The Trouble with Diversity: Fork-Join Networks with Heterogeneous Customer Population

Vien Nguyen

OR 268-92 October 1992

The Trouble with Diversity: Fork-Join Networks with Heterogeneous Customer Population

Viên Nguyen
Sloan School of Management, M.I.T., Cambridge, MA 02139

Abstract

Consider a feedforward network of single-server stations populated by multiple job types. Each job requires the completion of a number of tasks whose order of execution is determined by a set of deterministic precedence constraints. The precedence requirements allow some tasks to be done in parallel (in which case tasks would "fork") and require that others be processed sequentially (where tasks may "join"). Jobs of a given type share the same precedence constraints, interarrival time distributions, and service time distributions, but these characteristics may vary across different job types. We show that the heavy traffic limit of certain processes associated with heterogeneous fork-join networks can be expressed as a semimartingale reflected Brownian motion with polyhedral state space. The polyhedral region typically has many more faces than its dimension, and the description of the state space becomes quite complicated in this setting. One can interpret the proliferation of additional faces in heterogeneous fork-join networks as (i) articulations of the fork and join constraints, and (ii) results of the *disordering effects* that occur when jobs fork and join in their sojourns through the network.

KEYWORDS: fork-join networks, heterogeneous customer populations, reflected Brownian motion, non-simple polyhedral state space, diffusion approximations, heavy traffic analysis.

Contents:

October, 1992

1 Introduction and Summary

We consider in this paper the class of feedforward fork-join networks with heterogeneous customer populations. The network, which consists of d single-server stations, is populated by multiple job types. Each job requires the completion of a number of tasks whose order of execution is determined by a set of deterministic precedence constraints. The precedence requirements allow some tasks to be performed in parallel (in which case tasks would *fork*) and require that others be processed sequentially (where tasks may *join*). Jobs of a given type share the same precedence constraints, interarrival time distributions, and service time distributions, but these characteristics may vary across different job types. We restrict attention to the case where the network is *feedforward*: that is, stations are numbered in such a way that jobs always flow from lower numbered stations to higher numbered ones.

We present a heavy traffic analysis for processing networks of the type described above. It was shown in Nguyen [18] that when the customer population is *homogeneous* — that is, when all customers share the same precedence requirements, interarrival time distributions, and service time distributions — the heavy traffic behavior of the network can be approximated by a d-dimensional semimartingale reflected Brownian motion (SRBM) whose state space is a non-simple convex polyhedral cone in the nonnegative orthant. Unlike the corresponding results for conventional queueing networks (networks with strictly sequential processing) [11, 12, 13, 19, 20], the number of faces in the polyhedral region is greater than d. One can interpret the presence of additional faces as articulations of synchronization constraints embodied in the fork and join constructs.

In this paper, we show that the heavy traffic limit of certain processes associated with *heterogeneous* fork-join networks can also be expressed as d-dimensional SRBM's with polyhedral state space. However, the polyhedral region typically has many more faces than its homogeneous counterpart, and the description of the state space becomes vastly more complicated in this setting. This result is surprising when compared to those associated with conventional queueing networks, where the form of the limiting process does not change with the presence of multiple customer types (this is a result of the "state-space collapse" phenomenon) [19, 21]. One can interpret the proliferation of additional faces in heterogeneous fork-join networks as results of the *disordering effects* that occur when jobs fork and join in their sojourns through the network.

Processing systems that are characterized by parallel as well as sequential processing exist in many industrial settings. Readers may refer to Baccelli and Makowski [5], Avi-Itzhak [16], and Nguyen [18] for a survey of several interesting applications. The generalization of [18] to allow multiple job types constitutes an important extension from the practical point of view. Most current treatises of fork-join networks assume that all customers are statistically similar [5]. Baccelli and Liu [4] consider a fork-join network in which a job may send *batches* of tasks (that may include more than one task) to processing stations, and jobs of different types send batches of different sizes. Baccelli and Liu still assume, however, that all jobs share the same feedforward deterministic

routing structure. Such a model can represent, for example, systems in which some processing stations are capable of performing more than one kind of task; Baccelli and Liu are motivated by multiprocessor systems running parallel programs.

The recent works by Adler, Mandelbaum, Nguyen, and Schwerer [1, 2, 3] propose a processing network model for studying new product development. The model they describe, which they simply call a "processing network model," is more encompassing than the class of fork-join networks studied here. The key restriction in this paper, which is not assumed in [1, 2, 3], is that jobs must visit workstations in a *feedforward* manner. The possibility of feedback in the network is yet an important generalization that must be considered in future work. However, as the work by Adler, Mandelbaum, Nguyen, and Schwerer demonstrates, heterogeneity in the customer population is an essential characteristic that must be captured.

The paper is organized as follows. We give a formal description of the model in Section 2 and define the processes of interest Section 3. In order to state the heavy traffic limit results for these processes, one must refer to a "sequence of systems." Section 4 defines such a sequence. Before stating the heavy traffic limit theorems, we introduce some additional notation and preliminary results in Section 5. The main theorems are then summarized in Section 6, where we also illustrate the heavy traffic limit theorem for several special cases. The proof of these theorems are then given in Section 7. Finally, some concluding remarks are given in Section 8.

We end this section with some technical preliminaries. The space $\mathbf{D}^r[0,\infty)$ is the r-dimensional product space of functions $f : [0,\infty) \to \Re^r$ that are right continuous on $[0,\infty)$ and have left limits on $(0,\infty)$. The space $\mathbf{D}^r[0,\infty)$ is endowed with the Skorohod topology [6]. For X^n a sequence of processes in $\mathbf{D}^r[0,\infty)$ and $X \in \mathbf{D}^r[0,\infty)$, we write $X^n \Longrightarrow X$ to mean X^n converges to X in distribution.

For $f : [0,\infty) \to \Re$, set

$$||f||_t \equiv \sup_{0\le s\le t} |f(s)|,$$

and for a vector-valued function $f = (f_1, f_2, \ldots, f_r)' : [0,\infty) \to \Re^r$, we let

$$||f||_t \equiv (||f_1||_t, \ldots, ||f_r||_t)'.$$

A sequence of functions $\{f^n\}$ converges to a function f uniformly on compact sets (u.o.c.) if for each $t \ge 0$, $||f^n - f||_t \to 0$ as $n \to \infty$. For a sequence of functions $\{X^n\}$ on $\mathbf{D}^r[0,\infty)$ and X a process in $\mathbf{D}^r[0,\infty)$, we write $X^n \to X$ u.o.c if almost surely, X^n converges to X uniformly on compact sets.

In our heavy traffic limit theorems, the weak limit obtained is a semimartingale reflected Brownian motion whose state space is a polyhedral cone in the nonnegative orthant. A Brownian motion process having drift vector θ and covariance matrix Γ will be denoted as (θ, Γ)BM; likewise, a semimartingale reflected Brownian motion with these drift and covariance parameters, reflection matrix R, and state space is $\mathcal{S}$ is denoted as $(\mathcal{S}, \theta, \Gamma, R)$SRBM.

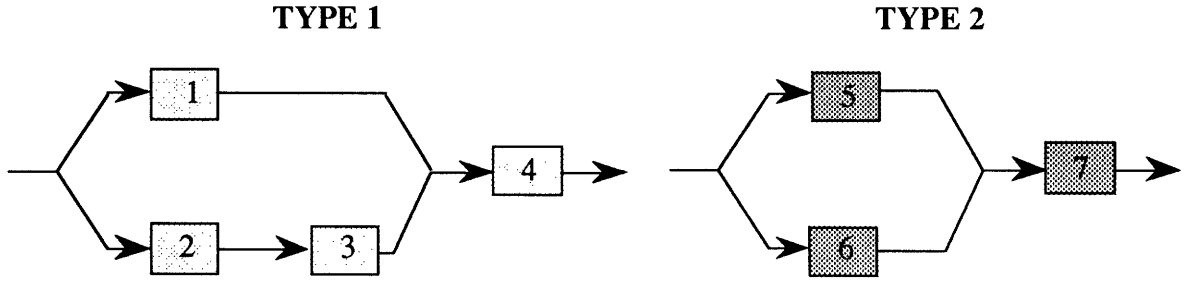

Figure 1: Tasks and Precedence Constraints of Two Job Types

2 Model Description

The network under consideration consists of d single-server stations and hosts p types of jobs. Jobs of type q arrive to the network at rate α_q. Each type q job requires completion of a number of activities. Hereafter, we refer to a job-type/activity pair as a *task* or a *class* interchangeably. Task k receives service from station $j = s(k)$, and we denote by τ_k the mean service time for task k. Letting $\mathcal{A}_q$ denote the set of tasks (or classes) in job type q, set

$$\mathcal{A} \equiv \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_p = \{1, \ldots, K\}.$$

Our convention will be to index workstations by $i, j = 1, \ldots, d$, job types by $q, r = 1, \ldots, p$, and tasks by $k, l = 1, \ldots, K$. For notational convenience, we define $\lambda_k = \alpha_q$ for all tasks $k \in \mathcal{A}_q$ and we write $q(k)$ to mean the job type q for which $k \in \mathcal{A}_q$.

The order in which tasks are executed is determined by a set of deterministic precedence constraints, which are articulated by way of a $K \times K$ precedence matrix $\mathsf{P} = (\mathsf{P}_{kl})$ defined as follows:

$$\mathsf{P}_{kl} \equiv \begin{cases} 1 & \text{if task } k \text{ is an immediate predecessor for task } l \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

(Because all elements of the precedence matrix P are zeros and ones, routing of tasks is clearly deterministic within each job type.) We assume that there exists a column and row permutation of P such that the resulting matrix is strictly upper triangular. In terms of the model, this means that each task is performed exactlly once and is never repeated. From the precedence matrix P, we can now define the set of immediate predecessors as

$$\mathcal{P}(l) \equiv \{k \in \mathcal{A} : \mathsf{P}_{kl} = 1\}. \tag{2.2}$$

From the modelling point of view, $\mathcal{P}(l)$ is the set of tasks that must be completed before task l can begin.

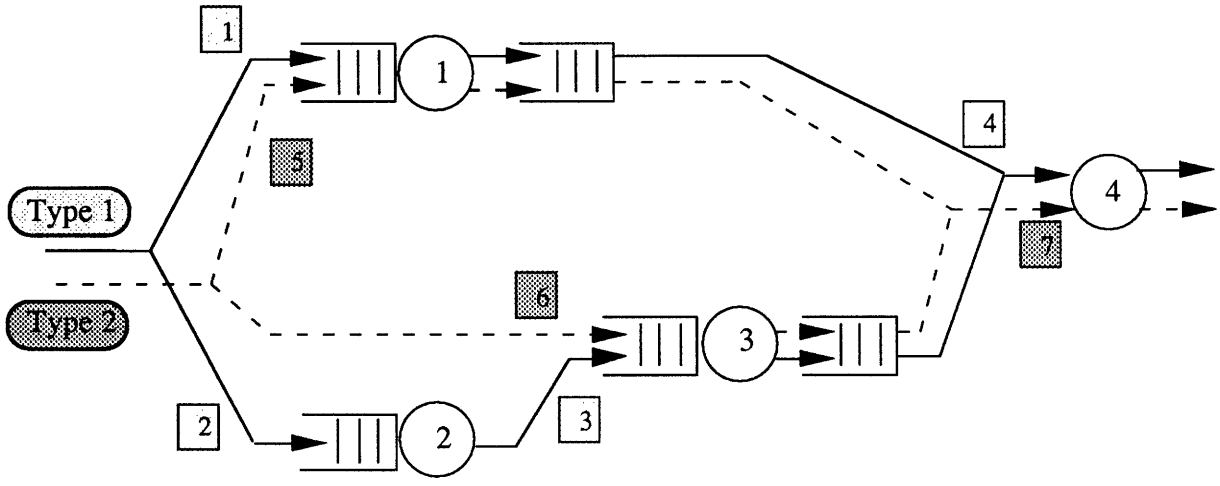

Figure 2: Tasks and Precedence Constraints of Two Job Types

For the two job types depicted in Figure 1, $\mathcal{A}_1 = \{1,2,3,4\}$, $\mathcal{A}_2 = \{5,6,7\}$, $\mathcal{P}(4) = \{1,3\}$, and the precedence matrix P is given by

$$\mathsf{P} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We allow tasks to map to stations in a many-one-to fashion, implying that while each station may be capable of performing several types of tasks, each task is performed at exactly one station. We define the *constitutency* of station $i = 1 \ldots, d$ as the set of tasks that are served at station i:

$$\mathcal{C}(i) = \{k \in \mathcal{A} : s(k) = i\}. \tag{2.3}$$

We write $c(i)$ to mean the cardinality of the constituency set $\mathcal{C}(i)$. Next, define the predecessor and successor station mappings for station $j = 1 \ldots, J$ as follows:

$$\pi(j) \equiv \{i = s(k) : k \in \mathcal{P}(l), l \in \mathcal{C}(j)\}, \tag{2.4}$$

$$\sigma(j) \equiv \{i = s(k) : l \in \mathcal{P}(k), l \in \mathcal{C}(j)\}. \tag{2.5}$$

That is, $\pi(j)$ denotes the set of stations whose output feeds directly into station j; analogously, $\sigma(j)$ is the set of stations that receive input from station j. In conjuction with the predecessor and

successor station mappings, we now define a $d \times d$ "routing" matrix $\mathsf{IP} = \mathsf{IP}_{ij}$ whose elements are given by

$$\mathsf{IP}_{ij} \equiv \begin{cases} 1 & \text{if } i \in \pi(j) \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

We assume that there is a column and row permutation of IP such that the resulting matrix is strictly upper triangular; in terms of the model, this means that we assume jobs traverse the network in such a way that jobs never return to a station it previously visited. In addition, this condition implies that there are no precedence constraints among the tasks at each station. This constitutes the *feedforward* routing assumption stated in Section 1.

It will be useful to think of new arrivals to the network as originating from a "dummy" station 0. With that interpretation in mind let us define for each job type q

$$\mathcal{A}_q^0 = \{k \in \mathcal{A}_q : P_{lk} = 0 \text{ for all } l \in \mathcal{A}_q\} \tag{2.7}$$

$$\mathcal{A}_q^e = \{k \in \mathcal{A}_q : P_{kl} = 0 \text{ for all } l \in \mathcal{A}_q\}. \tag{2.8}$$

Clearly, processing of a type q job begins with those tasks $k \in \mathcal{A}_q^0$ and ends with the tasks $k \in \mathcal{A}_q^e$. Next, set $\mathcal{P}(k) = 0$ if $\mathcal{P}(k) = \emptyset$ (or equivalently if $k \in \mathcal{A}_{q(k)}^0$), let $s(0) = 0$, and redefine (2.4)-(2.5) accordingly. Finally, we define

$$\sigma(0) = \{i : \pi(i) = \{0\}\}.$$

Thus, the stations in $\sigma(0)$ receive only external arrivals and the feedforward assumption guarantees that $\sigma(0) \neq \emptyset$. Figure 2 shows how the two job types depicted in Figure 1 are processed in a network consisting of four workstations. For this example, we have $\pi(1) = \pi(2) = \{0\}$, $\pi(3) = \{0, 2\}$, $\pi(4) = \{1, 3\}$, and $\sigma(0) = \{1, 2\}$.

A node j is said to be a fork node if it contains a task $k \in \mathcal{C}(j)$ such that $k \in \mathcal{P}(l)$ for more than one task l . Similarly, station j is join node if $\mathcal{P}(k)$ contains more than one element for some constituent task $k \in \mathcal{C}(j)$. At a join node, a type k task is said to be complete, or a unit, if all of its predecessors $l \in \mathcal{P}(k)$ have completed their service.

We assume that tasks are served at each station in a FIFO manner. At nodes that do not involve a joining of tasks, this simply means that the tasks are served in the order of their arrival to the station. At join nodes, the arrival time of a (complete) task is defined to be the time at which its *last* predecessor completes service. Note that such a service discipline considers only local, or station-level, information. For the special case of fork-join networks with homogeneous customers, we argued in [18] that such a policy is equivalent to the global scheme of serving tasks in the order of the arrival of their associated *jobs*. In this setting, where different customer types traverse different routes through the network, one observes a fundamentally different phenomenon. In particular, a task corresponding to a later arrival may enter a downstream station *before* those tasks associated with earlier jobs. For example, consider the scenario depicted in Figure 3, which shows the status of five jobs in their intermediate stages of processing. The network contains three

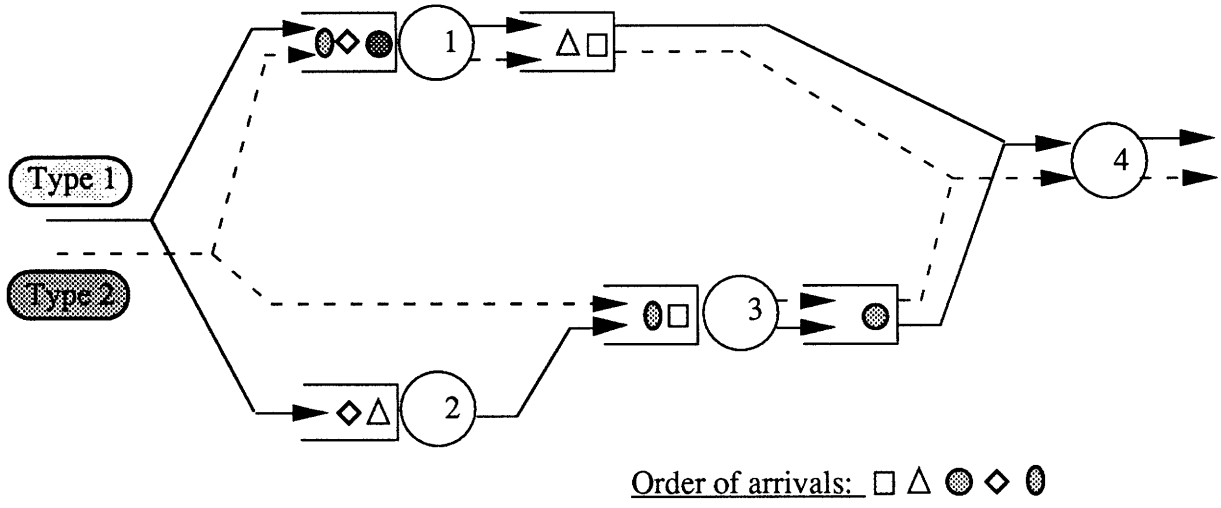

Figure 3: A Fork-Join Network with Jobs in Intermediate Stages of Processing

type 1 jobs and two type 2 jobs, which arrived to the network in the order $1, 1, 2, 1, 2$. Each job type follows the routing requirements described in Figure 2. We ask the reader to focus attention on the buffer between stations 3 and 4, hereafter referred to as buffer $(3, 4)$. Note that the first type 2 job (the third job to arrive to the network) has overtaken the first two arrivals and is the first job to reach buffer $(3, 4)$. Moreover, if server 1 completes his next task before server 3 can finish his, the first job to be processed by server 4 will be the type 2 job. The policy of serving tasks in the order of their arrival to a station may therefore result in serving jobs out of order. Moreover, a downstream station may incur delays due to the need for resequencing tasks (for example, if the station is a join node) that were overtaken by other tasks at upstream stations (for example, due to forks). In Figure 3, for instance, server 4 must remain idle even though each of his incident buffers is nonempty. In this paper we investigate how such a disordering is manifested in the heavy traffic limit.

3 Representations for Processes of Interest

To construct the basic stochastic processes associated with the fork-join network, let us assume a probability space $(\Omega, \mathcal{F}, P)$ on which are defined sequences of unitized random variables $\{u_q(i), i \geq 1\}$ and $\{v_k(i), i \geq 1\}$, $q = 1, \ldots, p$, $k = 1, \ldots, K$, where $u_q(i)$ and $v_k(i)$ are strictly positive with unit mean. As will be explained in the next section, we require very weak assumptions regarding the joint distribution of these sequences of unitized variables. However, readers may find it helpful to think in terms of the concrete case where each is a sequence of i.i.d. random variables and

the sequences are mutually independent. From these sequences, the interarrival times and service times for the network are constructed by setting the interarrival time for the i^{th} job of type q to be $\alpha_q^{-1}u_q(i)$, and the service time for task k of this job to be $\tau_k v_k(i)$. Recall that α_q is the average arrival rate for new type q jobs and τ_k is the mean service time for task k. Also recall that $\lambda_k = \alpha_q$ for each $k \in \mathcal{A}_q$.

To construct the external arrival process for type q jobs, set $u_q(0) \equiv 0$ and define

$$N_q(t) \equiv \max\{m : \sum_{i=0}^{m} \alpha_q^{-1} u_q(i) \leq t\}.$$

For $k = 1, \ldots K$, let $V_k(t)$ be the partial sums process associated with the service times for tasks k,

$$V_k(t) \equiv \sum_{i=1}^{[t]} \tau_k v_k(i).$$

Next, define for each $k \in \mathcal{C}(j)$ and $j = 1, \ldots, d$,

$$L_{jk}(t) \equiv V_k(N_{q(k)}(t)) = \tau_k v_k(1) + \cdots + \tau_k v_k(N_{q(k)}(t)), \qquad k \in \mathcal{C}(j). \tag{3.1}$$

The process $L_{jk}(t)$ is called the *class k total workload input process* for station j; it is the sum of all task k service times associated with those jobs that enter the network during $[0,t]$. Note that $L_{jk}(t)$ includes service times corresponding to tasks that may not arrive to station j until after time t. Set

$$\xi_j(t) = \left(\sum_{k \in \mathcal{C}(j)} L_{jk}(t) \right) - t; \tag{3.2}$$

because t is the *potential* amount of work that can be processed in t units of time, $\xi_j(t)$ is the difference between the workload input and the potential workload output, and for this reason it is called the *total workload netflow process* at station j.

Let us choose an "external" station $j \in \sigma(0)$, fixing j until further notice. For each $k \in \mathcal{C}(j)$, set $A_{jk}(t) \equiv N_{q(k)}(t)$. Next let $M_{jk}(t) \equiv L_{jk}(t)$ and $X_j(t) \equiv \xi_j(t)$. Because station j hosts only external arrivals, $A_{jk}(t)$ is the number of task k that has actually arrived to station j by time t. Similarly $M_{jk}(t)$ is the amount of task k work that has arrived to station j in $[0,t]$ and $X_j(t)$ is the corresponding *immediate* workload netflow process at this station.

From Section 2.2 of [10], we can verify that the processes W_j and I_j are uniquely defined by the following three statements:

$$W_j(t) = X_j(t) + I_j(t) \geq 0 \quad \text{for all } t \geq 0; \tag{3.3}$$

$$I_j(\cdot) \text{ is continuous and nondecreasing with } I_j(0) = 0; \tag{3.4}$$

$$I_j(\cdot) \text{ increases only at times } t \text{ when } W_j(t) = 0; \tag{3.5}$$

moreover, I_j is given by the *continuous* mapping

$$I_j(t) = -\inf_{0 \le s \le t} \{X_j(s)\}^-. \tag{3.6}$$

One interprets I_j as the *cumulative idleness process* for server j and W_j as the *immediate workload process* at station j. That is, $W_j(t)$ corresponds to the sum of the impending service times for (complete) tasks waiting at station j at time t, plus the remaining service time of any task that may be in service. If we define $Z_j(t)$ to be the *total* amount of work for server j that is present *anywhere* in the system at time t, then

$$Z_j(t) = \xi_j(t) + I_j(t), \tag{3.7}$$

and $Z_j(t) = W_j(t)$ as a consequence of $j \in \sigma(0)$. Hereafter we refer to $Z_j(t)$ as the *total workload process* for server j.

Next, let $\eta_j(t)$ be the arrival time of the customer in service at station j at time t if $W_j(t) > 0$, and set $\eta_j(t) = t$ otherwise. letting $Y_{jk}(t)$ be the amount of time server j has spent serving tasks k in $[0, t]$, it follows from the FIFO service discipline at each station that

$$Y_{jk}(t) = M_{jk}(\eta_j(t)) + \epsilon_{1k}(t), \tag{3.8}$$

where $\epsilon_{1k}(t)$ is the amount of service the current task has received if that task is of class k and $\epsilon_{1k}(t) = 0$ otherwise. As a matter of definition, $\eta_j(t)$ is bounded by the immediate workload process as follows:

$$W_j(\eta_j(t)) \le t - \eta_j(t) \le W_j(\eta_j(t)) + \epsilon_{2j}(t), \tag{3.9}$$

where $\epsilon_{2j}(t) = 0$ if $W_j(\eta_j(t)) = 0$ and otherwise $\epsilon_{2j}(t)$ is the service time of the customer currently occupying server j. Letting $S_k = \{S_k(t), t \ge 0\}$ be the renewal process associated with the task k service times $\{\tau_k v_k(1), \tau_k v_k(2), \ldots\}$, the number of tasks k to have departed from station j by time t, denoted as $D_{jk}(t)$, is then given by

$$D_{jk}(t) = S_k(Y_{jk}(t)). \tag{3.10}$$

Finally, defining $U_{jk}(t)$ to be the the total amount of *partial* work associated with tasks k that is present *anywhere* in the system at time t, it follows from the previous definitions that

$$U_{jk}(t) = L_{jk}(t) - Y_{jk}(t). \tag{3.11}$$

Moreover, it is a trivial consequence that $Z_j(t) = \sum_{k \in \mathcal{C}(j)} U_{jk}(t)$.

In an inductive manner, these definitions can be extended to all stations in the network. Consider a station j such that all immediate predecessor stations have been "treated", that is, if $i \in \pi(j)$ then for each $l \in \mathcal{C}(i)$ the processes $X_i(t)$, $W_i(t)$, $I_i(t)$, $Z_i(t)$, $D_{il}(t)$, and $U_{il}(t)$ have been defined. For each task $k \in \mathcal{C}(j)$ one defines its arrival process to be:

$$A_{jk}(t) \equiv \begin{cases} N_{q(k)}(t) & \text{if } k \in \mathcal{A}^0_{q(k)}, \\ \min_{l \in \mathcal{P}(k)} D_{s(l)l}(t) & \text{otherwise.} \end{cases} \tag{3.12}$$

One can interpret $A_{jk}(t)$ as the number of *complete* tasks k that have arrived to station j by time t. (We take the convention that work is associated with complete tasks, so incomplete tasks present no work to the server.) The immediate workload input process and immediate netflow process for station j are defined, respectively, via

$$M_{jk}(t) \equiv V_k(A_{jk}(t)) = \tau_k \left[v_k(1) + \cdots + v_k(A_{jk}(t))\right] \tag{3.13}$$

and

$$X_j(t) \equiv \left(\sum_{k \in \mathcal{C}(j)} M_{jk}(t) \right) - t. \tag{3.14}$$

The workload process W_j, the idleness process I_j, the total workload process Z_j, the departure process D_{jk}, and the partial workload process U_{jk} are then defined exactly as in (3.3)-(3.11). The vector processes N, V, L, A, M, X, W, I, Z, D, and U are then defined in the obvious manner.

The throughput time of a job is the length of time between the job's arrival and its subsequent departure from the system. Let $T_q(t)$ be the throughput time of the next type q job to enter the network after time t. The intermediate process $T_{qk}(t)$, $k \in \mathcal{A}_q$, is defined to be the "throughput time through task k," which is the amount of elapsed time until task k is completed. As a matter of definition, we have the relationship

$$T_q(t) \equiv \max_{k \in \mathcal{A}_q^e} \{T_{qk}(t)\}.$$

To define the intermediate processes $T_{qk}(t)$, we first define for each job type q the process

$$\Phi_q(t) \equiv \alpha_q^{-1} u(1) + \cdots + \alpha_q^{-1} u(N_q(t)) + \alpha_q^{-1} u(N_q(t) + 1),$$

interpreted as the arrival epoch of the next type q job to enter the network after time t. For each task $k \in \mathcal{A}_q^0$, let

$$\begin{aligned} \Phi_{qk}(t) &\equiv \Phi_q(t) \quad \text{and} \\ T_{qk}(t) &\equiv W_{s(k)}(\Phi_{qk}(t)). \end{aligned}$$

Because a type q job begins immediately with tasks $k \in \mathcal{A}_q^0$, $\Phi_{qk}(t)$ is the arrival time of this task to station $s(k)$. Furthermore, because tasks are served in a first-in-first-out manner, the amount of time this task must spend at station $s(k)$ is precisely the amount of work found at station $s(k)$ immediately after its arrival (which includes the service time associated with the new arrival). Thus $T_{qk}(t)$ is the total sojourn time of the job through task k.

For other stations in the network, the random processes $\Phi_{qk}(t)$ and $T_{qk}(t)$ are inductively defined as follows. Suppose that k is a task such that $T_{ql}(t)$ has been defined for each $l \in \mathcal{P}(k)$, and set

$$\Phi_{qk}(t) \equiv \Phi_q(t) + \max_{l \in \mathcal{P}(k)} T_{ql}(t) \tag{3.15}$$

$$T_{qk}(t) \equiv \max_{l \in \mathcal{P}(k)} T_{ql}(t) + W_{s(k)}(\Phi_{qk}(t)). \tag{3.16}$$

Recall that the arrival time of a task is the time at which its last precedessor task is completed. (If task k requires a join, there could be a gap between completion times of its multiple predecessor tasks.) Thus, $\max_{l\in\mathcal{P}(k)} T_{ql}(t)$ is the amount of time that elapses until task k "arrives" at station $s(k)$, $\Phi_{qk}(t)$ is precisely its time of arrival, and $T_{qk}(t)$ is the throughput time through task k.

4 A Sequence of Systems in Heavy Traffic

The limit theorems stated here apply to systems that satisfy conditions of "heavy traffic." For $k \in \mathcal{C}(j)$, let $\rho_{jk} = \lambda_k \tau_k$ be the workload factor at station j associated with tasks k, and define the total traffic intensity at station j to be

$$\rho_j \equiv \sum_{k\in\mathcal{C}(j)} \rho_{jk} = \sum_{k\in\mathcal{C}(j)} \lambda_k \tau_k. \tag{4.1}$$

The system is said to be *stable* if $\rho_j < 1$ for $j = 1, \ldots, J$, and it is said to be in *heavy traffic* if ρ_j is "approximately" 1 for each j. The precise formulation of our heavy traffic limit theorem requires the construction of a "sequence of systems," indexed by n, whose corresponding traffic intensities $\rho_j^{(n)}$ converge to 1 for all j.

Recall that the interarrival times and service times for the network are defined in terms of the basic sequences of unitized random variables $\{u_q(i) : i \geq 1\}$, $\{v_k(i) : i \geq 1\}$, $q = 1, \ldots, p$, $k = 1, \ldots, K$. To construct a sequence of fork-join networks we further require sequences of positive constants $\{\alpha_q^{(n)}, n \geq 1\}$, $\{\tau_k^{(n)}, n \geq 1\}$, $q = 1, \ldots, p$, $k = 1, \ldots, K$. In the n^{th} system of the sequence, the interarrival times and service times are taken to be $u_q^{(n)}(i) \equiv u_q(i)/\alpha_q^{(n)}$ and $v_k^{(n)}(i) \equiv \tau_k^{(n)} v_k(i)$, respectively. For the n^{th} system, $\alpha_q^{(n)}$ is the arrival rate of type q jobs and $\tau_k^{(n)}$ is the mean service time for task k. Setting $\lambda_k^{(n)} = \alpha_q^{(n)}$ for $k \in \mathcal{A}_q$, define the traffic intensities $\rho_j^{(n)}$ as in (4.1) using $\lambda_k^{(n)}$ and $\tau_k^{(n)}$ in place of λ_k and τ_j.

The convention here is to denote a parameter or a process associated with the n^{th} system by the superscript "(n)". For example, $N_q^{(n)}$ refers to the external arrival process for type q jobs in the n^{th} system. Define the centered processes

$$\begin{aligned}
\hat{N}_q^{(n)}(t) &\equiv N_q^{(n)}(t) - \alpha_q^{(n)} t & \qquad \hat{V}_k^{(n)}(t) &\equiv V_k^{(n)}(t) - \tau_k^{(n)} t \\
\hat{A}_{jk}^{(n)}(t) &\equiv A_{jk}^{(n)}(t) - \lambda_k^{(n)} t & \qquad \hat{L}_{jk}^{(n)}(t) &\equiv L_{jk}^{(n)}(t) - \rho_{jk}^{(n)} t \\
\hat{S}_k^{(n)}(t) &\equiv S_k^{(n)}(t) - (\tau_k^{(n)})^{-1} t & \qquad \hat{M}_{jk}^{(n)}(t) &\equiv M_{jk}^{(n)}(t) - \rho_{jk}^{(n)} t.
\end{aligned}$$

The results in this paper apply to processes that have been "scaled." Let $X^{(n)}$ denote a "generic" process associated with the n^{th} system. The scaled version of the process $X^{(n)}$, denoted as X^n, is defined via

$$X^n(t) \equiv n^{-1/2} X^{(n)}(nt).$$

Hereafter, when we say a "scaled" process and write the process with a superscript "n", we mean a process whose space and time dimensions have been scaled in the manner specified above.

It is assumed that the following conditions hold for the input processes of the network. First, the arrival rates and mean service times converge to finite constants, $\lambda_k^{(n)} \to \lambda_k$ and $\tau_k^{(n)} \to \tau_k$, $k = 1, \ldots, K$. This implies that $\rho_j^{(n)} \to \rho_j = \sum_{k \in \mathcal{C}(j)} \lambda_k \tau_k$. Furthermore, it is assumed that there exists a d-vector $\theta = (\theta_1, \ldots, \theta_d)$ such that for each $j = 1, \ldots, d$, $-\infty < \theta_j < \infty$ and

$$n^{1/2}(\rho_j^{(n)} - 1) \longrightarrow \theta_j \quad \text{as } n \to \infty. \tag{4.2}$$

Condition (4.2) is called the *heavy traffic condition*. It requires not only that $\rho_j = 1$ at each station, but also that the rate of convergence is "sufficiently fast" and is uniform for all stations. Finally, it is assumed that there is a $d \times d$ covariance matrix Ω such that the following functional central limit theorem holds as $n \to \infty$:

$$\begin{aligned} &(\hat{N}^n, \hat{V}^n, \hat{L}^n) \Longrightarrow (N^*, V^*, L^*), \text{where } L^* \text{ is a } (0, \Omega) \text{ Brownian motion} \\ &\text{and } N^*, V^* \text{ are also Brownian motions with zero drift.} \end{aligned} \tag{4.3}$$

To explore the implications and restrictions of assumption (4.3), write the scaled netflow process (3.2) as

$$\xi_j^n(t) = \sum_{k \in \mathcal{C}(i)} L_{jk}^n(t) + n^{1/2}(\rho_j^{(n)} - 1)t.$$

It follows from (3.1)-(3.2) and assumptions (4.2), (4.3) that

$$L_{jk}^*(t) = \hat{V}_k^*(\lambda_k t) + \tau_k N_{q(k)}^*(t), \quad \text{and} \tag{4.4}$$

$$\xi_j^*(t) = \sum_{k \in \mathcal{C}(j)} L_{jk}^*(t) + \theta_j t. \tag{4.5}$$

Defining the $d \times c$ *constituency matrix* C with elements

$$C_{ik} \equiv \begin{cases} 1 & \text{if } k \in \mathcal{C}(i) \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

and setting

$$\Gamma = C\Omega C', \tag{4.7}$$

one can conclude that

$$(\hat{N}^n, \hat{V}^n, \hat{L}^n, \xi^n) \Longrightarrow (N^*, V^*, L^*, \xi^*), \text{ where } \xi^* \text{ is } (\theta, \Gamma)\text{BM}. \tag{4.8}$$

Next, recall that S_k is the counting process associated with the partial sums process V_k. From Theorem 1 of [14], (4.3) implies that

$$\hat{S}_k^n \Longrightarrow -\tau_k^{-3/2} V_k^*. \tag{4.9}$$

Finally, consider the special case in which $\{u_q(i), i \geq 1\}$ and $\{v_k(i), i \geq 1\}$, are mutually independent sequences of i.i.d. random variables such that $u_q(i)$ and $v_k(i)$ have squared coefficients

of variation c_{aq}^2 and c_{sk}^2 respectively (the squared coefficient of variation of a random variable is defined to be its variance divided by the square of its mean). Then $N_q^{(n)}$ is a renewal process with rate $\lambda_q^{(n)}$, and a simple application of the functional central limit theorem for renewal processes [6] proves that $\hat{N_q}^n \Longrightarrow N_q^*$, where N_q^* is $(0, \lambda c_{aq}^2)$BM. Because $L_{jk}^{(n)}$ is a compound renewal process, $\hat{L}^n$ converges to $(0, \Omega)$BM by Theorem 2.1 of [23]. In particular, the covariance matrix is of the form

$$\Omega_{kl} = \begin{cases} \lambda_k \tau_k^2 (c_{sk}^2 + c_{aq(k)}^2) & k = l, \\ \lambda_k \tau_k \tau_l c_{aq(k)}^2 & k \neq l,\ q(k) = q(l), \\ 0 & \text{otherwise.} \end{cases} \tag{4.10}$$

It is not necessary to assume i.i.d. sequences for the convergence in (4.3) to hold. See Glynn [9] for examples of convergent sequences with dependencies and non-stationarity.

5 Additional Notation and Preliminaries

We devote this section to developing some additional notation and preliminary results that will be useful for future reference. For a station $j \in \sigma(0)$ we define its "depth" to be $\mathsf{d}(j) = 0$. The depths of all other stations are then defined recursively as:

$$\mathsf{d}(j) = 1 + \max\{\mathsf{d}(i), i > 0, i \in \pi(j)\}. \tag{5.1}$$

It follows from the feedforward structure that such a notion of depth is well defined and that $\mathsf{d}(j) \leq d$ for all stations $j = 1, \ldots, d$.

In Section 2 we defined $\pi(i)$ to be the set of predecessor *stations* to station i. It will also be useful to define the set of *tasks* preceding station i. Recall that $\mathcal{C}(i)$ is the constituency of station i and $c(i)$ is the cardinality of this set. Rather than labelling tasks in $\mathcal{C}(i)$ by $k = 1, \ldots, K$, we now enumerate these tasks by $a_1^i, \ldots, a_{c(i)}^i$. Set $\mathcal{T}(0) \equiv \{0\}$ and for each station $i = 1, \ldots, d$, define

$$\mathcal{T}(i) = \left\{x = (x_1, \ldots, x_{c(i)}) : x_l \in \mathcal{P}(a_l^i)\right\}. \tag{5.2}$$

Each element of $\mathcal{T}(i)$ is a vector of $c(i)$ components, and each component corresponds to a predecessor task of a constituent task in station i. Set $\mathcal{T}^1(i) \equiv \mathcal{T}(i)$,

$$\mathcal{T}^2(i) \equiv \left\{(x^1, x^2) : x^1 \in \mathcal{T}(i); x^2 = (x_1^2, \ldots, x_{c(i)}^2), x_l^2 \in \mathcal{T}\left(s(x_l^1)\right)\right\},$$

and define $\mathcal{T}^h(i)$ recursively as follows:

$$\begin{aligned} \mathcal{T}^h(i) \equiv \Big\{x = (x^1, \ldots, x^h) : \quad & x^1 \in \mathcal{T}(i), x_{l_1}^2 \in \mathcal{T}\left(s(x_{l_1}^1)\right), \ldots, \\ & x_{l_1 l_2, \ldots, l_{h-1}}^h \in \mathcal{T}\left(s(x_{l_1 l_2, \ldots, l_{h-1}}^{h-1})\right)\Big\}, \end{aligned} \tag{5.3}$$

where $l_1 = 1, \ldots, c(i)$; $l_2 = 1, \ldots, c(s(x^1_{l_1}))$; $\ldots$; $l_{h-1} = 1, \ldots, c(s(x^{h-2}_{l_1 l_2 \ldots l_{h-2}}))$, and we take the convention that $c(0) = 0$. For a station i with depth greater than or equal to h and $x \in \mathcal{T}^h(i)$, we define the set of indices

$$\begin{aligned}\mathcal{L}^h(x) \equiv \Big\{ & l = (l_1, \ldots, l_h) : l_1 = 1, \ldots, c(i);\ l_2 = 1, \ldots, c\left(s(x^1_{l_1})\right);\ \ldots; \\ & l_h = 1, \ldots, c\left(s(x^{h-1}_{l_1 l_2 \ldots l_{h-1}})\right)\Big\}\end{aligned} \tag{5.4}$$

In addition, let

$$\mathcal{L}^h_j(x) = \left\{ l = (l_1, \ldots, l_h) \in \mathcal{L}^h(x) : s(x^h_{l_1,\ldots,l_h}) = j \right\}. \tag{5.5}$$

Taking the network in Figure 2 as an example, we have $\mathsf{d}(4) = 2$ and $\mathcal{T}^2(4) = (w, x, y, z)$ where

$$\begin{array}{lllllllll} w^1 & = & (1,5) & & w^2_1 & = & (0,0) & & w^2_2 & = & (0,0) \\ x^1 & = & (1,6) & & x^2_1 & = & (2,0) & & x^2_2 & = & (0,0) \\ y^1 & = & (3,5) & & y^2_1 & = & (0,0) & & y^2_2 & = & (2,0) \\ z^1 & = & (3,6) & & z^2_1 & = & (2,0) & & z^2_2 & = & (2,0). \end{array} \tag{5.6}$$

For the moment, consider $x \in \mathcal{T}^2(4)$, for which we have

$$\begin{aligned} \mathcal{L}^2(x) &= \{(1,1),(1,2),(2,1),(2,2)\} \\ \mathcal{L}^2_2(x) &= \{(1,1)\}. \end{aligned}$$

One may think of each element $x \in \mathcal{T}^{\mathsf{d}(i)}(i)$ as describing a "path" of tasks traversed by the various job types on their sojourns towards station i. Moreover, $\mathcal{L}^h_j(x)$ identifies the particular "branch(es)" in the path x that would include a visit to station j. Using the notation established above, the following lemma can be verified directly.

Lemma 5.1 *For each station $i =, \ldots, d$ and b_{kl} a sequence of numbers associated with tasks k, l,*

$$\sum_{k \in \mathcal{C}(i)} \left(\max_{l \in \mathcal{P}(k)} b_{kl} \right) = \max_{x \in \mathcal{T}(i)} \left(\sum_{l \in \mathcal{L}(x)} b_{a^i_l x_l} \right).$$

6 The Main Results

Theorem 6.1 *Suppose that assumptions (4.2) and (4.3) hold. Then*

$$(\xi^n, U^n, Z^n, W^n, I^n) \Longrightarrow (\xi^*, U^*, Z^*, W^*, I^*),$$

where for each $i = 1, \ldots, d$ and $k \in \mathcal{C}(i)$: $U^*_{00} \equiv 0$,

$$\xi^* \text{ is a } (\theta, \Gamma) \text{ Brownian motion;} \tag{6.1}$$

$$Z^*_i = \xi^*_i + I^*_i; \tag{6.2}$$

$$X_i^* = \xi_i^* - \sum_{k \in \mathcal{C}(i)} \rho_{ik} \left(\max_{l \in \mathcal{P}(k)} \frac{U^*_{s(l)l}}{\rho_{s(l)l}} \right); \tag{6.3}$$

$$W_i^* = X_i^* + I_i^*; \tag{6.4}$$

$$U_{ik}^* = \rho_{ik} \left[Z_i^* + \max_{l \in \mathcal{P}(k)} \frac{U^*_{s(l)l}}{\rho_{s(l)l}} - \sum_{m \in \mathcal{C}(i)} \rho_{im} \left(\max_{n \in \mathcal{P}(m)} \frac{U^*_{s(n)n}}{\rho_{s(n)n}} \right) \right]; \tag{6.5}$$

$$I_i^* \text{ is continuous and nondecreasing with } I_i^*(0) = 0; \tag{6.6}$$

$$I_i^* \text{ increases only at times } t \text{ with } W_i^*(t) = 0. \tag{6.7}$$

Set

$$\gamma_{kl} = \begin{cases} 0 & \text{if } k = 0 \text{ or } l = 0, \text{ otherwise} \\ 1 - \rho_{s(k)k}, & \text{if } k = l, \\ -\rho_{s(l)l}, & \text{if } k \neq l. \end{cases} \tag{6.8}$$

For notational convenience, we henceforth write s^m to mean $s(x^m_{l_1,\ldots,l_m})$. Denoting by h the depth of a station i, define for each $x \in \mathcal{T}^h(i)$ and $j = 1 \ldots, d$ the following factors:

$$\begin{aligned} \beta_{ij}(x) \;=\; & \sum_{l \in \mathcal{L}_j^1(x)} \rho_{ia^i_l} + \sum_{l=(l_1,l_2) \in \mathcal{L}_j^2(x)} \rho_{ia^i_{l_1}} \gamma_{x^1_{l_1} a^{s^1}_{l_2}} + \cdots \\ & \cdots + \sum_{l=(l_1,\ldots,l_h) \in \mathcal{L}_j^h(x)} \rho_{ia^i_{l_1}} \gamma_{x^1_{l_1} a^{s^1}_{l_2}} \cdots \gamma_{x^{h-1}_{l_1 \cdots l_{h-1}} a^{s^{h-1}}_{l_h}} . \end{aligned} \tag{6.9}$$

We then define the convex polyhedral cone $\mathcal{S}$ to be

$$\mathcal{S} = \bigcap_{i=1}^{d} \bigcap_{x \in \mathcal{T}^{\mathsf{d}(i)}(i)} \left\{ z = (z_1, \ldots, z_d)' : z_i - \sum_{j=1}^{d} \beta_{ij}(x) z_j \geq 0 \right\}. \tag{6.10}$$

It is easily verified that $z \geq 0$ if $z \in \mathcal{S}$. For each $i = 1, \ldots, d$, we also define the boundary set

$$F^i = \bigcup_{x \in \mathcal{T}^{\mathsf{d}(i)}(i)} \left\{ z \in \mathcal{S} : z_i - \sum_{j=1}^{d} \beta_{ij}(x) z_j = 0 \right\}. \tag{6.11}$$

Theorem 6.2 *For each $i = 1, \ldots, d$,*

$$\xi^* \text{ is a } (\theta, \Gamma) \text{ Brownian motion}; \tag{6.12}$$

$$Z_i^* = \xi_i^* + I_i^* \in \mathcal{S}; \tag{6.13}$$

$$W_i^* = Z_i^* - \max_{x \in \mathcal{T}^{\mathsf{d}(i)}(i)} \left(\sum_{j=1}^{d} \beta_{ij}(x) Z_j^* \right); \tag{6.14}$$

$$I_i^* \text{ is continuous and nondecreasing with } I_i^*(0) = 0; \tag{6.15}$$

$$I_i^* \text{ increases only at times } t \text{ with } Z^*(t) \in F^i. \tag{6.16}$$

That is, the process Z^* as defined in Theorem 6.1 is a d-dimensional SRBM whose state space is a convex polyhedral cone $\mathcal{S}$. The SRBM Z^* has drift θ, covariance matrix Γ, and reflection matrix $R = I$ where I is the d-dimensional identity matrix.

Remark: The results of Nguyen [17] may be applied to show that such an SRBM is well defined in the strong sense.

To interpret Theorem 6.2, note that statements (6.13) and (6.15) are reiterations of the characterizations given in (3.7) and (3.4), respectively. Moreover, the approximation (6.12) of the netflow process by a Brownian motion was justified in Section 4 under assumptions (4.2) and (4.3). Next, recall that each element $x \in \mathcal{T}^{\mathsf{d}(i)}(i)$ describes a "path" of tasks traversed by the various constituent jobs on their way to station i. Equation (6.15) states that the immediate workload at station i is the minimum amount of work found among all the "paths" leading up to station i. In other words, (6.15) articulate the constraint that a task at station i cannot be processed until all of its precedessor tasks have been completed (this is the definition of a join node). With this interpretation in mind, statement (6.16) is then equivalent to (3.5). Thus, each idleness process is associated with potentially multiple boundaries on the state space $\mathcal{S}$. In Nguyen [18], it was argued that the additional faces correspond to the fork and join constraints in the network. As we will demonstrate in an example, the polyhedral state space associated with heterogeneous fork-join networks typically has many more faces than its homogeneous counterpart. These additional faces may be interpreted as results of the disordering effects that occur when jobs fork and join in their sojourns through the network.

Example 1: The Sample Fork-Join Network

The heavy traffic limit of the network pictured in Figure 2 is given by

$$\xi^* \text{ is a } (\theta, \Gamma) \text{ Brownian motion;} \tag{6.17}$$

$$Z_i^* = \xi_i^* + I_i^*; \tag{6.18}$$

$$I_i^* \text{ is continuous and nondecreasing with } I_i^*(0) = 0; \tag{6.19}$$

$$I_i^* \text{ increases only at times t with } Z_i^*(t) = 0,\ i = 1, 2; \tag{6.20}$$

$$I_3^* \text{ increases only at times t with } Z_3^*(t) - \rho_{33} Z_2^*(t) = 0; \tag{6.21}$$

I_4^* increases only at times t such that one of the following conditions hold:

$$Z_4^*(t) - Z_1^*(t) = 0, \text{ OR} \tag{6.22}$$

$$Z_4^*(t) - (\rho_{44}\rho_{36} - \rho_{47}\rho_{33}) Z_2^*(t) - Z_3^*(t) = 0, \text{ OR} \tag{6.23}$$

$$Z_4^*(t) - \rho_{44} Z_1^*(t) + \rho_{47}\rho_{33} Z_2^*(t) - \rho_{47} Z_3^*(t) = 0, \text{ OR} \tag{6.24}$$

$$Z_4^*(t) - \rho_{47} Z_1^*(t) - \rho_{44}\rho_{36} Z_2^*(t) - \rho_{44} Z_3^*(t) = 0. \tag{6.25}$$

Hence, setting

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -\rho_{33} & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -(\rho_{44}\rho_{36} - \rho_{47}\rho_{33}) & -1 & 1 \\ -\rho_{44} & \rho_{47}\rho_{33} & -\rho_{47} & 1 \\ -\rho_{47} & -\rho_{44}\rho_{36} & -\rho_{44} & 1 \end{bmatrix}$$

and letting $\mathcal{S} = \{x : Ax \geq 0\}$, Z^* is a 4-dimensional SRBM with drift θ, covariance matrix Γ, reflection matrix $R = I$, and whose state space is the convex polyhedral cone $\mathcal{S}$ with seven faces. We now turn to the interpretation of conditions (6.20)-(6.25). Because the immediate workload is identical to the total workload process at stations 1 and 2, condition (6.20) states that the idleness processes at these stations increases only when the immediate workload at the respective stations is zero. Emulating the arguments in Harrison and Nguyen [12, 13], one may think of $\lambda_2 Z_2^*(t)$ as the number of tasks 2 occupying station 2 at time t. Consequently, $\tau_3\lambda_2 Z_2^*(t) = \rho_{33} Z_2^*(t)$ is the amount of work associated with task 3 destined for station 3 that still reside at station 2 at time t. Hence $W_3^*(t) = Z_3^*(t) - \rho_{33} Z_2^*(t)$ and condition (6.21) specifies that the idleness process at station 3 increases only when there is no immediate work at that station.

Conditions (6.22)-(6.25) describe the four scenarios under which server 4 is forced to remain idle. It can be verified that conditions (6.22), (6.23), (6.24), (6.25) correspond to the paths described by w, z, y, and x of equation (5.6), respectively. The first path, w, contains the predecessor tasks 1 and 5, hence condition (6.22) states that the total amount of work destined for station 4 is contained in those tasks currently waiting for service at station 1 either in the form of task 1 or task 5. That is, the immediate work content in buffer $(1,4)$ (the buffer joining stations 1 and 4), given by $Z_4^*(t) - Z_1^*(t)$, is zero. Path z, on the other hand, contains the predecessor tasks 3 and 6, and one can apply a similar argument to arrive at the conclusion that condition (6.23) corresponds to the scenario in which buffer $(3,4)$ is empty. That is, the immediate work content in buffer $(3,4)$ is given by $Z_4^*(t) - (\rho_{44}\rho_{36} - \rho_{47}\rho_{33})\, Z_2^*(t) - Z_3^*(t)$. Because station 4 is a join node for both types of customers, it seems clear that both buffers $(2,4)$ (the buffer joining stations 2 and 4) and $(3,4)$ (the buffer joining stations 3 and 4) must be *nonempty* if server 4 is to remain busy. However, these are not the only times at which server 4 may be idle. Condition (6.24), which corresponds to path x, states that the total work destined for station 4 are completely contained in tasks 3 at station 2 and tasks 5 at station 1. That is, buffer $(1,4)$ contains no tasks corresponding to type 2 jobs and buffer $(3,4)$ does not have any type 1 work. Because there are no complete tasks for station 4, the server is forced to remain idle although both of its incident buffers may be nonempty. Similarly, condition (6.25) describes the converse situation in which buffer $(1,4)$ has only type 2 tasks and buffer $(3,4)$ contains only type 1 tasks. ∎

Example 2: Multiple Customer Types with Common Routing Structure

Consider a fork-join network in which all customers share the same routing constraints, but the interarrival times and service times may be different across job types. This class of networks correspond to a particularly simple case of the networks considered in this paper. For stations $j \in \sigma(0)$, it follows from (6.5) that $U_{ik}^* = \rho_{ik} Z_i^*$. By induction on the depth of stations, one can verify that equation (6.5) becomes

$$U_{ik}^* = \rho_{ik} Z_i^* + \rho_{ik} \max_{l \in \mathcal{P}(k)} Z_{s(l)}^* - \rho_{ik} \sum_{m \in \mathcal{C}(i)} \rho_{im} \max_{n \in \mathcal{P}(m)} Z_{s(n)}^*. \tag{6.26}$$

Because all customer types share the same the same routing structure, the sets $\{s(n) : n \in \mathcal{P}(m)\}$ are identical for each task $m \in \mathcal{C}(i)$. Hence the last term on the right side of (6.26) is equal to

$$-\rho_{ik} \left(\sum_{m \in \mathcal{C}(i)} \right) \rho_{im} \max_{j \in \pi(i)} Z_j^* = -\rho_{ik} \max_{j \in \pi(i)} Z_j^*$$

and equation (6.26) reduces to

$$U_{ik}^* = \rho_{ik} Z_i^*.$$

Hence, for $i = 1, \ldots, d$ and $k \in \mathcal{C}(i)$,

ξ^* is a (θ, Γ) Brownian motion;

$Z_i^* = \xi_i^* + I_i^*$;

$W_i^* = Z_i^* - \max_{j \in \pi(i)} Z_j^*$;

$U_{ik}^* = \rho_{ik} Z_i^*$;

I_i^* is continuous and nondecreasing with $I_i^*(0) = 0$;

I_i^* increases only at times t with $W_i^*(t) = 0$.

Fork-join networks with one customer type (that is, homogeneous fork-join networks) are clearly a special subset of the networks discussed in this section. It is straightforward to verify that the results above agree with those given in Nguyen [18]. ∎

Example 3: Feedforward Multi-Class Queueing Networks

Consider now the feedforward multi-class queueing network studied by Peterson [19]. The networks described in Peterson [19] are essentially similar to those considered here with one important exception: The networks in [19] require that $\mathcal{P}(k)$ contains at most one element for each task k; that is, there are no join nodes. (Peterson's work does not explicitly consider the case in which

tasks may fork, but the inclusion of the forking structure would not pose much hardship to his analysis.) Recall that $Z_i^*(t) = \sum_{k \in \mathcal{C}(i)} U_{ik}^*(t)$, hence equations (6.2)-(6.5) imply

$$U_{ik}^* = \rho_{ik} \left[W_i^* + \max_{l \in \mathcal{P}(k)} \frac{U_{s(l)l}^*}{\rho_{s(l)l}} \right], \tag{6.27}$$

$$Z_i^* = W_i^*(t) + \sum_{k \in \mathcal{C}(i)} \max_{l \in \mathcal{P}(k)} \rho_{ik} \frac{U_{s(l)l}^*}{\rho_{s(l)l}}. \tag{6.28}$$

We denote by $\mathsf{p}(k)$ the one predecessor task of task k, and efine $\mathsf{p}(0) = 0$. Setting $\mathsf{p}^1(k) \equiv \mathsf{p}(k)$, we recursively define $\mathsf{p}^h(k) \equiv \mathsf{p}(\mathsf{p}^{h-1}(k))$. Letting $U_{00}^* = W_0^* = 0$, equations (6.27)-(6.28) thus reduce to

$$U_{ik}^* = \rho_{ik} \left[W_i^* + \frac{U_{s(\mathsf{p}^1(k))\mathsf{p}^1(k)}^*}{\rho_{s(\mathsf{p}^1(k))\mathsf{p}^1(k)}} \right]$$

$$Z_i^* = W_i^*(t) + \sum_{k \in \mathcal{C}(i)} \rho_{ik} \frac{U_{s(\mathsf{p}^1(k))\mathsf{p}^1(k)}^*}{\rho_{s(\mathsf{p}^1(k))\mathsf{p}^1(k)}}.$$

Similarly to the previous example, we can use induction to show that for a station i of depth h,

$$U_{ik}^* = \rho_{ik} \left[W_i^* + \sum_{g=1}^{h} W_{s(\mathsf{p}^g(k))}^* \right]$$

$$Z_i^* = W_i^*(t) + \sum_{g=1}^{h} \sum_{k \in \mathcal{C}(i)} \rho_{ik} W_{s(\mathsf{p}^g(k))}^*.$$

Readers can verify that this agrees with the result obtained by Peterson [19]. ∎

Theorem 6.3 *Under assumptions (4.2) and (4.3),* $(T_1^n, \ldots, T_p^n) \Longrightarrow (T_1^*, \ldots, T_p^*)$ *where*

$$T_q^*(t) = \max_{k \in \mathcal{A}_q^e} T_{qk}^*(t),$$

$$T_{qk}^*(t) = \max_{l \in \mathcal{P}(k)} T_{ql}^*(t) + W_{s(k)}^*, \quad T_{q0}^*(t) \equiv 0.$$

If we denote by l_q the PERT/CPM "longest path operator" for type q jobs, Theorem 6.3 implies the representation

$$T_q^*(t) = l_q(W_{s(k)}^*, k \in \mathcal{A}_q). \tag{6.29}$$

As discussed in Nguyen [18], expression (6.29) is an example of Reiman's "snapshot" principle [20]. That is, in the heavy traffic scaling, the fluctuation in workload levels is insignificantly small relative to the length of time that a job spends in the system, hence a "snapshot" of the system at the time of a job's arrival remains representative throughout the job's sojourn in the network. Equation (6.29) expresses the remarkable result that sojourn time analysis of a fork-join network may be phrased in terms of the familiar longest path analysis of PERT/CPM methods, where traditional task times are now replaced by waiting times at stations corresponding to the tasks.

7 Proofs

By Skorohod's representation theorem and the continuity of Brownian motions, we can and will assume that the convergence in (4.3) is almost surely uniform on compact sets; that is, we henceforth assume

$$(\hat{N}^*, \hat{V}^*, \hat{L}^*) \to (\hat{N}^*, \hat{V}^*, \hat{L}^*) \text{ u.o.c.} \tag{7.1}$$

We begin the proof of Theorem 6.1 with a few preliminary results. The first lemma is an immediate consequence of assumptions (4.2) and (7.1).

Lemma 7.1 *$\xi^n \to \xi^*$ u.o.c., where ξ^* is a (θ, Γ) Brownian motion where $\Gamma = C\Omega C'$.*

Lemma 7.2 *For $k = 1, \ldots, K$, $j = 1, \ldots, d$, let $\epsilon^n_{1k}(t) = n^{-1/2}\epsilon^{(n)}_{1k}(nt)$ and $\epsilon^n_{2j}(t) = n^{-1/2}\epsilon^{(n)}_{2j}(nt)$. Then $\epsilon^n_{1k} \to 0$ and $\epsilon^n_{2j} \to 0$ u.o.c.*

Proof. Note that

$$0 \le \epsilon^{(n)}_{1k}(t) \le \max_{1 \le i \le N^{(n)}_{q(k)}(t)} \tau^{(n)}_k v_k(i)$$

and

$$0 \le \epsilon^{(n)}_{2j}(t) \le \max_{k \in \mathcal{C}(j)} \max_{1 \le i \le N^{(n)}_{q(k)}(t)} \tau^{(n)}_k v_k(i)$$

The lemma follows directly from assumption (7.1) and Lemma 3.3 of Iglehart and Whitt [15]. ∎

Let $\bar{\eta}^n_j(t) = n^{-1}\eta^{(n)}_j(nt)$ and $\hat{\eta}^n_j(t) = n^{-1/2}(nt - \eta^{(n)}_j(nt))$.

Lemma 7.3 *If $W^n_j \to W^*_j$ u.o.c., then $\bar{\eta}^n_j \to e$ u.o.c. where $e(t) = t$.*

Proof. Because $\bar{\eta}^n_j(t) \le t$, it follows from equation (3.9) that for each $t \ge 0$,

$$\begin{aligned} \|e(\cdot) - \bar{\eta}^n_j(\cdot)\|_t &\le n^{-1/2}\|W^n_j(\bar{\eta}^n_j(\cdot))\|_t + n^{-1/2}\|\epsilon^n_2(\cdot)\|_t \\ &\le n^{-1/2}\|W^n_j(\cdot)\|_t + n^{-1/2}\|\epsilon^n_2(\cdot)\|_t. \end{aligned}$$

From Lemma 7.2 and the assumption that $W^n_j \to W^*_j$ u.o.c., it follows that $\|e(\cdot) - \bar{\eta}^n_j(\cdot)\|_t \to 0$. ∎

Lemma 7.4 *If $W^n_j \to W^*_j$ u.o.c., then $\hat{\eta}^n_j \to W^*_j$ u.o.c..*

Proof. It follows from equation (3.9) that for each $t \ge 0$,

$$\|\hat{\eta}^n_j(\cdot) - W^*_j(\cdot)\|_t \le \|W^*_j(\cdot) - W^n_j(\cdot)\|_t + \|W^n_j(\bar{\eta}^n_j(\cdot)) - W^n_j(\cdot)\|_t + \|\epsilon^n_2(\cdot)\|_t. \tag{7.2}$$

As a result of Lemma 7.3 and the continuity of W_j^*, the first two terms on the right side of (7.2) converges to zero. Invoking Lemma 7.2, one concludes that $\hat{\eta}_j^n \to W_j^*$ u.o.c. ∎

The proof of Theorem 6.1 proceeds by induction on the depth of stations. We begin with the following result for stations of depth 0.

Lemma 7.5 *Theorem 6.1 holds for all stations of depth 0, namely, stations $j \in \sigma(0)$.*

Proof. Note that $\mathcal{P}(k) = \emptyset$ for each task $k \in \mathcal{C}(j)$ when $j \in \sigma(0)$. Because $X_j^n = \xi_j^n$ for $j \in \sigma(0)$, it follows from Lemma 7.1, equations (3.3), (3.6), (3.7), and the continuous mapping theorem that $(X_j^n, W_j^n, I_j^n, Z_j^n) \to (X_j^*, W_j^*, I_j^*, Z_j^*)$ u.o.c. where $X_j^* = \xi_j^*$, $I_j^*(t) = -\inf_{0 \le s \le t} X_j^*(s)$, $W_j^*(t) = X_j^*(t) + I_j^*(t)$, and $Z_j^*(t) = \xi_j^*(t) + I_j^*(t)$. Because

$$U_{jk}^n(t) = L_{jk}^n(t) - L_{jk}^n(\bar{\eta}_j^n(t)) + \rho_{jk}^{(n)} \hat{\eta}_j^n(t) - \epsilon_1^n(t),$$

it follows from Lemma 7.2, Lemma 7.4, and the continuity of L_{jk}^* that $U_{jk}^n \to U_{jk}^*$ u.o.c. where $U_{jk}^*(t) = \rho_{jk} W_j^*(t) = \rho_{jk} Z_j^*(t)$. Joint convergence of the processes of interest is a natural consequence of their continuity. ∎

Proof of Theorem 6.1: Define

$$\hat{Y}_{jk}^n(t) = n^{-1/2}(Y_{jk}^{(n)}(nt) - \rho_{jk}^{(n)} t) \quad \text{and} \quad \bar{Y}_{jk}^n(t) = n^{-1} Y_{jk}^{(n)}(nt),$$

and note that as a consequence of (3.11),

$$\hat{Y}_{jk}^n(t) = \hat{L}_{jk}^n(t) - U_{jk}^n(t). \tag{7.3}$$

With Lemma 7.5 we may assume inductively that the convergence in Theorem 6.1 has been established for all stations with depth h or less. Consider a station j with $\mathsf{d}(j) = h + 1$. For each task $k \in \mathcal{C}(j)$, it follows from (3.12) that

$$\hat{A}_{jk}^n(t) = \begin{cases} \hat{N}_{q(k)}^n(t) & \text{if } \mathcal{P}(k) = \emptyset, \\ \min_{l \in \mathcal{P}(k)} \left\{ \hat{S}_{s(l)l}^n(\bar{Y}_{s(l)l}^n(t)) + \frac{1}{\tau_l^{(n)}} \hat{Y}_{s(l)l}^n(t) \right\} & \text{otherwise.} \end{cases}$$

(Note that $\lambda_k = \lambda_l$ for all $l \in \mathcal{P}(k)$.) Setting $\bar{A}_{jk}^n(t) \equiv n^{-1} A_{jk}^{(n)}(nt)$, equation (3.13) gives

$$\hat{M}_{jk}^n(t) = \hat{V}_k^n(\bar{A}_{jk}^n(t)) + \tau_k^{(n)} \hat{A}_{jk}^n(t).$$

Because $\mathsf{d}(s(l)) \leq h$ for all tasks $l \in \mathcal{P}(k)$, it follows from (7.3) and the induction hypothesis that $\hat{Y}^n_{s(l)l} \to Y^*_{s(l)l}$ u.o.c. where $Y^*_{s(l)l}(t) = L^*_{s(l)l}(t) - U^*_{s(l)l}(t)$, and $\bar{Y}^n_{s(l)l} \to \rho_{s(l)l} \cdot e$ u.o.c. where that $e(t) = t$. Consequently, if $\mathcal{P}(k) \neq \emptyset$, $\hat{M}^n_{jk} \to M^*_{jk}$ u.o.c. with

$$\begin{aligned} M^*_{jk}(t) &= V^*_k(\lambda_k t) + \tau_k \min_{l \in \mathcal{P}(k)} \left\{ -\tau_l^{-3/2} V^*_l(\rho_{s(l)l} t) + \tau_l^{-1} L^*_{s(l)l}(t) - \tau_l^{-1} U^*_{s(l)l}(t) \right\} \\ &= V^*_k(\lambda_k t) + \tau_k \min_{l \in \mathcal{P}(k)} \left\{ -\tau_l^{-1} V^*_l(\lambda_l t) + \tau_l^{-1} \left(V^*_l(\lambda_l t) + \tau_l N^*_{q(l)}(t) \right) - \tau_l^{-1} U^*_{s(l)l}(t) \right\} \\ &= V^*_k(\lambda_k t) + \tau_k N^*_{q(k)}(t) - \tau_k \max_{l \in \mathcal{P}(k)} \frac{U^*_{s(l)l}(t)}{\tau_l}, \end{aligned} \tag{7.4}$$

where the last equality follows because $q(k) = q(l)$ for $l \in \mathcal{P}(k)$. On the other hand, if $\mathcal{P}(k) = \emptyset$, we have $\hat{M}^n_{jk} \to M^*_{jk}$ u.o.c. with

$$M^*_{jk} = V^*_k(\lambda_k t) + \tau_k N^*_{q(k)}(t).$$

Applying the continuous mapping theorem to equation (3.14), $X^n_j \to X^*_j$ u.o.c. where

$$\begin{aligned} X^*_j(t) &= \sum_{k \in \mathcal{C}(j)} \left(V^*_k(\lambda_k t) + \tau_k N^*_{q(k)}(t) \right) - \sum_{k \in \mathcal{C}(j)} \max_{l \in \mathcal{P}(k)} \frac{\tau_k}{\tau_l} U^*_{s(l)l}(t) + \theta_j t \\ &= \xi^*_j(t) - \sum_{k \in \mathcal{C}(j)} \rho_{jk} \max_{l \in \mathcal{P}(k)} \frac{U^*_{s(l)l}(t)}{\rho_{s(l)l}}, \end{aligned}$$

and we use the convention that $\max \emptyset = 0$. That $(W^n_j, I^n_j, Z^n_j) \to (W^*_j, I^*_j, Z^*_j)$ u.o.c. is again a consequence of the continuous mapping theorem by virtue of equations (3.3), (3.6), and (3.7). All that remains is to prove the convergence of U^n_j. From Lemma 7.3 and Lemma 7.4, we have $\bar{\eta}^n_j \to e$ u.o.c. and $\hat{\eta}^n_j \to W^*_j$ u.o.c. Consequently, it follows from (3.8) that

$$Y^n_j(t) = \hat{M}^n_{jk}(\bar{\eta}^n_j(t)) - \rho^{(n)}_{jk} \hat{\eta}^n_j(t) + \epsilon^n_{1k}(t),$$

from which we can conclude $Y^n_j \to Y^*_j$ u.o.c. where

$$Y^*_j(t) = M^*_{jk}(t) - \rho_{jk} W^*_j(t). \tag{7.5}$$

Because $U^n_{jk}(t) = \hat{L}^n_{jk}(t) - \hat{Y}^n_{jk}(t)$, it follows from (7.4) and (7.5) that $U^n_{jk} \to U^*_{jk}$ u.o.c. and

$$\begin{aligned} U^*_{jk}(t) &= L^*_{jk}(t) - T^*_{jk}(t) \\ &= L^*_{jk}(t) - \left(V^*_k(\lambda_k t) + \tau_k N^*_{q(k)}(t) \right) + \tau_k \max_{l \in \mathcal{P}(k)} \frac{U^*_{s(l)l}(t)}{\tau_l} + \rho_{jk} W^*_j(t) \\ &= \rho_{jk} Z^*_j(t) + \rho_{jk} \max_{l \in \mathcal{P}(k)} \frac{U^*_{s(l)l}(t)}{\rho_{s(l)l}} - \rho_{jk} \sum_{m \in \mathcal{C}(j)} \rho_{jm} \max_{n \in \mathcal{P}(m)} \frac{U^*_{s(n)n}(t)}{\rho_{s(n)n}}. \end{aligned}$$

∎

Proof of Theorem 6.2: Define γ_{kl} as in (6.8), and rewrite equation (6.5) as

$$U^*_{ik} = \rho_{ik}\left[Z^*_i + \sum_{m\in\mathcal{C}(i)} \gamma_{km}\left(\max_{n\in\mathcal{P}(m)} \frac{U^*_{s(n)n}}{\rho_{s(n)n}}\right)\right]. \tag{7.6}$$

Recall that a^i_l, $l = 1,\ldots,c(i)$, enumerate the elements of $\mathcal{C}(i)$. Applying Lemma 5.1 to (7.6), we obtain

$$U^*_{ik} = \rho_{ik}\left[Z^*_i + \max_{x\in\mathcal{T}(i)} \left(\sum_{l=1}^{c(i)} \gamma_{ka^i_l} \frac{U^*_{s(x_l)x_l}}{\rho_{s(x_l)x_l}}\right)\right]. \tag{7.7}$$

Similarly, equation (6.3) is equivalent to the following expression as a result of Lemma 5.1,

$$X^*_i = \xi^*_i - \max_{x\in\mathcal{T}(i)} \left(\sum_{l=1}^{c(i)} \rho_{ia^i_l} \frac{U^*_{s(x_l)x_l}}{\rho_{s(x_l)x_l}}\right). \tag{7.8}$$

Substituting (6.2) and (6.3) in (6.4) and applying (7.8), we have

$$W^*_i = Z^*_i - \max_{x\in\mathcal{T}(i)} \left(\sum_{l=1}^{c(i)} \rho_{ia^i_l} \frac{U^*_{s(x_l)x_l}}{\rho_{s(x_l)x_l}}\right).$$

Substituting (7.7) in the above expression, we obtain

$$\begin{aligned} W^*_i &= Z^*_i - \max_{x^1\in\mathcal{T}(i)} \left(\sum_{l_1=1}^{c(i)} \rho_{ia^i_{l_1}} \left[Z^*_{s(x^1_{l_1})} + \max_{x^2_{l_1}\in\mathcal{T}(s(x^1_{l_1}))} \sum_{l_2=1}^{c(s(x^1_{l_1}))} \gamma_{x^1_{l_1} a^{s(x^1_{l_1})}_{l_2}} \frac{U^*_{s(x^2_{l_1l_2})x^2_{l_1l_2}}}{\rho_{s(x^2_{l_1l_2})x^2_{l_1l_2}}}\right]\right) \\ &= Z^*_i - \max_{x=(x^1,x^2)\in\mathcal{T}^2(i)} \left(\sum_{l_1=1}^{c(i)} \rho_{ia^i_{l_1}} Z^*_{s(x^1_{l_1})} + \sum_{l_1=1}^{c(i)} \sum_{l_2=1}^{c(s(x^1_{l_1}))} \rho_{ia^i_{l_1}} \gamma_{x^1_{l_1} a^{s(x^1_{l_1})}_{l_2}} \frac{U^*_{s(x^2_{l_1l_2})x^2_{l_1l_2}}}{\rho_{s(x^2_{l_1l_2})x^2_{l_1l_2}}}\right). \end{aligned} \tag{7.9}$$

For notational convenience, we henceforth write s^m to mean $s(x^m_{l_1\ldots l_m})$. Substituting (7.7) in (7.9) recursively, one can verify that for a station of depth h

$$\begin{aligned} W^*_i = Z^*_i - \max_{x\in\mathcal{T}^h(i)} \Bigg(&\sum_{l_1=1}^{c(i)} \rho_{ia^i_{l_1}} Z^*_{s^1} + \sum_{l_1=1}^{c(i)} \sum_{l_2=1}^{c(s^1)} \rho_{ia^i_{l_1}} \gamma_{x^1_{l_1} a^{s^1}_{l_2}} Z^*_{s^2} + \cdots \\ &\sum_{l_1=1}^{c(i)} \cdots \sum_{l_h=1}^{c(s^{h-1})} \rho_{ia^i_{l_1}} \gamma_{x^1_{l_1} a^{s^1}_{l_2}} \cdots \gamma_{x^{h-1}_{l_1\ldots l_{h-1}} a^{s^{h-1}}_{l_h}} Z^*_{s^h}\Bigg). \end{aligned} \tag{7.10}$$

It is straightforward to verify that (7.10) is equivalent to equation (6.15) and the theorem is thus proved. ∎

Remark: Substituting (6.2)-(6.4) in equation (6.5), we obtain

$$U_{ik}^* = \rho_{ik}\left[W_i^* + \max_{l\in\mathcal{P}(k)} \frac{U_{s(l)l}^*}{\rho_{s(l)l}}\right].$$

Because $Z_i^*(t) = \sum_{k\in\mathcal{C}(i)} U_{ik}^*(t)$, it follows that

$$Z_i^* = W_i^* + \sum_{k\in\mathcal{C}(i)} \max_{l\in\mathcal{P}(k)} \rho_{ik}\frac{U_{s(l)l}^*}{\rho_{s(l)l}}. \tag{7.11}$$

Proceeding in the same manner as in the proof of Theorem 6.2, we can show that for a station i of depth h, (7.11) can be written as

$$Z_i^* = W_i^* + \max_{x\in\mathcal{T}^h(i)} \left\{ \sum_{l\in\mathcal{L}^1(x)} \rho_{ia_{l_1}^i} W_{s^1}^* + \cdots + \sum_{l\in\mathcal{L}^h(x)} \rho_{ia_{l_1}^i} W_{s^h}^* \right\}. \tag{7.12}$$

Readers may recognize that equation (7.12) is an "inverse" formulation of the relationship described in (6.15). It states that the amount of total work in the system for station i is the maximum of the amount of immediate work destined for station i found along each path to that station.

Proof of Theorem 6.3: Define $\bar{\Phi}_q^n(t) \equiv n^{-1}\Phi_q^{(n)}(nt)$ and $\bar{\Phi}_{qk}^n(t) \equiv n^{-1}\Phi_{qk}^{(n)}(nt)$. In addition, note that

$$t \le \Phi_q^{(n)}(t) \le t + \max_{1\le i\le N_q^{(n)}(t)+1} u_k(i)/\lambda_k^{(n)},$$

hence by Lemma 3.3 of Iglehart and Whitt [15],

$$\bar{\Phi}_q^n \to e \text{ u.o.c.} \tag{7.13}$$

We begin with tasks $k \in \mathcal{A}_q^0$, for which

$$\begin{aligned} \bar{\Phi}_{qk}^n(t) &\equiv \bar{\Phi}_q^n(t) \quad \text{and} \\ T_{qk}^n(t) &\equiv W_{s(k)}^n(\bar{\Phi}_{qk}(t)). \end{aligned}$$

It follows from (7.13) and Theorem 6.1 that $\bar{\Phi}_{qk}^n \to e$ u.o.c. and $T_{qk}^n \to T_{qk}^*$ where $T_{qk}^* = W_{s(k)}^*(t)$. The theorem is then proved by applying induction on (3.15)-(3.16). ∎

8 Concluding Remarks

We presented in this paper a heavy traffic analysis of feedforward fork-join networks with heterogeneous customers. We made several assumptions to simplify the exposition, but the results proved here apply for more general networks as well. For example, we assumed that each station is staffed a

single server. Using the machinery developed by Chen and Shanthikumar [7], one can extend these results to fork-join networks of multi-server queues. Secondly, whereas we assumed that all servers are reliable, it is possible to analyze networks in which stations may experience server breakdown [13, 8]. Lastly, batch arrivals can be accommodated within the framework presented here [20]. (The model discussed by Baccelli and Liu[4] is an example of such networks would thus become a special case of Example 2). In this case, the issue reduces to calculating Ω, the covariance matrix of the total workload input process [20].

Acknowledgements: I would like to thank Professors J. Michael Harrison and Avi Mandelbaum for the many helpful conversations throughout the course of this research.

References

[1] Adler, P., Mandelbaum, A., Nguyen, V., and Schwerer, E. (1992). From Project to Process Management in Engineering: An Empirically-based Framework for Analysis of Product Development, in preparation.

[2] Adler, P., Mandelbaum, A., Nguyen, V., and Schwerer, E. (1992). From Project to Process Management in Engineering: Strategies for Improving Development Cycle Time, in preparation.

[3] Adler, P., Mandelbaum, A., Nguyen, V., and Schwerer, E. (1992). Managerial and Methodological Challenges in Engineering Process Management: Lessons from a Case Study. UCLA Design Conference, Los Angeles.

[4] Baccelli, F. and Liu, Z. (1990). On the Execution of Parallel Programs on Multiprocessor Systems – A Queueing Theory Approach. *J. ACM* **37** 373-414.

[5] Baccelli, F. and Makowski, A. M. (1989). Queueing Models for Systems with Synchronization Constraints. *Proceedings of the IEEE* **77** 138-161.

[6] Billingsley, P. (1968). *Convergence of Probability Measures.* Wiley, New York.

[7] Chen, H. and Shanthikumar, J. G. (1992). Fluid Limits and Diffusion Approximations for Networks of Multi-Server Queues in Heavy Traffic, submitted.

[8] Chen, H. Whitt, W. (1992). Diffusion Approximations for Open Queueing Networks with Server Interruptions, submitted.

[9] Glynn, P. W. (1990). Diffusion Approximations. *Handbook on OR&MS, Vol. 2*, D. P. Heyman and M. J. Sobel (Eds). Elsevier Science Publishers B.V., North Holland 145-198.

[10] Harrison, J. M. (1985). *Brownian Motion and Stochastic Flow Systems.* John Wiley & Sons, New York.

[11] Harrison, J. M. (1988). Brownian Models of Queueing Networks with Heterogeneous Customer Populations. *Proceedings of the IMA Workshop on Stochastic Differential Systems.* Springer-Verlag, New York.

[12] Harrison, J. M. and Nguyen, V. (1990). The QNET Method for Two-Moment Analysis of Open Queueing Networks. *Queueing Systems* **6** 1-32.

[13] Harrison, J. M. and Nguyen, V. Brownian Models of Multiclass Queueing Networks: Current Status and Open Problems. *Queueing Systems*, to appear.

[14] Iglehart, D. L. and Whitt, W. (1969). The Equivalence of Functional Central Limit Theorems of Counting Processes and Associated Partial Sums. Technical Report No. 121. Dept of Operations Research, Stanford University.

[15] Iglehart, D. L. and Whitt, W. (1970). Multiple Channel Queues in Heavy Traffic, I and II. *Adv. Appl. Prob.* **2** 150-177 and 355-364.

[16] Mandelbaum, M. and Avi-Itzhak, B. (1968). Introduction to Queueing with Splitting and Matching. *Israel J. of Tech.* **6** 376-382.

[17] Nguyen, V. (1990). Heavy Traffic Analysis of Processing Networks with Parallel and Sequential Tasks. Ph.D. Dissertation, Department of Operations Research, Stanford University.

[18] Nguyen, V. Processing Networks with Parallel and Sequential Tasks: Heavy Traffic Analysis and Brownian Limits. *Ann. Appl. Prob.*, to appear.

[19] Peterson W. P. (1991). Diffusion Approximations for Networks of Queues with Multiple Customer Types. *Math. Oper. Res.* **16** 90-118.

[20] Reiman, M. I. (1984). Open Queueing Networks in Heavy Traffic. *Math. of Oper. Res.* **9**, 441-458.

[21] Reiman, M. I. (1988). A Multiclass Feedback Queue in Heavy Traffic. *Adv. Appl. Prob.* **20**, 179-207.

[22] Varma, S.(1990). Heavy and Light Traffic Approximations for Queues with Synchronization Constraints. Ph.D. Thesis, Dept. of Elect. Eng., University of Maryland.

[23] Whitt, W. (1971). Weak Convergence Theorems for Priority Queues: Preemptive-Resume Discipline. *J. Appl. Prob.* **8** 74-94.