaled, J-dimensional process is close to a deterministic function of the diffusion-scaled, one-dimensional workload process. The workload process is essentially equivalent to that of a $G/G/1$ queue under any work-conserving policy. By a classical result (c.f. [17]), the limiting workload process is a one-dimensional reflected Brownian motion (RBM).

Our proof strategy is analogous to the modular approach proposed in Bramson [20] and Williams [19]. We already have studied a critically loaded fluid model in Section III. The critically loaded fluid model exhibits an SSC: each fluid model solution converges to an invariant state in some uniform sense, and each invariant state has an SSC.

We adopt Bramson's framework in [20] to translate the fluid model SSC result into the diffusion-scaled SSC result. The diffusion scaled process on the interval $[0, T]$ corresponds to the unscaled process on the interval $[0, n^2 T]$. Fix a constant $L > 1$,then the interval $[0, n^2 T]$ is covered by the $\lfloor nT \rfloor + 1$ overlapping intervals

$$[nm, nrm + nL] \quad n = 0, 1, \cdots, \lfloor nT \rfloor.$$

On each of these intervals, the diffusion scaled process can be viewed as a shifted, fluid-scaled process defined by

$$\bar{X}^{n,m}(t) = \frac{1}{n} X^n(nm + nt) \quad 0 \le t \le L.$$

To carry out the program suggested by Bramson and Williams in our particular case, we need to show that (a) each limit from the family of shifted, fluid-scaled processes is a solution to the fluid model (such a limit is called a *fluid limit* in this paper, but is also known as a "cluster point" [20].); (b) the set of fluid limits is "rich": each shifted, fluid-scaled process is close to the fluid model introduced in the previous section. A major step to proving (a) and (b) is to show, the precompactness of the shifted fluid scaled processes. Details can be found in the appendix.

We first establish that the shifted fluid scaled processes can be uniformly approximated by the fluid model solution studied in Section III.

Proposition 5. *Fix $T > 0$ and $L > 1$. The family of shifted fluid scaled process $\{\bar{X}^{n,m}(\cdot)\}_{m \le nT, n \in \mathbb{N}}$ is relatively compact in $\mathbf{D}([0, L], \mathbb{R}_+^J)$ (the space of right continuous function with left limits). Moreover, each limit of the weakly converging subsequence is a solution to the ODE* (3)*.*

The next step is to study the diffusion limit by establishing state-space collapse. Define the map $\Delta : R_+ \to \mathbb{R}_+^J$ by

$$\Delta_1(w) = \frac{w \wedge w^*}{w^*} K_1 + \frac{(w - w^*)^+}{\tau_1}, \tag{12}$$

$$\Delta_i(w) = \frac{w \wedge w^*}{w^*} \frac{\mu_1 K_1}{\gamma_1} \frac{\gamma_i}{\mu_i}, \quad i = 2, \ldots, J. \tag{13}$$

This map is called *lifting map*, as it will be used to construct the multi-dimensional limiting queue length process from the one-dimensional limiting workload process.

Proposition 6 (State-space collapse)**.** *Assume that*

$$|\hat{X}^n(0) - \Delta(\hat{W}^n(0))| \Rightarrow 0,$$

as $n \to \infty$ specified by (11)*. We have*

$$\sup_{t \in [0,T]} \left| \hat{X}^n(t) - \Delta(\hat{W}^n(t)) \right| \Rightarrow 0, \quad \text{as } n \to \infty. \tag{14}$$

Let $X^*(\cdot)$ be the diffusion limit of the head-count process, and $W^*(\cdot)$ be the diffusion limit of the workload process, in the heavy traffic limit specified by (11). Based on our

discussion on the equivalency with the $M/PH/1$ queue, W^* is known to be an RBM with drift $-\theta$ and variance $\sigma^2 = \mathbb{E}(S)(1+c_s^2) = E[S^2]/E[S]$, where $c_s^2 = Var(S)/\mathbb{E}^2(S)$. According to the calculation in Section II, $\sigma^2 = a^T\tau^{(2)}/(a^T\tau)$. We have the following result that fully characterizes the diffusion limit X^* based on W^*.

Theorem 2 (Diffusion limit)**.** *Under the same condition as in Proposition 6, the diffusion-scaled process $\hat{X}^n$ converges weakly to the limit X^* in heavy traffic. The limit X^* can be characterized as*

$$X_1^*(t) = \frac{W^*(t) \wedge w^*}{w^*} K_1 + \frac{(W^*(t) - w^*)^+}{\tau_1},$$
$$X_i^*(t) = \frac{W^*(t) \wedge w^*}{w^*} \frac{\mu_1 K_1}{\gamma_1} \frac{\gamma_i}{\mu_i}, \quad i = 2, \ldots, J.$$

Proof: Note that the mapping Δ defined by (12) and (13) is continuous. The result follows immediately from the continuous mapping theorem by invoking Proposition IV.2 and the fact that the workload process converges to $W^*(\cdot)$. ■

V. Steady-state performance approximations

It is well-known that the normalized steady-state workload of an $M/G/1$ queue in heavy traffic converges to an exponentially distributed random variable; i.e. if we consider the sequence of systems introduced in the previous section, and let $W^n(\infty)$ be the steady-state workload in the nth system, then

$$\hat{W}^n(\infty) \Rightarrow W^*(\infty),$$

where $W^*(\infty)$ is an exponentially distributed random variable with mean $m = \frac{\sigma^2}{2\theta}$, by the classical steady-state analysis of RBM.

Since $W^*(\infty)$ can also be seen as the limit (in distribution) of $W^*(t)$ as $t \to \infty$, it is natural to expect that the heavy traffic and steady-state limits can be interchanged. Without being able to go into the details due to space considerations, it turns out that it is possible to do this in the same way as has been carried out in [14], where uniform bounds (in n) on coupling times for (diffusion-scaled) work-conserving systems have been established, allowing a limit interchange to take place.

We can exploit this to derive a heavy traffic limit theorem for X^n, which is a J-dimensional random vector denoting the customer population in steady state in the nth system. We have the following result by the continuous mapping theorem:

$$\hat{X}^n(\infty) \Rightarrow X^*(\infty) := \Delta(W^*(\infty)).$$

Note that

$$\mathbb{P}(X_i^*(\infty) > x) = \mathbb{P}\left(\Delta_i(W^*(\infty)) > x\right)$$

Since the distribution of $W^*(\infty)$ is explicit, as is the mapping Δ, the above formula is explicit. Thus, we can develop explicit approximations for the original system that will be accurate in heavy traffic. For that, it is necessary to remove the index n from the limit theorem, as we observe a single process only: we observe ρ and K_i, rather than ρ^n and nK_i. Simply set $k_i = K_i(1-\rho)$ (this does not change the definition of the bottleneck), take $\theta = 1$, and modify w^* and the lifting map Δ by replacing K_i with k_i.

Observe that ρ_j/ρ is the probability that an arbitrary customer in the system is of type j, so that

$$\sum_j \rho_j \tau_j / \rho = E[S^2]/(2E[S]) = m.$$

After some rewriting we get

$$w^* = (1-\rho)\sum_j \rho_j \tau_j K_1/\rho_1 = (1-\rho)K_1 \rho m/\rho_1.$$

Since W^* is exponentially distributed with mean m, the probability of delay at the bottleneck becomes

$$p_d \approx P(W^* > w^*) \approx \rho^{K_1 \frac{\rho}{\rho_1}}. \tag{15}$$

In the second equation we used that $e^{-(1-\rho)} \approx \rho$ to obtain an approximation more in line with the single-node approximation proposed by [9]. Due to lack of space, we focus on one performance measure only. The expected total response time (i.e. the sojourn time) $E[V]$ of an arbitrary job can be computed using Little's law:

$$E[V] = E[\sum_j X_j]/\lambda \approx \frac{1/\lambda}{1-\rho} E[\sum_j \Delta_j(W^*)].$$

Straightforward computations, combined with the above approximations, yield

$$E[\sum_j \Delta_j(W^*)] \approx (1-p_d) + p_d \frac{m}{\tau_1}.$$

It makes sense to multiply the final approximation with ρ to obtain a result that would yield an exact expression for the single-node case, and from a heavy traffic point of view, this would still yield asymptotically optimal estimates. Thus, we obtain as final approximation:

$$E[V] \approx \frac{E[S]}{1-\rho}\left[(1-p_d) + p_d \frac{m}{\tau_1}\right]. \tag{16}$$

In the single node case for exponential job sizes, we have that $m = E[S] = \tau_1$ so our approximation indeed reduces to $E[S]/(1-\rho)$ which is the expected sojourn time in an $M/M/1$ queue. We now develop an extension valid for more general service times combining the insights of the heavy-traffic analysis of our network model with available results for the single node case.

VI. Extension to general job sizes

In [16], it was shown that, for general job sizes, the queue length process X^* in heavy traffic satisfies

$$X^*(t) = (W^*(t) - K\beta^e)^+/\beta + \min\{W^*(t), K\beta^e\}/\beta^e.$$

It is clear that this is a generalization of what is obtained in this paper for $J = 1$, but the structure is also clear: locally, the process behaves like a PS queue if the queue length is smaller than K, and like a FCFS queue when the queue length is larger than K. The fluid model is no longer a simple ODE, and [16]

develops a framework using measure-valued processes, which is beyond the scope of this paper.

We, therefore, only use the physical insight obtained from the above equation to construct an approximation of the invariant queue length vector. We let β and β^e be vectors corresponding to means, and the means of residual job sizes of type i, i.e. $\beta_i^e = \beta_i^{(2)}/2\beta_i$.

We need to slightly modify our notion of the lifting map to account for general job sizes, the only change being the definition of w^*. The expected residual service requirement of a job at node i will not be τ_i but $\zeta_i := \tau_i + \beta_i^e - \beta_i$, so that $w^* = \zeta^T x^*$. With this definition, we expect (12)–(13) to be correct for generally distributed job sizes.

Assuming this is indeed the case, we carry out the same procedure as in the previous section. We set $k_i = K_i(1-\rho)$, take $\theta = 1$, and modify w^* and the lifting map Δ by replacing K_i with k_i to get $w^* = (1-\rho)\sum_j \rho_j\zeta_j K_1/\rho_1$. As in the previous section, note that $\sum_j \rho_j\zeta_j/\rho = E[S^2]/(2E[S]) = m$, so that we again obtain $p_d \approx P(W^* > w^*) \approx \rho^{K_1\frac{\rho}{\rho_1}}$, and $E[V] \approx \frac{E[S]}{1-\rho}\left[(1-p_d) + p_d\frac{m}{\tau_1}\right]$.

If $J = 1$ and $K_1 = \infty$, we have $p_d = 0$, so the resulting approximation formula for $E[V]$ indeed reduces to the exact expression for $E[V] = E[S]/(1-\rho)$ in the case of PS. If $J = 1$ and $K_1 = 1$, our approximation for p_d simplifies to ρ so that

$$E[V] \approx \frac{E[S]}{1-\rho}\left((1-\rho) + \lambda m\right) = E[S] + \frac{\rho}{1-\rho}m,$$

which equals the Pollaczek-Khintchin formula of the sojourn time of a user in the $M/G/1$ queue. Our approximation thus coincides with the two special cases, for which an exact expression is known. For $J = 1$ and arbitrary values of K_1, it reduces to the approximation given in [9] for a single node.

As mentioned before, proving that our approximation is asymptotically exact for general service times is beyond the scope of this paper. We, therefore, validate our approximation with some simulation results in the two-node tandem case. For the two-node tandem case we have $\zeta_2 = \beta_2^e$, $\zeta_1 = \beta_1^e + \beta_2$, $\gamma_1 = \gamma_2 = \lambda$.

In addition, $p_d \approx \rho^{\rho K_{i^*}/\rho_{i^*}}$ if node i^* is the bottleneck. This leads to

$$E[V] \approx \frac{E[S]}{1-\rho}\left[(1-p_d) + p_d m\frac{1}{\tau_{i^*}}\right].$$

We now present some numerical results for the case that both service times follow a hyper-exponential distribution. In all examples, we focus on a moderately loaded system with $\rho = 0.7$. We let the coefficient of variation of the service times range from 4 to 10 at both nodes (in fact we take the same parameters as done in the experiment of [6]).

Generally, the heavy-traffic approximations are quite accurate, always within 15% of the outcome predicted by simulation, and in several cases the error is as small as 2%. We find that the results become less accurate if the coefficient of variation of the service time at the bottleneck is high.

TABLE I
SIMULATION RESULTS

$(\beta_1, \beta_2, c_1^2, c_2^2, K_1, K_2)$	approximation	simulation
(1, 2, 4, 4, 3, 7)	10.24	10.41
(1, 2, 4, 10, 4, 6)	11.37	10.71
(1, 2, 10, 4, 4, 6)	10.77	10.57
(1, 2, 10, 10, 4, 6)	11.58	10.87
(2, 1, 4, 4, 6, 4)	10.24	10.49
(2, 1, 4, 10, 6, 4)	10.38	10.70
(2, 1, 10, 4, 6, 4)	10.78	10.98
(2, 1, 10, 10, 6, 4)	10.91	11.18
(1, 10, 4, 4, 2, 8)	38.86	37.43
(1, 10, 4, 10, 2, 8)	43.20	37.83
(1, 10, 10, 4, 2, 8)	38.91	37.53
(1, 10, 10, 10, 2, 8)	43.24	37.97
(10, 1, 4, 4, 8, 2)	38.52	38.88
(10, 1, 4, 10, 8, 2)	38.56	39.11
(10, 1, 10, 4, 8, 2)	42.46	40.77
(10, 1, 10, 10, 8, 2)	42.50	41.00

VII. CONCLUDING REMARKS

By establishing fluid and diffusion approximations of a two-layered queuing network, we have shown that, under critical loading, different layers in the network operate at different time scales. From the macroscopic CPU point of view, the system behaves like a simple one-server queue, which when critically loaded fluctuates at time scale of $O(1/(1-\rho)^2)$, and the network dynamics taking place at the other layer evolve at a faster time scale $O(1/(1-\rho))$, thus always reaching an invariant point as if the total workload at the CPU is constant.

We have established this result by introducing fluid and diffusion approximation techniques to study layered networks. It is interesting to examine the potential of such techniques to analyze other layered networks, such as those in [7], [8].

For our model, state-space collapse was established as a consequence of the single bottleneck assumption. Driven by curiosity, we are currently extending the analysis to multiple bottlenecks, although we note that the single bottleneck assumption will typically be an artefact of the fact that the buffer sizes K_i need to be chosen as integers in implementations.

Another interesting topic is to allow for general job sizes, as well as time-varying arrival rates. Finally, we expect the results to be directly useful to dimension thread-pools in web servers in a static fashion. The techniques in this paper are likely to be useful for dynamic thread-pool dimensioning as well, as the application of the techniques in this paper seems promising to formulate tractable (Brownian) control problems.

REFERENCES

[1] R. van der Mei, R. Hariharan, and P. Reeser, "Web server performance modeling," *Telecommunication Systems*, vol. 16, pp. 361–378, 2001.

[2] V. Cardellini, E. Casalicchio, M. Colajanni, and P. Yu, "The state of the art in locally distributed web server systems," *ACM Computing Surveys*, vol. 34, pp. 263–Ű311, 2002.

[3] M. Crovella, R. Frangioso, and M. Harchol-Balter, "Connection scheduling in web servers," in *Proceedings USENIX symposium on Internet Technologies and Systems*, 1999.

[4] M. Harchol-Balter, B. Schroeder, N. Bansal, and N. Agrawal, "Srpt scheduling for web servers," *Lecture Notes in Computer Science*, vol. 2221, pp. 11–21, 2001.

[5] M. Jonckheere, R. van der Mei, and W. van der Weij, "Stability and throughput for two-layered queueing networks," *Performance Evaluation*, vol. 67, pp. 28–42, 2010.

[6] W. van der Weij, R. van der Meia, and B. G. F. Phillipson, "Optimal server assignment in a two-layered tandem of multi-server queues," in *Proceedings 3rd International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HETNETS), volume P01, Ilkley, England*, July 2004.

[7] J. Rolia and K. Sevcik, "The method of layers," *IEEE Transactions on Software Engineering*, vol. 21, 689–699.

[8] C. Woodside, J. Neilson, D. Petriu, and S. Majumdar, "The stochastic rendezvous network model for the performance of synchronous client-server like distributed software," *IEEE Transactions on Computers*, vol. 44, pp. 20–34, 1995.

[9] B. Avi-Itzhak and S. Halfin, "Expected response times in a non-symmetric time sharing queue with a limited number of service positions," in *Proceedings of the 12th International Teletraffic Congress*, Torino, 1988.

[10] F. Zhang and L. Lipsky, "Modelling restricted processor sharing," in *Proc. of the 2006 Int'l Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA06)*, 2006.

[11] ——, "An analytical model for computer systems with non-exponential service times and memory thrashing overhead," in *Proc. of the 2007 Int'l Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA07)*, 2007.

[12] M. Nuyens and W. van der Weij, "The limited processor sharing queue," CWI, Amsterdam, Tech. Rep., 2007.

[13] J. Nair, A. Wierman, and B. Zwart, "Tail-robust scheduling via limited processor sharing," *Performance Evaluation*, 2010.

[14] J. Zhang and B. Zwart, "Steady state approximations of limited processor sharing queues in heavy traffic," *Queueing Systems. Theory and Applications*, vol. 60, no. 3-4, pp. 227–246, 2008.

[15] J. Zhang, J. G. Dai, and B. Zwart, "Law of Large Number Limits of Limited Processor-Sharing Queues," *Math. Oper. Res.*, vol. 34, no. 4, pp. 937–970, 2009.

[16] ——, "Diffusion Limits of Limited Processor-Sharing Queues," *Ann. Appl. Probab.*, vol. 21, no. 2, pp. 745–799, 2011.

[17] S. Asmussen, *Applied probability and queues*, 2nd ed., ser. Applications of Mathematics (New York). New York: Springer-Verlag, 2003, vol. 51.

[18] W. Walter, *Ordinary differential equations*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1998, vol. 182.

[19] R. J. Williams, "Diffusion approximations for open multiclass queueing networks: sufficient conditions involving state space collapse," *Queueing Systems Theory Appl.*, vol. 30, no. 1-2, pp. 27–88, 1998.

[20] M. Bramson, "State space collapse with application to heavy traffic limits for multiclass queueing networks," *Queueing Systems Theory Appl.*, vol. 30, no. 1-2, pp. 89–148, 1998.

[21] S. N. Ethier and T. G. Kurtz, *Markov processes*, ser. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. New York: John Wiley & Sons Inc., 1986.

APPENDIX

In this appendix, we collect several technical proofs of results stated in Sections III and IV.

Proof of Proposition 3: Due to space considerations, we only give a sketch of the argument. The continuity is clear by the definition. It is also clear that $\mathcal{L}(x) = 0$ if and only if $x \in \mathcal{I}$. We now focus on the derivative. Since for any solution $\bar{X}$ to the ODE (2), Proposition 1 yields that the workload load $\bar{W}$ does not change, for $w = \bar{W}(0)$ we have

$$\begin{aligned}\frac{d}{dt}\mathcal{L}(\bar{X}(t)) &= 2\left(\bar{X}(t) - x^\dagger(w)\right)^T (I - P^T)^{-1}\bar{X}'(t)\\ &= 2\left(\bar{X}(t) - x^\dagger(w)\right)^T \\ &\quad (I - P^T)^{-1}\left[\lambda - (I - P^T)\mu R(\bar{X}(t))\right]\\ &= 2\left(\bar{X}(t) - x^\dagger(w)\right)^T \left[\gamma - \mu R(\bar{X}(t))\right].\end{aligned}$$

It remains to show that for any i, if $\bar{X}_i(t) > x_i^\dagger(w)$, then $\frac{\gamma_i}{\mu_i} < R_i(\bar{X}(t))$, and vice versa. Consider first the case where $w \le w^*$ and the hyperplane

$$H = \{x \in \mathbb{R}_+^J : e^T x = e^T x^\dagger(w)\}.$$

This is the hyperplane that crosses the point $x^\dagger(w)$ and on which the total number of jobs remains the same. Let $\bar{Y}(t)$ be the intersection of H and the line determined by the origin and $\bar{X}(t) \wedge K$, where the minimum is taken component-wise. It is clear that $R_i(\bar{X}(t)) = \frac{\bar{Y}_i(t)}{\sum_j \bar{Y}_j(t)}$. Since the workload hyperplane from (6) and H are defined by linear equations with positive coefficients, they have no intersection with:

$$\begin{aligned}S_+(x_i^\dagger(w)) &= \{x \in \mathbb{R}_+^J : x_i > x_i^\dagger(w) \text{ for all } i\},\\ S_-(x_i^\dagger(w)) &= \{x \in \mathbb{R}_+^J : x_i < x_i^\dagger(w) \text{ for all } i\}.\end{aligned}$$

Intuitively, both H and the workload hyperplane are almost perpendicular to the line that connects the origin and $x^\dagger(w)$. Thus, $\bar{X}_i(t) > x_i^\dagger(w)$ if and only if $\bar{Y}_i(t) > x_i^\dagger(w)$. Note here that $\bar{X}_i(t) > x_i^\dagger(w)$ is equivalent to $\bar{X}_i(t) \wedge K_i > x_i^\dagger(w)$ due to $w \le w^*$. Since both $\bar{Y}_i(t)$ and $x_i^\dagger(w)$ are on H (meaning that the total number of jobs remains the same), $\bar{Y}_i(t) > x_i^\dagger$ implies $\frac{\bar{Y}_i(t)}{\sum_j \bar{Y}_j(t)} > R_i(x^\dagger(w)) = \gamma_i/\mu_i$, where the equality is due to $x^\dagger(w)$ being on the invariant manifold. This implies that $R_i(\bar{X}(t)) > \gamma_i/\mu_i$.

The case where $w > w^*$ can be argued similarly. The only difference is that we also need to project $x^\dagger(w)$ back to the region $\{x : x_i \le K_i\}$ by taking $x^\dagger(w) \wedge K$. Essentially, we only change $x_1^\dagger(w)$ to K_1. We omit the detailed presentation. ■

The remainder of this appendix is devoted to proofs of results appearing in Section IV. Let $E_i^n(\cdot)$ and $N_i^n(\cdot)$, $i = 1, \ldots, J$, be independent Poisson processes, both with rate 1. Let $\Phi_{i,j}(n) = \sum_{l=1}^n \phi_{i,j}(l)$ be the routing process, where $\phi_{i,j}(l)$, $i = 1, 2, \ldots$ is a sequence of independent Bernoulli random variables, with $\mathbb{P}\{\phi_{i,j}(1) = 1\} = p_{i,j}$.

The dynamics of the queuing system under fluid scaling are for any $s \le t$ and $i = 1, \ldots, J$,

$$\begin{aligned}\bar{X}_i^n(t) = \bar{X}_i^n(s) &+ \frac{1}{n}[E_i^n(\lambda^n nt) - E_i^n(\lambda^n nt)]\\ &- \frac{1}{n}N_i^n\left(\mu_i \int_{ns}^{nt} R_i(\frac{1}{n}X^n(\tau))d\tau\right)\\ &+ \sum_j \frac{1}{n}\Phi_{j,i}\left(N_j^n\left(\mu_j \int_{ns}^{nt} R_j(\frac{1}{n}X^n(\tau))d\tau\right)\right).\end{aligned} \tag{17}$$

Observe that via change of variable, we have that

$$\int_{ns}^{nt} R(\frac{1}{n}X^n(\tau))d\tau = \int_s^t R(\bar{X}^n(s))d\tau.$$

Let

$$\begin{aligned}\bar{\Phi}_i^n(t) = &\sum_j \frac{1}{n}\Phi_{j,i}\left(N_j^n\left(\mu_j \int_{ns}^{nt} R_j(\frac{1}{n}X^n(\tau))d\tau\right)\right)\\ &- \sum_j p_{j,i}\frac{1}{n}N_j^n\left(\mu_j \int_{ns}^{nt} R_j(\frac{1}{n}X^n(\tau))d\tau\right),\end{aligned}$$

and $\bar{\Phi}^n$ be the vector with ith component being $\bar{\Phi}_i^n(t)$. We can center the stochastic processes (17) with their mean to obtain a representation involving martingales.

$$\begin{aligned}\bar{X}^n(t) = \bar{X}^n(s) + \bar{M}_a^n(t) - \bar{M}_a^n(s) + \bar{\Phi}^n(t) \\ - (I - P^T)^{-1}\left[\bar{M}_s^n(t) - \bar{M}_s^n(s)\right] + \lambda^n(t-s) \\ - (I - P^T)^{-1}\left(\mu \int_s^t R(\bar{X}^n(\tau))d\tau\right), \end{aligned} \tag{18}$$

where

$$\begin{aligned}\bar{M}_a^n(t) &= \frac{1}{n}E^n(\lambda^n nt) - \lambda^n t, \\ \bar{M}_s^n(t) &= \frac{1}{n}N^n\left(\mu\int_0^{nt} R(\frac{1}{n}X^n(s))ds\right) \\ &\quad - \mu\frac{1}{n}\int_0^{nt} R(\frac{1}{n}X^n(s))ds.\end{aligned}$$

Again, the martingales are of dimension J. We can also write the dynamic equation (18) for the shifted fluid scaled process $\bar{X}^{n,m}$ in exactly the same way. We now study some estimates.

Lemma 1. *Fix $T > 0$ and $L > 1$. For all $\epsilon, \epsilon' > 0$, there exists an n_0 such that whenever $n \geq n_0$,*

$$\mathbb{P}^n\Big(\max_{m\leq\lfloor nT\rfloor}\sup_{s,t\in[0,L]}|\bar{E}^{n,m}(s,t) - \lambda(t-s)| > \epsilon'\Big) < \epsilon, \tag{19}$$

$$\mathbb{P}^n\Big(\max_{m\leq\lfloor nT\rfloor}\sup_{s,t\in[0,L]}|\bar{N}^{n,m}(s,t) - \mu(t-s)| > \epsilon'\Big) < \epsilon. \tag{20}$$

Proof: The result stated in (19) is the same as Lemma 4.1 in [16]. The result in (20) follows straightforwardly from by specializing the renewal process to a Poisson process. ■

Proof of Proposition 5: The relative compactness follows immediately from Lemma 1 and the dynamic equation (18). To see that the limiting process is a fluid model solution, note that both $\bar{\Phi}^n$ and the martingale terms in (18) converges to 0 along any subsequence following classical arguments. ■

A technical disclaimer: when convenient, we will assume, using Skorohod's representation theorem [21, Theorem 3.1.8], that the weak convergence in Proposition 5 takes place almost surely on a suitable probability space, and also assume that all our processes are actually defined on that space without changing notation.

The task of proving state-space collapse is divided into the following steps: we first show that the set of limits in Proposition 5, which called fluid limits, is "rich" in the sense that itself and the set of shifted fluid scaled process mutually approximates each other (inequality (21)); the proof of the state-space collapse result is then presented based on the richness of fluid limits and the convergence to the invariant manifold for fluid model.

Let $\mathcal{L}_{T,L}$ denote the set of fluid limits of all convergent subsequences of sequences in Proposition 5. Pick an element $\tilde{X}(\cdot) \in \mathcal{L}_{T,L}$; for any $\epsilon > 0$ and $n_0 \in \mathbb{N}_+$, there exists an $n \geq n_0$, $m \leq \lfloor nT\rfloor$ such that

$$\left|\bar{X}^{n,m}(\cdot) - \tilde{X}(\cdot)\right| \leq \epsilon. \tag{21}$$

Roughly speaking, any element in $\mathcal{L}_{T,L}$ can be approximated by a shifted fluid-scaled process of the nth system.

Following the same proof as for Lemma 5.3 in [16], we have that any shifted fluid-scaled process of the nth system with index n large enough can be approximated by some element in $\mathcal{L}_{T,L}$. Mathematically, for each $\epsilon > 0$, there exists an $n_0 \in \mathbb{R}_+$ such that for any $n \geq n_0$ and $m \leq \lfloor nT\rfloor$, we can find a $\tilde{X}(\cdot) \in \mathcal{L}_{T,L}$ satisfying (21).

Proof of Proposition 6: The proof of this result follows using ideas similar to the proof of Theorem 2.2 in [16], which follows a classical approach introduced in Bramson [20]. Due space limitations, we only give a sketch of the proof.

From Proposition 5, we have that each $\tilde{X} \in \mathcal{L}_{T,L}$ is a fluid model solution. By Proposition 4, there exists an $L^* > 0$ such that when $s > L^*$, $|\tilde{X}(s) - x^\dagger(w)| < \epsilon/6$, where $w = \tilde{W}(s)$. Here $\tilde{W}$ is the fluid workload for $\tilde{X}$, defined in the same way as (4). Since $x^\dagger(w)$ is on the invariant manifold, we have that $x^\dagger(w) = \Delta(w)$. Thus

$$\left|\tilde{X}(s) - \Delta(\tilde{W}(s))\right| < \epsilon/3. \tag{22}$$

Now, fix a constant $L > L^* + 1$. Note that

$$[0, n^2T] \subset [0, nL^*] \cup \bigcup_{m=0}^{\lfloor nT\rfloor}[n(m+L^*), n(m+L)].$$

By the definition of diffusion and shifted fluid scaling, to show (14) it suffices to show that for all large enough n,

$$\max_{m\leq\lfloor nT\rfloor}\sup_{s\in[L^*,L]}\left|\bar{X}^{n,m}(s) - \Delta(\bar{W}^{n,m}(s))\right| < \epsilon, \tag{23}$$

$$\sup_{s\in[0,L^*]}\left|\bar{X}^{n,0}(s) - \Delta(\bar{W}^{n,0}(s))\right| < \epsilon. \tag{24}$$

Again, here we utilize the convenience resulting from Skorohod's representation theorem. We first prove (23). Fix an $m \leq \lfloor nT\rfloor$. Then, from the fluid approximation, there exists $\tilde{X} \in \mathcal{L}_{T,L}$ such that

$$\sum_{s\in[0,L]}\left|\bar{X}^{n,m}(s) - \tilde{X}(s)\right| \leq \epsilon/3. \tag{25}$$

Following the classical argument as in [20], we can also show that the workload process converges, thus we have that the shifted fluid scaled workload process $\bar{W}^{n,m}$ converges to the workload fluid limit $\tilde{W}$. It is clear to see that Δ is a Lipschitz continuous function, so we have

$$\sum_{s\in[0,L]}\left|\Delta(\bar{W}^{n,m}(s)) - \Delta(\tilde{W}(s))\right| \leq \epsilon/3, \tag{26}$$

for large n. Then (23) follows from the triangle inequality, (22), (25) and (26). The proof for (24) follows in a similar fashion but is slightly easier by utilizing the initial condition. ■