The Annals of Applıed Probabılıty
1993, Vol. 3, No. 1, 28–55

PROCESSING NETWORKS WITH PARALLEL AND SEQUENTIAL TASKS: HEAVY TRAFFIC ANALYSIS AND BROWNIAN LIMITS

By Viên Nguyen

Stanford University

In queueing theory one seeks to predict in quantitative terms the congestion delays that occur when jobs or customers complete for processing resources. At present no satisfactory methods exist for the analysis of systems that allow *simultaneous performance* of tasks associated with a single job or customer. We present a heavy traffic analysis for the class of homogeneous fork-join networks in which jobs are routed in a feedforward deterministic fashion. We show that under certain regularity conditions the vector of total job count processes converges weakly to a multidimensional *reflected Brownian motion* (RBM) whose state space is a *polyhedral cone* in the nonnegative orthant. Furthermore, the weak limits of workload levels and throughput times are shown to be simple transformations of the RBM. As will be explained, the "steady-state throughput time" (a random variable) is expressed in terms of workload levels via the "longest path functional" of classical PERT/CPM analysis.

1. Introduction and summary. In conventional queueing network theory each arriving "customer" is assumed to require certain "services," and those services are provided in sequential fashion at specified "work centers" or "stations" of the network. Equivalently, one may think in terms of "jobs" that arrive over time, each job consisting of particular "tasks" that are to be executed sequentially at specified work centers. Except for computer simulation, no satisfactory methods exist for analysis of processing systems that allow the *simultaneous* performance of tasks associated with a single job or customer. Nevertheless, parallel processing is important in many areas of application, including manufacturing systems, product development and parallel computing. Hereafter, the term *processing networks* will be used when referring to this larger class of systems, as distinct from the more familiar and restrictive class of queueing network models, where sequential processing is assumed. In this paper we analyze a class of processing networks called *fork-join networks*. Loosely speaking, such a network processes a sequence of statistically identical and independent "jobs." Each job consists of a fixed number of tasks whose order of execution is constrained by certain deterministic precedence requirements. We assume that there is a "server" or "processing

Received July 1991; revised December 1991.

AMS 1991 *subject classifications*. 60K25, 60J65, 60K20.

Key words and phrases. Fork-join networks, processing networks, heavy traffic analysis, reflected Brownian motion, sojourn time analysis, performance analysis.

station" dedicated to each task, and tasks compete for resources at each processing station in a first-in-first-out (FIFO) manner.

The distinguishing features of this model class are the so-called "fork" and "join" constructs. A *fork* occurs whenever several tasks are allowed to begin processing at the same time. In the network model, this is represented by a "splitting" of the job into multiple tasks, which are then sent simultaneously to their respective servers. A *join* node, on the other hand, corresponds to a task that may not be initiated until several other tasks have been completed. Components are joined only if they correspond to the same job; thus a join is always preceded by a fork. If the last stage of operation consists of multiple tasks, then these tasks regroup into a single job before departing from the system.

The dependencies created by the fork and join constructs make this class of network models highly intractable [2]. In light of this situation, we propose an approximation scheme that is motivated by heavy traffic theory. We show that in the heavy traffic limit, the vector of total job count processes converges weakly to a *reflected Brownian motion* (RBM) whose dimension is precisely the dimension of the network, namely, the number of processing stations. Furthermore, the weak limits of queue lengths, workload levels, and throughput times are shown to be simple transformations of the RBM. Previously, when reflected Brownian motions have arisen as heavy traffic limits of queueing networks, the state space of the RBM was a nonnegative orthant, but in this application to processing networks, the state space is a *nonsimple polyhedral cone* within the nonnegative orthant.

1.1. *Some applications.* Let us now consider two examples that illustrate the different types of systems that one can model by a fork-join network. The first is an assembly center which makes a particular product. Each job that enters the system may be thought of as a *kit* of parts; after the arrival of a kit the individual parts are routed to subassembly stations. When all the preliminary work (which may include several stages of minor assembly on some of the parts) has been completed, the part undergoes final assembly and the finished product exits from the system. Suppose that there are as many "processing resources" as there are operations. One can imagine that the individual parts of each job are "routed" from one service center to the next as they are being processed and assembled. As an example one can think of a production shop of a coat manufacturer; Figure 1 (cf. page 77, [29]) illustrates the main operations involved in the making of a coat. Note that a fork in the system corresponds to a physical splitting of a job (in this case the parts of a coat), and a join represents a physical assembly of parts.

Next, consider a builder of tract housing. An arriving "job" here corresponds to an authorization or a request to build another house. Suppose that all houses in the tract are similar enough to be regarded as identical. The building of each house requires the completion of numerous tasks which are subject to certain precedence constraints; Figure 2 (cf. page 311, [18]) gives an

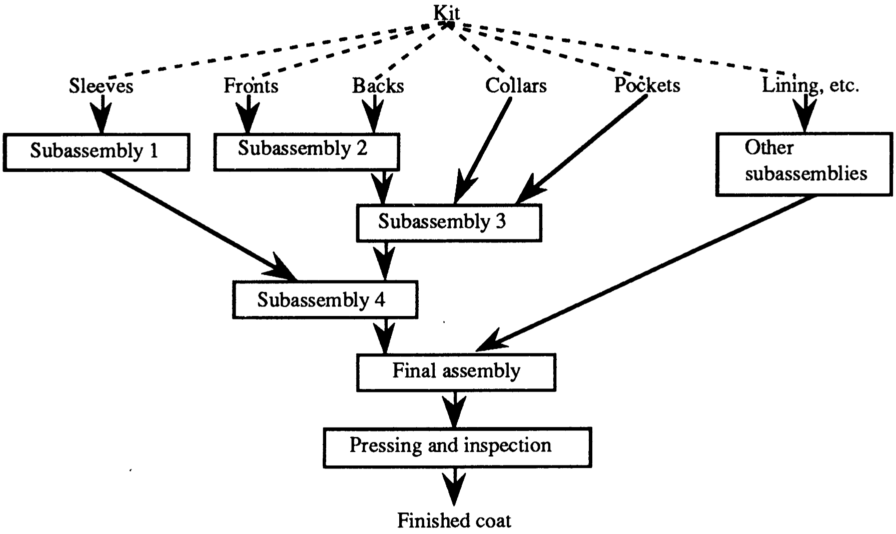

FIG. 1. *An assembly system: a coat shop.*

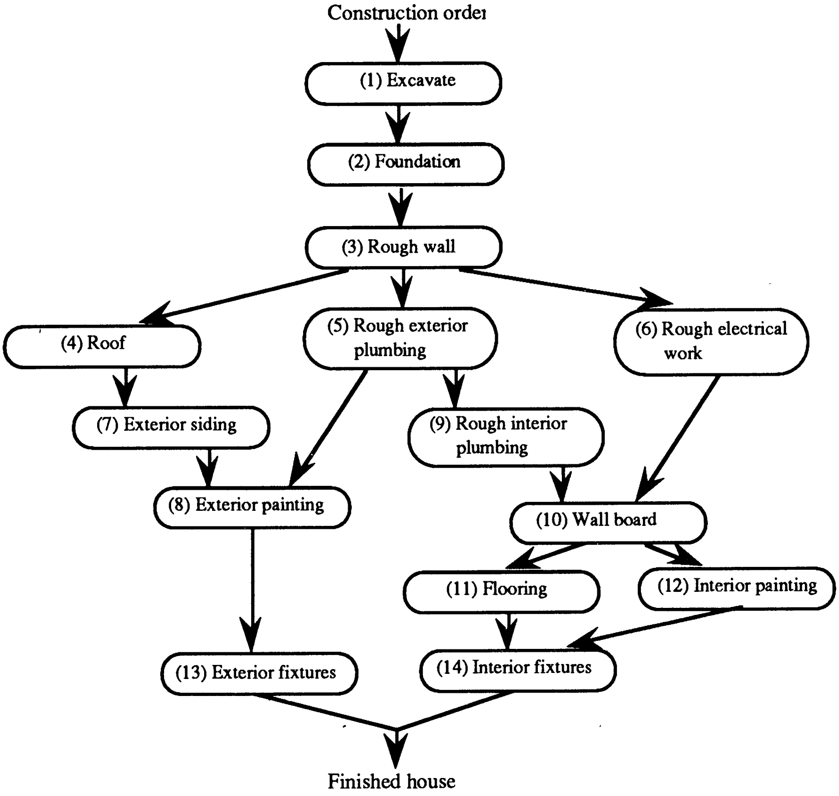

FIG. 2. *Construction of a house: the task graph.*

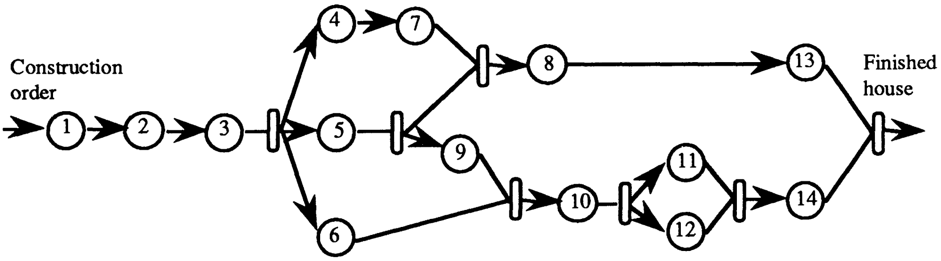

FIG. 3. *Construction of a house: the network.*

example of the tasks involved in building a house and the associated precedence requirements.

Suppose that the builder has one crew of workers dedicated to each of these tasks. Rather than picturing the crews as moving between work sites, one can consider the logically equivalent scenario in which crews remain stationary and houses "travel" from one crew to the next to receive service. Hence the task graph in Figure 2 can be represented by the network shown in Figure 3, and one may think of a job, or a house that is being constructed, as working itself through the network in the order shown. Once the house is completed, that is, all necessary tasks have been performed, it "departs from the network." In this example there is no physical splitting of a job or assembly of components. A fork in this context is used to enable simultaneity of operations, whereas a join requires the completion of several tasks before a successor operation can be undertaken. In this sense the fork and join constructs are only logical operators.

For a survey of other interesting applications readers can refer to the papers by Baccelli and Makowski [2] and Mandelbaum and Avi-Itzhak [22].

1.2. *A brief literature survey.* Baccelli and Liu [1], Baccelli and Makowski [2], Baccelli, Makowski and Shwartz [3] and Baccelli, Massey and Towsley [4] have established stability conditions for a class of fork-join networks that subsumes the models considered in this paper. In the special case where the interarrival and service times are mutually independent sequences of independent, identically distributed (i.i.d.) random variables, with mean interarrival time λ^{-1} and mean service time τ_j at station j, their condition reduces to $\lambda\tau_j < 1$, $j = 1, \dots, J$; this is the usual stability condition associated with conventional queueing networks.

It is known that fork-join networks exhibit stationary behavior, but the specifics of that behavior are not understood. The simplest example of a fork-join network is the *fork-join queue* (see Figure 4). In this system an arriving job immediately forks into K tasks, which are to be served simultaneously; the job is completed when all K tasks have been finished. Even for this simple class of network models, few analytical results are available (see, e.g., [9, 10, 21, 22, 23]). Due to the intractability of fork-join networks, several

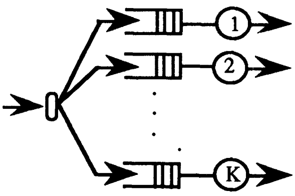

FIG. 4. *A fork-join queue.*

efforts have been made to find reasonable bounds or approximations for such systems. Nelson and Tantawi [23] have proposed an approximation for the throughput time of the fork-join queue with Poisson arrivals and exponentially distributed service times at each queue. The formula proposed in [23] derives from a combination of theoretical results and empirical data. The works by Baccelli and Liu [1], Baccelli and Makowski [2], Baccelli, Makowski and Shwartz [3] and Baccelli, Massey and Towsley [4] have been directed at finding bounds for the throughput time of fork-join networks. The bounds were obtained using results from stochastic ordering and the theory of associated random variables. The tightness of these bounds, however, has not been investigated.

One approach that has proved effective for studying otherwise intractable queueing networks is to analyze these systems in the so-called "heavy traffic limit." In this approach various processes associated with the conventional queueing network are shown to converge (after an appropriate scaling) to a *diffusion process* as the traffic intensity at each station in the network approaches one. For open queueing networks, certain processes such as queue length and workload level have been shown to converge to a multidimensional diffusion process called a *reflected Brownian motion* (*RBM*) *in an orthant.* The interested reader is referred to Harrison and Reiman [15] for a rigorous definition of RBM in an orthant.

Although the method of heavy traffic analysis has been applied successfully to conventional open queueing networks [8, 14, 25, 27, 28], not much progress has been reported regarding its applicability to networks with fork and join contructs [12]. A recent work by Varma [30] offered a heavy traffic analysis of fork-join networks. Varma was able to show weak convergence for certain processes of interest; however, he was unable to characterize the limiting processes in a useful way. His results reveal that the limiting processes can be expressed via complex functions of a multidimensional Brownian motion, but these mappings are never explicitly defined.

Although several of the results discussed here were also obtained independently by Varma, an important contribution of this paper, which does not appear in Varma's work, is the characterization of various limiting processes in terms of a certain reflected Brownian motion whose state space is a polyhedral cone in the nonnegative orthant. In particular, the total job count process is characterized in the limit as a J-dimensional RBM, where J is the number of processing stations in the network. In addition, the limiting workload process is shown to be a simple transformation of the job count process; this transformation can be interpreted naturally in terms of the original processing network. Finally, the heavy traffic limit of the throughput time process is expressed in terms of workload levels and processing times via the "longest path functional" of classical PERT/CPM analysis.

As the results in Nguyen [24] indicate, the characterization of the limiting processes as functions of a multidimensional RBM enables steady-state analysis of these processes. Conditions can be established for the limiting processes to have stationary distributions. Furthermore, algorithms can be developed that numerically solve for these distributions, from which one can obtain approximate performance measures. The success of the characterization relies heavily on identifying the correct representations for the processes of interest, and the next two sections will be devoted to defining these processes and developing the representations.

1.3. *Preliminaries.* The results obtained in this paper, which involve convergence of normalized stochastic processes to a limit process, derive from the theory of weak convergence of probability measures on metric spaces. In the setting of this paper, the metric space is $\mathbf{D}^r[0,1]$, the r-dimensional product space of right continuous functions on $[0,1]$ that have left limits, endowed with the Skorohod topology. We denote by ρ and d the uniform metric and Skorohod metric, respectively. The choice of $[0,1]$ as a time interval is merely a matter of standardization; furthermore, extension to the half line $[0,\infty)$ can be easily handled [31]. Throughout the paper weak convergence is denoted by $\Rightarrow$. The standard reference for weak convergence is Billingsley [6].

Let $\mathbf{D}_+^r$ denote the set of $x \in \mathbf{D}^r$ which satisfy $x(0) \in \Re_+^r$. We will denote by $\mathbf{D}_0$ the set of elements ϕ in $\mathbf{D}$ that are nondecreasing and satisfy $0 \le \phi(t) \le 1$; that is, $\mathbf{D}_0$ is the set of time transformations on $[0,1]$. Next, for h a measurable mapping, let D_h be the set of discontinuities of h.

In each of our heavy traffic limit theorems, the weak limit obtained is a reflected Brownian motion whose state space is a polyhedral cone in the nonnegative orthant. A brief explanation of how such a process behaves will be given in Section 5. For a more detailed description of multidimensional RBM see the works by Harrison and Reiman [15] and Harrison and Williams [16, 17]. For the special case of RBM's of the form considered in this paper, readers should refer to Nguyen [24]. A Brownian motion process having drift vector θ and covariance matrix Γ will be denoted as (θ, Γ)BM; likewise, a reflected Brownian motion with these drift and covariance parameters, reflection matrix R, and state space is S is denoted as (S, θ, Γ, R)RBM.

We end this section with a few words regarding notation. We will not distinguish between the writing of vectors and scalars. Unless otherwise stated, a vector is always taken to be a column vector. The letter e is used to denote the vector whose components are all 1's (the dimension of the vector should be clear from context). In addition, $\operatorname{diag}(x_1, \dots, x_m)$ is used to represent an $m \times m$ diagonal matrix whose diagonal elements are $x_1, \dots, x_m$. Finally, for x a real number, $\lfloor x \rfloor$ denotes the integer part of x.

2. Model definition. This paper is concerned with the analysis of a homogeneous fork-join network in which jobs are routed in a feedforward, deterministic fashion. The network is composed of J single-server stations, indexed by $j = 1, \dots, J$. We assume that each server works at a constant rate of 1. The network has an input stream of homogeneous jobs and we denote by λ the average arrival rate of new jobs. The processing of each job requires the completion of J tasks. Each task is performed at a specific single-server processing station, and the task performed at station j is referred to as task j. The mean service duration at station j, or equivalently, the mean completion time of task j, is τ_j. To rigorously prove our heavy traffic limit theorems, some additional distributional assumptions are required for the interarrival and service times. These assumptions will be stated in Section 4.

The order in which tasks are performed is specified by a given set of precedence constraints, which may allow some tasks to be performed in parallel and may require that others be performed sequentially. Task i is said to be an *immediate predecessor* for task j if upon completing task i the job moves immediately to station j. If such is the case, we also call j an *immediate successor* of task i. The precedence relationships can be expressed via a precedence matrix $\mathsf{P} = (\mathsf{P}_{ij})$ defined as follows:

$$(2.1) \qquad \mathsf{P}_{ij} = \begin{cases} 1, & \text{if task } i \text{ is an immediate predecessor for task } j, \\ 0, & \text{otherwise.} \end{cases}$$

(Because all elements of the precedence matrix P are 0's and 1's, routing is clearly deterministic.) We assume that there is a column and row permutation of P such that the resulting matrix is strictly upper triangular; in terms of the model, this means that we consider only systems in which tasks are not repeated. With this restriction, one can assume without loss of generality that the numbering of stations is such that if task i cannot be started until task j has been completed, then $i > j$. Thus one can picture a job flowing through the network in a deterministic and feedforward fashion, always moving from lower numbered stations to higher numbered ones.

The set of immediate predecessors to station j, denoted as $\mathscr{P}(j)$, is defined to be the set of stations whose departing jobs proceed directly to station j, that is,

$$\mathscr{P}(j) \equiv \{i \in \{1, \dots, J\} : \mathsf{P}_{ij} = 1\}.$$

Let $p(j)$ be the cardinality of the set $\mathscr{P}(j)$, or in words, $p(j)$ is the number of stations whose output feeds directly into station j. Next let $\mathscr{S}(j)$ be the set of immediate successors of station j, that is,

$$\mathscr{S}(j) \equiv \{i \in \{1, \dots, J\} : \mathsf{P}_{ji} = 1\}.$$

As an example, for the network depicted in Figure 5, $\mathscr{P}(3) = \{1, 2\}$, $p(3) = 2$ and $\mathscr{S}(1) = \{3\}$. It will be useful to think of new arrivals to the network as originating from a "dummy" station 0, and departures from the network as destined for a "dummy" station $J + 1$. With that interpretation in mind let us

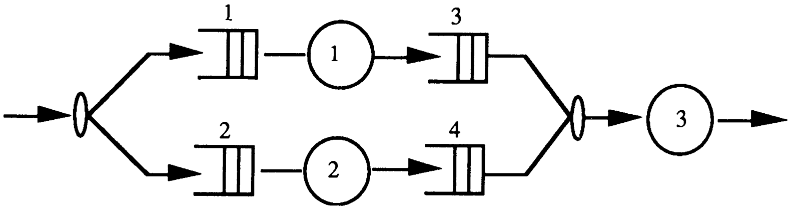

FIG. 5. *A simple fork-join network.*

define

$$\mathscr{S}(0) \equiv \{i \colon \mathscr{P}(i) = \varnothing\},$$

$$\mathscr{P}(J+1) \equiv \{i \colon \mathscr{S}(i) = \varnothing\}$$

and then redefine

$$\mathscr{P}(i) \equiv \{0\} \quad \text{for } i \in \mathscr{S}(0).$$

Thus $\mathscr{S}(0)$ is the collection of stations to which a job proceeds immediately (perhaps after splitting) upon arrival to the network, and $\mathscr{P}(J+1)$ represents the final stage of processing. In Figure 5, $\mathscr{S}(0) = \{1, 2\}$ and $\mathscr{P}(4) = \{3\}$.

It will also be convenient to associate with each station $j = 1, \dots, J$ a set of $p(j)$ different waiting rooms or "buffers." Each buffer associated with station j corresponds to a different immediate predecessor $i \in \mathscr{P}(j)$, and whenever a service is completed at such a station i, one can think of the departing task as entering a buffer corresponding to the pair (i, j). The task remains in the buffer until it can be joined as necessary and then served. The total number of buffers is $K \equiv \Sigma_{j=1}^{J} p(j)$, and rather than indexing buffers by (i, j) pairs, it will be notationally convenient to index them by $k = 1, \dots, K$. For each station $j = 1, \dots, J$ we define $\mathscr{B}(j)$ as the set of buffers k that are incident to j. Thus each buffer $k \in \mathscr{B}(j)$ corresponds to a station $i \in \mathscr{P}(j)$, and tasks departing from station i enter buffer k as a preliminary to service at station j. We let $s(k)$ be the source of buffer k, that is, the station whose output feeds into buffer k [if $k \in \mathscr{S}(0)$, then $s(k) \equiv 0$], and define the destination of buffer k to be $d(k) = j$ if $k \in \mathscr{B}(j)$. For the network in Figure 5, $d(4) = 3$ and $s(4) = 2$.

A station $i \in \{0, 1, \dots, J-1\}$ is said to be a fork node if $\mathscr{S}(i)$ has more than one element, and a station $j \in \{1, \dots, J\}$ is said to be a join node if $\mathscr{P}(j)$ has more than one element (a station may be both a fork node and a join node). At a join node a task is said to be complete or to be a unit if all of its components have entered their respective buffers.

We assume that tasks compete for resources 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. At join nodes, the arrival time of a task is defined to be the arrival time of a complete unit, or equivalently, the arrival time of the last component of the task. Such a service discipline can be characterized as a local policy because it considers only station-level information. The question may arise as to what effect such a policy would have at the system level; namely, does this policy process jobs in the order of their arrivals to the network. Because jobs are generated from one arrival stream and all jobs have the same "route" through the network, the station-level first-come-first-serve discipline does in fact preserve the ordering of the jobs. In other words, tasks never overtake each other, and the policy stated here is equivalent to the global scheme of serving tasks in the order of arrival of the associated jobs.

3. Representations for processes of interest. To state the heavy traffic limit theorem, it is convenient to construct the basic stochastic processes associated with a fork-join network by taking as primitive a collection of "unitized" interarrival and service time random variables. Let $(\Omega, \mathscr{F}, P)$ be a probability space on which are defined sequences of random variables $\{u(i), i \geq 1\}$ and $\{v_j(i), i \geq 1\}$, $j = 1, \dots, J$, where $u(i)$ and $v_j(i)$ 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 $J + 1$ 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 $J + 1$ sequences are mutually independent. From these sequences, the interarrival times and the service times for the network are constructed by setting the interarrival time for the ith job to be $\lambda^{-1}u(i)$, and its service time at station j to be $\tau_j v_j(i)$. Recall that λ is the average arrival rate for new jobs and τ_j is the mean service time at station j.

To construct the *external arrival process* generated by the normalized interarrival times $\{u(i)\colon i \geq 1\}$, set $u(0) \equiv 0$ and define

$$N(t) \equiv \max\left\{k\colon \sum_{i=0}^{k} \lambda^{-1}u(i) \leq t\right\}.$$

For $j = 1, \dots, J$, let $V_j(t)$ be the partial sums process associated with the service times at station j,

$$V_j(t) \equiv \sum_{i=1}^{[t]} \tau_j v_j(i)$$

and set

$$L_j(t) \equiv V_j(N(t)) = \tau_j v_j(1) + \cdots + \tau_j v_j(N(t)). \tag{3.1}$$

The process $L_j(t)$ is called the *total workload input process* for station j; it is the sum of all service times at station j for customers or jobs that enter the network during $[0, t]$. Note that $L_j(t)$ includes service times corresponding to services that may not be completed, or perhaps even started, until after time t. Set

$$\xi_j(t) = L_j(t) - t; \tag{3.2}$$

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. The vector processes V, L and ξ are then defined in the natural way.

Let us choose an "external" station $j \in \mathscr{S}(0)$, fixing j until further notice. Set $\mathbf{A}_j(t) \equiv N(t)$ and for $k \in \mathscr{B}(j)$ let $A_k(t) \equiv N(t)$. [In this case $\mathscr{B}(j)$ has exactly one element.] Next let $M_j(t) \equiv L_j(t)$ and $X_j(t) \equiv \xi_j(t)$. Because external arrivals enter station j, $\mathbf{A}_j(t)$ is the number of jobs that have actually arrived to station j by time t, and $A_k(t)$ may be interpreted as the number of tasks that have entered buffer k by time t. Similarly $M_j(t)$ is the amount of work that has arrived at station j in $[0, t]$ and $X_j(t)$ is the corresponding netflow process.

We now claim that there exist unique processes W_j and I_j which simultaneously satisfy the following three properties:

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

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

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

That (3.3)–(3.5) uniquely define W_j and I_j comes from Section 2.2 of [13]; furthermore, I_j is given by

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

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 all jobs that are present in buffers incident to j at time t, plus the remaining service time of any task that may be in service. These representations of unfinished work and idleness are standard in the literature of queueing theory (cf. Benes [5]).

To define the counting process associated with the sequence of service times at station j, set $v_j(0) \equiv 0$ and define

$$S_j(t) \equiv \max\left\{k \colon \sum_{i=0}^{k} \tau_j v_j(i) \leq t\right\}.$$

Let $B_j(t) \equiv t - I_j(t)$ be the amount of time server j spent working in $[0, t]$, and define

$$D_j(t) \equiv S_j\big(B_j(t)\big), \tag{3.6}$$

$$Q_k(t) \equiv A_k(t) - D_j(t) \geq 0. \tag{3.7}$$

Equation (3.6) identifies $D_j(t)$ as the number of services completed at station j in $[0, t]$, or equivalently, the number of tasks that have departed from the station by time t. It then follows from the previous interpretations that $Q_k(t)$

represents the queue length process at buffer k at time t (the number of tasks waiting in buffer k plus any job being serviced at station j).

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 $i \in \mathscr{P}(j)$, then for each $l \in \mathscr{B}(i)$ the processes $X_i(t)$, $W_i(t)$, $I_i(t)$, $D_i(t)$ and $Q_l(t)$ have been defined. For such a station j and each buffer $k \in \mathscr{B}(j)$ one defines the arrival process to be the departure process from its source,

$$A_k(t) \equiv D_{s(k)}(t). \tag{3.8}$$

Next, set

$$\mathbf{A}_j(t) = \min_{k \in \mathscr{B}(j)} A_k(t).$$

One can interpret $\mathbf{A}_j(t)$ as the number of complete jobs or "units" that have arrived to station j by time t. We take the convention that work is associated with job *units* so that incomplete jobs present no work to the server. With such an accounting procedure the *immediate workload input process* and *immediate netflow process* for station j are defined, respectively, via

$$M_j(t) \equiv V_j(\mathbf{A}_j(t)) = \tau_j\left[v_j(1) + \cdots + v_j(\mathbf{A}_j(t))\right]$$

and

$$X_j(t) \equiv M_j(t) - t.$$

The workload process W_j, the idleness process I_j, the departure process D_j and the queue length processes Q_k are then defined exactly as in (3.3)–(3.7). The vector processes A, $\mathbf{A}$, X, W, I, Q and D are then defined in the obvious manner.

Let

$$U(t) \equiv \xi(t) + I(t). \tag{3.9}$$

The process $U_j(t)$ represents the amount of unfinished work destined for station j that is present anywhere in the system at time t. In particular, $U_j(t)$ may contain work corresponding to jobs which at time t are still queued at stations preceding j. Hence, U_j could alternatively be called the total workload process for station j. Similarly, let

$$Z(t) \equiv eN(t) - D(t), \tag{3.10}$$

where e is the vector of 1's. The process $Z_j(t)$ represents the total number of jobs in the system at time t that still need service at station j, and is called the total job count process. Suppose that $i \in \mathscr{P}(j)$ is a station preceding j, and k is the buffer corresponding to the (i, j) pair [i.e., $s(k) = i$]. It is clear that the total job count for station j equals the sum of the total job count for station i and the number of tasks found in buffer k. In fact, it can be shown from (3.7) and (3.8) that for each $k \in \mathscr{B}(j)$,

$$Z_j(t) = Q_k(t) + Z_{s(k)}(t). \tag{3.11}$$

The final task at hand is to define the throughput time, or sojourn time, of a job, which is the length of time between the job's arrival and its subsequent departure from the system. Note that this time interval includes both waiting times and service times. Let $T(t)$ be the throughput time of the next job to enter the network after time t. A formal definition will be developed via inductive definition of intermediate processes $T_1(t), \dots, T_J(t)$, where $T_j(t)$ is interpreted as the throughput time through station j, which is the time interval between the arrival epoch of a job and when it completes service at station j.

Let $\Phi(t)$ be the random process defined by

$$\Phi(t) \equiv \lambda^{-1}u(1) + \cdots + \lambda^{-1}u(N(t)) + \lambda^{-1}u(N(t)+1).$$

One interprets $\Phi(t)$ as the arrival epoch of the next job to enter the network after time t. For each station $j \in \mathscr{S}(0)$, let

$$\Phi_j(t) \equiv \Phi(t),$$

$$T_j(t) \equiv W_j\big(\Phi_j(t)\big).$$

Because station $j \in \mathscr{S}(0)$ is among the first stations to be visited, $\Phi_j(t)$ is the arrival time of this job to station j. Furthermore, because jobs are served in a first-in-first-out manner, the amount of time this job must spend at station j is precisely the amount of work found at station j immediately after arrival (which includes the service time associated with the new arrival). Thus $T_j(t)$ is the total sojourn time of the job through station j.

For other stations in the network, the random processes $\Phi_j(t)$ and $T_j(t)$ are inductively defined as follows. Suppose that j is a station such that $T_i(t)$ has been defined for each $i \in \mathscr{P}(j)$, and set

$$\Phi_j(t) \equiv \Phi(t) + \max_{i \in \mathscr{P}(j)} T_i(T), \tag{3.12}$$

$$T_j(t) \equiv \max_{i \in \mathscr{P}(j)} T_i(t) + W_j\big(\Phi_j(t)\big). \tag{3.13}$$

Recall that the arrival time of a job is taken to be the time at which its last component arrives. (If j were a join node, there could be a gap between arrival times of the various components of the job.) Thus, $\max_{i \in \mathscr{P}(j)} T_i(t)$ is the amount of time that elapses until the job "arrives" at station j and $\Phi_j(t)$ is precisely its time of arrival. Hence $T_j(t)$ corresponds to the throughput time through station j. Setting

$$T(t) \equiv \max_{j \in \mathscr{P}(J+1)} \big\{T_j(t)\big\},$$

one can conclude that $T(t)$ is the total sojourn time corresponding to the next job to enter the system after time t.

4. A sequence of systems in heavy traffic. The analytical methods developed here apply to systems that satisfy conditions of "heavy traffic." The *traffic intensity* at station j is defined to be

$$\rho_j \equiv \lambda\tau_j. \tag{4.1}$$

The system is said to be *stable* if $\rho_j < 1$ for $j = 1, \dots, J$, 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 $\{u(i)\colon i \geq 1\}$, $\{v_j(i)\colon i \geq 1\}$, $j = 1, \dots, J$. To construct a sequence of fork-join networks we further require sequences of positive constants $\{\lambda^{(n)}, n \geq 1\}$, $\{\tau_j^{(n)}, n \geq 1\}$, $j = 1, \dots, J$. In the nth system of the sequence, the interarrival times and service times are taken to be $u^{(n)}(i) \equiv u(i)/\lambda^{(n)}$ and $v_j^{(n)}(i) \equiv \tau_j^{(n)} v_j(i)$, respectively. Thus for the nth system, $\lambda^{(n)}$ is the arrival rate of new jobs and $\tau_j^{(n)}$ is the mean service time at station j. Define the traffic intensities $\rho_j^{(n)}$ as in (4.1) using $\lambda^{(n)}$ and $\tau_j^{(n)}$ in place of λ and τ_j.

The convention here is to denote a parameter or a process associated with the nth system by the superscript "(n)." For example, $N^{(n)}$ refers to the external arrival process in the nth system. For $j = 1, \dots, J$ and $k = 1, \dots, K$, define the centered processes

$$\begin{aligned} \hat{N}^{(n)}(t) &\equiv N^{(n)}(t) - \lambda^{(n)}t, & \hat{V}_j^{(n)}(t) &\equiv V_j^{(n)}(\lfloor t \rfloor) - \tau_j^{(n)}\lfloor t \rfloor, \\ \hat{A}_k^{(n)}(t) &\equiv A_k^{(n)}(t) - \lambda^{(n)}t, & \hat{\mathbf{A}}_j^{(n)}(t) &\equiv \mathbf{A}_j^{(n)}(t) - \lambda^{(n)}t, \\ \hat{L}_j^{(n)}(t) &\equiv L_j^{(n)}(t) - \rho_j^{(n)}t, & \hat{S}_j^{(n)}(t) &\equiv S_j^{(n)}(t) - \left(\tau_j^{(n)}\right)^{-1}t. \end{aligned}$$

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

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

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,

$$\lambda^{(n)} \to \lambda \quad \text{and} \quad \tau_j^{(n)} \to \tau_j, \qquad j = 1, \dots, J.$$

This implies that $\rho_j^{(n)} \to \rho_j = \lambda_j \tau_j$ for all j. Furthermore, it is assumed that there exists a J-vector $\theta = (\theta_1, \dots, \theta_J)$ such that for each $j = 1, \dots, J$, $-\infty < \theta_j < \infty$ and

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

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 $J \times J$ covariance matrix Γ such that the following functional central limit theorem holds as $n \to \infty$:

(4.3) $(\hat{N}^n, \hat{V}^n, \hat{L}^n) \Rightarrow (N^*, V^*, L^*)$, where L^* is a $(0, \Gamma)$ Brownian motion and N^*, V^* are also Brownian motions with zero drift.

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

$$\xi_j^n(t) = L_j^n(t) + n^{1/2}\big(\rho_j^{(n)} - 1\big)t.$$

One can conclude from assumptions (4.2) and (4.3) that

$$(\hat{N}^n, \xi^n) \quad \Rightarrow \quad (N^*, \xi^*), \quad \text{where } \xi^* \text{ is } (\theta, \Gamma)\text{BM}. \tag{4.4}$$

Moreover, it follows from (3.1), (3.2) and assumption (4.3) that

$$\xi_j^*(t) = \hat{V}_j^*(\lambda t) + \tau_j N_j^*(t) + \theta_j t. \tag{4.5}$$

Next, recall that S_j is the counting process associated with the partial sums process V_j. From Theorem 1 of [19], (4.3) implies that

$$\hat{S}^n \quad \Rightarrow \quad S^*, \quad \text{where } V^*(t) = -\tau^{3/2} S^*(t). \tag{4.6}$$

Finally, consider the special case in which $\{u(i),\ i \geq 1\}$ and $\{v_j(i),\ i \geq 1\}$, $j = 1, \dots, J$, are mutually independent sequences of i.i.d. random variables such that $u(i)$ and $v_j(i)$ have squared coefficients of variation c_a^2 and c_{sj}^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 $N^{(n)}$ is a renewal process with rate $\lambda^{(n)}$, and a simple application of the functional central limit theorem for renewal processes [6] proves that $\hat{N}^n \Rightarrow N^*$, where N^* is $(0, \lambda c_a^2)$BM. Because $L_j^{(n)}$ is a compound renewal process, $\hat{L}^n$ converges to $(0, \Gamma)$BM by Theorem 2.1 of [32]. In particular, the covariance matrix is of the form

$$\Gamma_{ij} = \begin{cases} \lambda \tau_j^2\big(c_{sj}^2 + c_a^2\big), & i = j, \\ \lambda \tau_i \tau_j c_a^2, & i \neq j. \end{cases} \tag{4.7}$$

To summarize, the assumptions needed for the analysis are of two types. First, the heavy traffic condition (4.2) must hold. Second, the sequences of interarrival times and services times must jointly satisfy a functional central limit theorem. The latter assumption is a fairly weak constraint on the distributions of interarrival and service times. Convergence can be shown to hold even when these stochastic quantities are not i.i.d. Several generalizations have been made to the basic functional central limit theorems mentioned here, with the aim of allowing some dependencies and perhaps mild nonstationary in the quantities of interest. For a more detailed discussion of the available extensions, we refer the reader to Glynn [11].

5. The heavy traffic limit theorem. We now state our main results, which say that the limits of various scaled stochastic processes associated with the fork-join network can all be described in terms of a certain reflected Brownian motion whose state space is a polyhedral cone in the nonnegative orthant.

Recall that $d(k)$ denotes the station corresponding to the destination of buffer k, while $s(k)$ is the station corresponding to its source. Let A be a $K \times J$ matrix whose elements A_{kj} ($k = 1, \dots, K$ and $j = 1, \dots, J$) are given by

$$A_{kj} = \begin{cases} 1, & \text{if } j = d(k), \\ -1, & \text{if } j = s(k), \\ 0, & \text{otherwise}. \end{cases} \tag{5.1}$$

Using the matrix A, we now define S to be the following polyhedral cone in J-dimensional space:

$$S = \{x \in \Re^J \colon Ax \geq 0\}. \tag{5.2}$$

It can be verified that S is contained in the nonnegative orthant, and that the cone has a total of K distinct faces. Define the kth face of S to be

$$F_k = \{x \in S \colon A_k x = 0\},$$

where A_k is the kth row of the matrix A.

THEOREM 5.1. *Suppose that assumptions* (4.2) *and* (4.3) *hold. Then*

$$(\xi^n, U^n, Z^n, Q^n, W^n, I^n) \quad \Rightarrow \quad (\xi^*, U^*, Z^*, Q^*, W^*, I^*),$$

where for each $j = 1, \dots, J$ *and* $k \in \mathscr{B}(j)$:

$$\xi^* \text{ is a } (\theta, \Gamma) \text{ Brownian motion}; \tag{5.3}$$

$$U^* = \xi^* + I^*; \tag{5.4}$$

$$\tau_j Z_j^* = U_j^*; \tag{5.5}$$

$$Q_k^* = Z_j^* - Z_{s(k)}^*, \qquad Z_0^*(t) \equiv 0; \tag{5.6}$$

$$Q_k^*(t) \geq 0 \quad \text{for all } t \geq 0; \tag{5.7}$$

$$W_j^* = \tau_j \Big\{ \min_{k \in \mathscr{B}(j)} Q_k^* \Big\}; \tag{5.8}$$

(5.9) $I_j^*(\cdot)$ *is continuous and nondecreasing with* $I_j^*(0) = 0$; *and*

$$\begin{aligned} &I_j^*(\cdot) \text{ increases only at times } t \text{ when} \\ &Q_k^*(t) = 0 \text{ for at least one } k \in \mathscr{B}(j). \end{aligned} \tag{5.10}$$

Observe that properties (5.4), (5.6), (5.7) and (5.9) simply restate (3.9), (3.11), (3.7) and (3.4), respectively. In other words, those characterizations of the limit processes are "exact" in the setting of the original model. Property (5.10) is also a restatement of (3.5). One can interpret the workload process

$W_j(t)$ as the amount of work associated with "complete" jobs at station j. Hence, server j has no work $[W_j(t) = 0]$ if one of the incident buffers to station j has no customers, that is, $Q_k(t) = 0$ for some $k \in \mathscr{B}(j)$.

Property (5.3) is a direct result of assumptions (4.2) and (4.3), as demonstrated in (4.4). Properties (5.5) and (5.8) are approximations of the original model and may be interpreted as a law-of-large-number effect. One can think of τ_j as the long-run average amount of work associated with each job of server j. In heavy traffic the number of jobs waiting to be served tends to infinity so one would expect the average relationship to hold under some appropriate scaling. It turns out that in the heavy traffic scaling, this long-run average is in fact observed at every time point, as demonstrated by (5.5) and (5.8).

Using the characterization of Q_k^* in (5.6), property (5.10) may be equivalently stated as

> $I_j^*(\cdot)$ increases only at times t when $A_k Z^*(t) = 0$ for at least one $k \in \mathscr{B}(j)$.

Coupling the above statement with (5.4), (5.5) and (5.9), one has the following set of properties for the vector of total job count processes Z^*:

$$Z_j^* = \tau_j^{-1}(\xi_j^* + I_j^*), \tag{5.11}$$

$$I_j^*(\cdot) \text{ is continuous and nondecreasing with } I_j^*(0) = 0, \tag{5.12}$$

$$\begin{aligned} &I_j^*(\cdot) \text{ increases only at times } t \text{ when} \\ &A_k Z^*(t) = 0 \text{ for at least one } k \in \mathscr{B}(j). \end{aligned} \tag{5.13}$$

Setting

$$R = \operatorname{diag}(\tau_1^{-1}, \dots, \tau_J^{-1}), \qquad \mu = R\theta \quad \text{and} \quad \Omega = R\Gamma R', \tag{5.14}$$

statements (5.3) and (5.11)–(5.13) collectively identify Z^* as a J-dimensional (S, μ, Ω, R)RBM.

A rigorous treatment of the RBM can be found in [24]; we only briefly describe its behavior here. The RBM Z^* is constrained to lie in the state space S. In the interior of the polyhedral cone S, Z^* behaves as a J-dimensional Brownian motion with drift vector μ and covariance matrix Ω. When the process hits a boundary face F_k, $k \in \mathscr{B}(j)$, it is instantaneously "pushed back" into the interior of S in the direction R^j, the jth column of R. One can think of the behavior at the boundary as an instantaneous "displacement" in a direction characteristic of the boundary surface struck. The amount of pushing is the minimum amount needed in order to keep the process inside the state space S.

In previous analyses of traditional queueing networks, the natural state space of the limiting processes is the nonnegative orthant. By linear transformations, the class of limiting processes is equivalent to the class of RBM's in simple polyhedral domains. (A d-dimensional polyhedron is simple if at most d of its faces intersect.) The RBM's that arise in this study are novel in that the corresponding state space is typically a nonsimple polyhedral region (see Section 7 for an example). Specifically, these RBM's cannot be mapped into the well studied class of RBM's in the orthant.

THEOREM 5.2. *Under the assumptions of* (4.2) *and* (4.3),

$$(T^n, T_1^n, \dots, T_J^n) \Rightarrow (T^*, T_1^*, \dots, T_J^*),$$

where

$$T^* = \max_{j \in \mathscr{P}(J+1)} T_j^*,$$

$$T_j^* = \max_{i \in \mathscr{P}(j)} T_i^* + W_j^*, \qquad T_0^* \equiv 0,$$

and W_j^* *is defined as in Theorem* 5.1.

It is important that the proposed method first derives an approximate joint distribution for workload levels at the stations of a fork-join network. Methods that are presently used for approximate analysis of conventional network models, such as Whitt's Queueing Network Analyzer [33], treat congestion levels at the various stations as if they were independent (this is the meaning of the term "decomposition approximation"), and produce as output only estimates of the average workload levels or average customer delays at various stations. This approach is completely unsatisfactory for fork-join networks, where the whole focus is on throughput time. Our ultimate output is an approximation for the entire distribution of throughput times experienced by arriving jobs, taking account of the dependencies that exist among workload levels at various stations, and of the complicated longest-path functional that maps workload levels into total throughput time.

To be more specific, one has the following representation for the throughput time process:

$$T^*(t) = l(W^*(t)), \tag{5.15}$$

where l: $\mathbf{C}^J \to \mathbf{C}$ is the familiar "longest-path functional" associated with critical path analysis of PERT/CPM. This is an example of the "snapshot principle" first enunciated by Reiman [26]. In the heavy traffic scaling, the fluctuation in workload levels is negligible relative to the length of time that a job spends in the system. Hence one can take a "snapshot" of the system at the time of the job's arrival to represent its state during the job's entire sojourn. Thus, *sample path by sample path*, the throughput time is calculated via classical critical path analysis, with sums of workload levels taking the place of (traditional) task times. In fact, expression (5.15) brings it all back to the starting point, with a "new take" on parallel processing: With the correct interpretation of "task times" (which turns out to be waiting times), the analysis of a processing network with parallel and sequential tasks reduces to a sample path by sample path analysis of a PERT/CPM network; the results of this paper provide the underlying probabilistic structures necessary to complete that analysis.

6. Proof of the heavy traffic limit theorem. It will be convenient to assume that the limiting arrival rate satisfies $\lambda < 1$. Because this simply amounts to choosing an appropriate time scale, there is no loss of generality. With this assumption, on the other hand, theorems may be developed in $\mathbf{D}^r[0,1]$ without intermediate rescaling. Furthermore, since the assumption implies $\lambda^{(n)} < 1$ for all sufficiently large n, one can also assume for purposes of establishing limit results as $n \to \infty$ that $\lambda^{(n)} < 1$ for all n. The proofs that will be presented here are modeled after those of Peterson [25].

6.1. *Proof of Theorem* 5.1. The proof is a straightforward exercise in induction with the help of the continuous mapping theorem. The induction is on $\mathsf{d}(j)$, the "depth" of station j, defined in the following way. For $j \in \mathscr{S}(0)$, set $\mathsf{d}(j) \equiv 1$. Next, consider a station j such that $\mathsf{d}(i)$ has been defined for all stations in its predecessor set $\mathscr{P}(j)$. The depth of station j is given by

$$\mathsf{d}(j) \equiv \max\{\mathsf{d}(i) : i \in \mathscr{P}(j)\} + 1.$$

To begin, consider $\mathscr{S}(0)$, the collection of stations with depth 1. Let us choose a station $j \in \mathscr{S}(0)$ and denote by $k \in \mathscr{B}(j)$ its one incident buffer. From Sections 3 and 4, the scaled workload process at station j is defined by the following three statements:

$$W_j^n = \xi_j^n + I_j^n \geq 0, \tag{6.1}$$

$$I_j^n(\cdot) \text{ is increasing and continuous with } I_j^n(0) = 0, \tag{6.2}$$

$$I_j^n(\cdot) \text{ increases only at times } t \text{ when } W_j^n(t) = 0. \tag{6.3}$$

Furthermore, (6.1)–(6.3) and Proposition 2.2.3 in Harrison [13] imply the elegant representations

$$I_j^n(t) = \phi(\xi_j^n)(t) = -\inf_{0 \leq s \leq t} \{\xi_j^n(s)\}^-,$$

$$W_j^n(t) = \psi(\xi_j^n)(t) = \xi_j^n(t) + \phi(\xi_j^n)(t),$$

where both $\phi, \psi \colon \mathbf{D}_+ \to \mathbf{D}$ are continuous mappings under the metric ρ. Let

$$\mathbf{V}_1^n = \left(\xi^n, W_j^n, I_j^n, Q_k^n, Z_j^n, U_j^n, \mathsf{d}(j) = 1, k \in \mathscr{B}(j)\right).$$

PROPOSITION 6.1. $\mathbf{V}_1^n \Rightarrow \mathbf{V}_1^* = (\xi^*, W_j^*, I_j^*, Q_k^*, Z_j^*, U_j^*,\ \mathsf{d}(j) = 1,\ k \in \mathscr{B}(j))$. *Here, ξ^* is a (θ, Γ) Brownian motion. Furthermore, for each j with $\mathsf{d}(j) = 1$ and $k \in \mathscr{B}(j)$,*

$$W_j^* = \xi_j^* + I_j^* \geq 0,$$

$$I_j^*(\cdot) \textit{ is continuous and nondecreasing with } I_j^*(0) = 0,$$

$$I_j^*(\cdot) \textit{ increases only at times } t \textit{ when } W_j^*(t) = 0,$$

$$\tau_j Q_k^* = W_j^*,$$

$$Z_j^* = Q_k^*,$$

$$U_j^* = \tau_j Z_j^*.$$

PROOF. Let $x = (x_1, \ldots, x_J) \in \mathbf{D}_+^J$, $y \in \mathbf{D}$, and consider the mapping defined by

$$h(x,y) = \big(x, y, \psi(x_j), \phi(x_j), \mathsf{d}(j) = 1\big).$$

Since $\phi, \psi \colon \mathbf{D}_+ \to \mathbf{D}$ are continuous mappings under the metric ρ, it follows that $P\{(\xi^*, N^*) \in D_h\} = 0$ and thus by the continuous mapping theorem (Theorem 5.1, [6]),

$$\begin{aligned} h(\xi^n, \hat{N}^n) &= \big(\xi^n, \hat{N}^n, W_j^n, I_j^n, \mathsf{d}(j) = 1\big) \\ \Rightarrow\quad h(\xi^*, N^*) &= \big(\xi^*, N^*, W_j^*, I_j^*, \mathsf{d}(j) = 1\big), \end{aligned} \tag{6.4}$$

where $W_j^* = \psi(X_j^*)$ and $I_j^* = \phi(X_j^*)$.

Now consider the time transformation $Y_j^{(n)}(t) \equiv t + W_j^{(n)}(t)$. Recall that the only "source" of input to station $j \in \mathscr{S}(0)$ is the external arrival stream, so tasks do not need to be joined prior to service at j. This implies that at time $Y_j(t)$ all current work in buffer $k \in \mathscr{B}(j)$ will be processed and all current tasks in the buffer will have departed from the station. Hence $D_j^{(n)}(Y_j^{(n)}(t)) = A_k^{(n)}(t)$ and (3.7) reduces to

$$Q_k^{(n)}\big(Y_j^{(n)}(t)\big) = A_k^{(n)}\big(Y_j^{(n)}(t)\big) - A_k^{(n)}(t). \tag{6.5}$$

In words, (6.5) states that tasks found in buffer k at time $Y_j(t)$ are precisely those tasks that have arrived since time t. Define $\hat{Y}_j^n$ and $\bar{Y}_j^n$ as

$$\hat{Y}_j^n(t) \equiv n^{-1/2}\big(Y_j^{(n)}(nt) - nt\big) = W_j^n(t),$$

$$\bar{Y}_j^n(t) \equiv n^{-1}\big(Y_j^{(n)}(nt)\big) = t + \bar{W}_j^n(t),$$

respectively, where $\bar{W}_j^n(t) = n^{-1}W_j^{(n)}(nt) = n^{-1/2}W_j^n(t)$. Then,

$$\begin{aligned} Q_k^n\big(\bar{Y}_j^n(t)\big) &= n^{-1/2}\Big(A_k^{(n)}\big(Y_j^{(n)}(nt)\big) - A_k^{(n)}(nt)\Big) \\ &= \hat{A}_k^n\big(\bar{Y}_j^n(t)\big) - \hat{A}_k^n(t) + \lambda^{(n)}\hat{Y}_j^n(t). \end{aligned} \tag{6.6}$$

Because $W_j^n \Rightarrow W_j^*$, Theorem 5.5 of [6] implies that $\bar{W}_j^n \Rightarrow \eta$ where $\eta(t) \equiv 0$. Applying the continuous mapping theorem, one arrives at

$$\begin{aligned} &\big(\xi^n, \hat{N}^n, W_j^n, I_j^n, \hat{Y}_j^n, \bar{Y}_j^n, \mathsf{d}(j) = 1\big) \\ &\quad\Rightarrow\quad \big(\xi^*, N^*, W_j^*, I_j^*, Y_j^*, \bar{Y}_j^*, \mathsf{d}(j) = 1\big), \end{aligned} \tag{6.7}$$

where $Y_j^* = W_j^*$ and $\bar{Y}_j^* = \chi$ with $\chi(t) \equiv t$. Recall that for $k \in \mathscr{B}(j)$ and $j \in \mathscr{S}(0)$, $\hat{A}_k^n = \hat{N}^n$. To apply the random time change theorem (Section 17, [6]), let

$$\alpha_j^n(t) = \begin{cases} \bar{Y}_j^n(t), & \text{if } \bar{Y}_j^n(t) \le 1, \\ t, & \text{otherwise}. \end{cases}$$

Then, $\alpha_j^n \Rightarrow \chi$ and consequently $\hat{A}_k^n \circ \alpha_j^n \Rightarrow N^* \circ \chi$. Note that $\hat{A}_k^n \circ \alpha_j^n$ and $\hat{A}_k^n \circ \bar{Y}_j^n$ have the same value at t if $\alpha_j^n(t) = \bar{Y}_j^n(t)$, the probability of which goes to 1 as n tends to infinity. Thus applying the continuous mapping theorem to the first two terms of (6.6), one has

$$\hat{A}_k^n \circ \bar{Y}_j^n - \hat{A}_k^n \quad \Rightarrow \quad N^* \circ \chi - N^* = \eta. \tag{6.8}$$

It then easily follows as a consequence of (6.6)–(6.8) that

$$\begin{aligned} &\left(\xi^n, \hat{N}^n, W_j^n, I_j^n, Q_k^n \circ \bar{Y}_j^n, \mathsf{d}(j) = 1, k \in \mathscr{B}(j)\right) \\ &\quad \Rightarrow \quad \left(\xi^*, N^*, W_j^*, I_j^*, Q_k^*, \mathsf{d}(j) = 1, k \in \mathscr{B}(j)\right), \end{aligned} \tag{6.9}$$

where $Q_k^* = \lambda W_j^*$. In heavy traffic, $1 = \rho_j = \lambda \tau_j$, so the limiting process Q_k^* can also be expressed via $Q_k^* = \tau_j^{-1} W_j^*$.

Since $P\{W_j^* \in C\} = 1$, the inverse random time change theorem (Theorem A.1) may be applied to show that for each $j \in \mathscr{S}(0)$ and $k \in \mathscr{B}(j)$,

$$d\left(Q_k^n, Q_k^n \circ \bar{Y}_j^n\right) \quad \Rightarrow \quad 0.$$

[The fact that $\bar{Y}_j^n$ is not bounded by 1 can be resolved by the same truncation argument as in the proof leading up to (6.8).] Thus, by Theorem 4.1 of [6],

$$Q_k^n \quad \Rightarrow \quad Q_k^*. \tag{6.10}$$

It remains to establish convergence for the total workload and total job count processes. For each $j \in \mathscr{S}(0)$ and $k \in \mathscr{B}(j)$,

$$Z_j^n(t) \equiv N^n(t) - D_j^n(t) = A_k^n(t) - D_j^n(t) = Q_k^n(t)$$

and

$$U_j^n(t) \equiv \xi_j^n(t) + I_j^n(t) = W_j^{(n)}(t).$$

Hence it follows from (6.9) and (6.10) that $Z_j^n \Rightarrow Z_j^*$ and $U_j^n \Rightarrow U_j^*$, where $Z_j^* = Q_k^*$, $U_j^* = W_j^* = \tau_j Z_j^*$, and that the convergence is joint with $(\xi^n, W_j^n, I_j^n, Q_k^n, \mathsf{d}(j) = 1, k \in \mathscr{B}(j))$. Hence the proposition is proved. □

What remains is simply an inductive argument to extend these results to a feedforward fork-join network. Define the composite vector processes

$$\mathbf{V}_d^n = \left(\hat{N}^n, \xi^n, W_j^n, I_j^n, Q_k^n, Z_j^n, U_j^n, \mathsf{d}(j) \le d, k \in \mathscr{B}(j)\right)$$

and

$$\mathbf{V}_d^* = \left(N^*, \xi^*, W_j^*, I_j^*, Q_k^*, Z_j^*, U_j^*, \mathsf{d}(j) \le d, k \in \mathscr{B}(j)\right).$$

Suppose that joint convergence has been established for $\mathbf{V}_d^n$, that is,

$$\mathbf{V}_d^n \quad \Rightarrow \quad \mathbf{V}_d^* \tag{6.11}$$

and the limiting processes $(\xi_j^*, W_j^*, I_j^*, Q_k^*, Z_j^*, U_j^*)$ satisfy the conditions of Theorem 5.1 for all $\mathsf{d}(j) \le d$ and $k \in \mathscr{B}(j)$.

PROPOSITION 6.2. $\mathbf{V}_{d+1}^n \Rightarrow V_{d+1}^*$. *Furthermore, for each j with $\mathsf{d}(j) \le d+1$ and $k \in \mathscr{B}(j)$, the following hold for all $t \ge 0$:*

$$W_j^*(t) = X_j^*(t) + I_j^*(t) \ge 0, \tag{6.12}$$

$$X_j^*(t) = V_j^*(\lambda t) + \tau_j \mathbf{A}_j^*(t) + \theta_j t, \tag{6.13}$$

$$I_j^*(\cdot) \text{ is continuous and increasing with } I_j^*(0) = 0, \tag{6.14}$$

$$I_j^*(\cdot) \text{ increases only at times } t \text{ when } W_j^*(t) = 0, \tag{6.15}$$

$$Q_k^*(t) = A_k^*(t) - S_j^*(t) + \tau_j^{-1} I_j^*(t) + \tau_j^{-1}\theta_j t, \tag{6.16}$$

$$Z_j^*(t) = Q_k^*(t) + Z_{s(k)}^*(t), \tag{6.17}$$

$$U_j^*(t) = \xi_j^*(t) + I_j^*(t), \tag{6.18}$$

where $A_k^* = N^* - Z_{s(k)}^*$, $\mathbf{A}_j^* = \min_{k \in \mathscr{B}(j)} A_k^*$ and ξ^* is a (θ, Γ) Brownian motion.

PROOF. Let us choose a station j of depth $d+1$. Because $\hat{A}_k^n(t) = \hat{N}^n(t) - Z_{s(k)}^n(t)$, $\hat{\mathbf{A}}_j^n(t) \equiv \min_{k \in \mathscr{B}(j)} \hat{A}_k^n(t)$ and $\mathsf{d}(s(k)) \le d$ for each $k \in \mathscr{B}(j)$, one can apply the continuous mapping theorem to the induction hypothesis (6.11) to obtain

$$\begin{aligned} &\left(\mathbf{V}_d^n, \hat{A}_k^n, \hat{\mathbf{A}}_j^n, \mathsf{d}(j) = d+1, k \in \mathscr{B}(j)\right) \\ &\qquad \Rightarrow \left(\mathbf{V}_d^*, A_k^*, \mathbf{A}_j^*, \mathsf{d}(j) = d+1, k \in \mathscr{B}(j)\right), \end{aligned} \tag{6.19}$$

where $A_k^*(t) = N^*(t) - Z_{s(k)}^*(t)$ and $\mathbf{A}_j^*(t) = \min_{k \in \mathscr{B}(j)} A_k^*(t)$. Define $\overline{\mathbf{A}}_j^n \equiv n^{-1}\mathbf{A}_k^{(n)}(nt)$; the scaled immediate netflow process can be written as

$$\begin{aligned} X_j^n(t) &= n^{-1/2}\Big(V_j^{(n)}\big(\mathbf{A}_j^{(n)}(nt)\big) - \tau_j^{(n)}\mathbf{A}_j^{(n)}(nt)\Big) \\ &\qquad + n^{-1/2}\Big(\mathbf{A}_j^{(n)}(nt) - \lambda^{(n)} nt\Big) + n^{1/2}\Big(\rho_j^{(n)} - 1\Big)t \\ &= \Big(\hat{V}_j^n \circ \overline{\mathbf{A}}_j^n\Big)(t) + \tau_j^{(n)}\hat{\mathbf{A}}_j^n(t) + n^{1/2}\Big(\rho_j^{(n)} - 1\Big)t. \end{aligned}$$

From Theorem 5.5 of [6], the convergence of $\hat{\mathbf{A}}_j^n$ in (6.19) implies that $\overline{\mathbf{A}}_j^n \Rightarrow f$, where $f(t) = \lambda t$. The continuous mapping theorem together with the random time change theorem and the joint convergence of (6.19) yield

$$\hat{X}_j^n \quad \Rightarrow \quad X_j^* = (V_j^* \circ f)(t) + \tau_j \mathbf{A}_j^*(t) + \theta_j t, \tag{6.20}$$

where the convergence is understood to be joint with the vector in (6.19) and with all j such that $\mathsf{d}(j) = d + 1$. [Again, the fact that $\overline{\mathbf{A}}_j^n(\cdot)$ may exceed 1 can be handled by a truncation argument as in the proof of Proposition 6.1.]

The scaled workload and idleness processes for station j are expressed as $W_j^n = \psi(X_j^n)$, $I_j^n = \phi(X_j^n)$, and the scaled queue length process at each buffer

$k \in \mathscr{B}(j)$ is given by

$$\begin{aligned} Q_k^n(t) &= n^{-1/2}\big(A_k^{(n)}(nt) - \lambda^{(n)}nt\big) - n^{-1/2}\left(S_j^{(n)}\big(B_j^{(n)}(nt)\big) - \frac{1}{\tau_j^{(n)}}B_j^{(n)}(nt)\right) \\ &\qquad + \frac{1}{\tau_j^{(n)}}n^{-1/2}\big(nt - B_j^{(n)}(nt)\big) + \frac{1}{\tau_j^{(n)}}n^{1/2}\big(\rho_j^{(n)} - 1\big)t \\ &= \hat{A}_k^n(t) - \hat{S}_j^n\big(\bar{B}_j^n(t)\big) + \frac{1}{\tau_j^{(n)}}I_j^n(t) + \frac{1}{\tau_j^{(n)}}n^{1/2}\big(\rho_j^{(n)} - 1\big)t, \end{aligned}$$

where $\bar{B}_j^n(t) \equiv n^{-1}B_j^{(n)}(nt) = t - n^{-1}I_j^{(n)}(nt)$. Because ϕ and ψ are continuous mappings,

$$\big(X_j^n, W_j^n, I_j^n, \mathsf{d}(j) = d+1\big) \quad \Rightarrow \quad \big(X_j^*, W_j^*, I_j^*, \mathsf{d}(j) = d+1\big),$$

where $W_j^* = \psi(X_j^*)$ and $I_j^* = \phi(X_j^*)$. It then follows that $\bar{B}_j^n \Rightarrow \chi$ with $\chi(t) \equiv t$, and again applying the continuous mapping theorem, one obtains $Q_k^n \Rightarrow Q_k^*$ with

$$Q_k^*(t) = A_k^*(t) - S_j^*(t) + \tau_j^{-1}I_j^*(t) + \tau_j^{-1}\theta_j t, \qquad k \in \mathscr{B}(j).$$

It remains to prove the convergence results for the total job count and total workload processes, but this is straightforward because $U_j^n(t) = \xi_j^n(t) + I_j^n(t)$ and $Z_j^n(t) = Q_k^n(t) + Z_{s(k)}^n(t)$ for each $k \in \mathscr{B}(j)$. One can thus use the continuous mapping theorem to conclude that $\mathbf{V}_{d+1}^n \Rightarrow \mathbf{V}_{d+1}^*$, where for each j, $U_j^* = \xi_j^* + I_j^*$ and for each $k \in \mathscr{B}(j)$, $Z_j^* = Q_k^* + Z_{s(k)}^*$. □

Substituting (6.16) in (6.17) and noting that $A_k^*(t) = N^*(t) - Z_{s(k)}^*(t)$, one can express the vector of total headcount processes as

$$\begin{aligned} Z_j^*(t) &= A_k^*(t) - S_j^*(t) + \tau_j^{-1}I_j^*(t) + \tau_j^{-1}\theta_j t + Z_{s(k)}^*(t) \\ &= N^*(t) - S_j^*(t) + \tau_j^{-1}I_j^*(t) + \tau_j^{-1}\theta_j t. \end{aligned} \tag{6.21}$$

Substituting the expression for ξ^* (4.5) into (6.18), one obtains

$$U_j^*(t) = \tau_j N^*(t) + V_j^*(\lambda t) + I_j^*(t) + \theta_j t. \tag{6.22}$$

With the characterization of S^* and V^* given in (4.6), one can conclude from (6.21) and (6.22) that

$$\tau_j Z_j^*(t) = U_j^*(t). \tag{6.23}$$

Finally, note that

$$\begin{aligned} \min_{k \in \mathscr{B}(j)} Q_k^*(t) &= \min_{k \in \mathscr{B}(j)} A_k^*(t) - S_j^*(t) + \tau_j^{-1}I_j^*(t) + \tau_j^{-1}\theta_j t \\ &= \mathbf{A}_k^*(t) - S_j^*(t) + \tau_j^{-1}I_j^*(t) + \tau_j^{-1}\theta_j t. \end{aligned} \tag{6.24}$$

Comparing (6.24) with (6.12) and (6.13) and applying (4.6), one arrives at

$$(6.25)\qquad \tau_j\Big(\min_{k\in\mathscr{B}(j)} Q_k^*(t)\Big) = W_j^*(t).$$

Theorem 5.1 is thus proved. □

6.2. *Proof of Theorem* 5.2. We now turn to the proof of Theorem 5.2, which characterizes the distribution of the limit sojourn time process. Let $\overline{\Phi}^n$ and $\overline{\Phi}_j^n$ be scaled processes defined as $\overline{\Phi}^n(t) \equiv n^{-1}\Phi^{(n)}(nt)$ and $\overline{\Phi}_j^n(t) \equiv n^{-1}\Phi_j^{(n)}(nt)$.

Lemma 6.3. $(W^n, \overline{\Phi}_j, T_j^n,\ j \in \mathscr{S}(0)) \Rightarrow (W^*, \chi_j, T_j^*,\ j \in \mathscr{S}(0))$, *where for each* $j \in \mathscr{S}(0)$, $\chi_j(t) \equiv t$ *and* $T_j^* = W_j^*$.

Proof. Recall that $T_j^n(t) = W_j^n(\overline{\Phi}_j^n(t))$. The process $\Phi^{(n)}(t)$ is bounded by

$$t \le \Phi^{(n)}(t) \le t + \max_{1\le i\le N^{(n)}(t)+1} u^{(n)}(i).$$

Let $r^{(n)}(t) \equiv \max_{1\le i\le N^{(n)}(t)+1} u^{(n)}(i)$ and $\bar{r}^n(t) \equiv n^{-1}r^{(n)}(nt)$. From Lemma 3.3 of [20], $\|\bar{r}^n\| \equiv \sup_{0\le t\le 1}|\bar{r}^n(t)| \Rightarrow 0$. Letting χ be the identity mapping $\chi(t) \equiv t$, it follows that

$$d\big(\overline{\Phi}^n, \chi\big) \le \rho\big(\overline{\Phi}^n, \chi\big) \le \|\bar{r}^n\| \quad \Rightarrow \quad 0,$$

so $\overline{\Phi}^n \Rightarrow \chi$. Since $\Phi_j^{(n)} = \Phi^{(n)}$ for each $j \in \mathscr{S}(0)$ and $\chi_j \equiv \chi$ are constant elements of D, one has the following joint convergence as a simple consequence of Theorem 5.1:

$$(6.26)\qquad \big(W^n, \overline{\Phi}_j^n, j \in \mathscr{S}(0)\big) \quad \Rightarrow \quad \big(W^*, \chi_j, j \in \mathscr{S}(0)\big).$$

Because W^* and χ_j are continuous with probability 1, Theorem 5.5 of [6] applies to give

$$\big(W^n, \overline{\Phi}_j^n, W_j^n \circ \overline{\Phi}_j^n, j \in \mathscr{S}(0)\big) \quad \Rightarrow \quad \big(W^*, \chi_j, W_j^*, j \in \mathscr{S}(0)\big),$$

hence

$$\big(W^n, \overline{\Phi}_j^n, T_j^n, j \in \mathscr{S}(0)\big) \quad \Rightarrow \quad \big(W^*, \chi_j, T_j^*, j \in \mathscr{S}(0)\big),$$

where $T_j^* = W_j^*$. □

By an inductive argument we can now extend these results to the remaining stations in the network. Suppose that for depth level d, one has established

$$\big(W^n, \overline{\Phi}_j^n, T_j^n, \mathsf{d}(j) \le d\big) \quad \Rightarrow \quad \big(W^*, \chi_j, T_j^*, \mathsf{d}(j) \le d\big),$$

where $T_j^* = \max_{i\in\mathscr{P}(j)} T_i^* + W_j^*$. Let us consider stations j with depth $d+1$. For each of these stations, the intermediate sojourn time is given by

$$T_j^n(t) = \max_{i\in\mathscr{P}(j)} T_i^n(t) + W_j^n\big(\overline{\Phi}_j^n(t)\big),$$

where

$$\overline{\Phi}_j^n(t) = \overline{\Phi}^n(t) + n^{-1/2} \max_{i \in \mathscr{P}(j)} T_i^n(t).$$

From the inductive hypothesis $T_i^n \Rightarrow T_i^*$ for $i \in \mathscr{P}(j)$, so it follows from the continuous mapping theorem that $\max_{i \in \mathscr{P}(j)} T_i^n \Rightarrow \max_{i \in \mathscr{P}(j)} T_i^*$. Consequently, $n^{-1/2} \max_{i \in \mathscr{P}(j)} T_i^n \Rightarrow \eta$, where $\eta(t) \equiv 0$, and one has $\overline{\Phi}_j^n \Rightarrow \chi_j$ with $\chi_j(t) \equiv t$. Imitating the proof of Lemma 6.3, one can show that

$$\left(W^n, \overline{\Phi}_j^n, T_j^n, \mathsf{d}(j) \le d + 1\right) \Rightarrow \left(W^*, \chi_j, T_j^*, \mathsf{d}(j) \le d + 1\right),$$

where

$$T_j^* = \max_{i \in \mathscr{P}(j)} T_i^* + W_j^*.$$

Finally, $T^n(t) = \max_{j \in \mathscr{P}(J+1)} T_j^n(t)$, hence it follows from the continuous mapping theorem that $T^n \Rightarrow T^*$, where

$$T^* = \max_{j \in \mathscr{P}(J+1)} T_j^*.$$

7. Approximating a fork-join network with RBM. For practical purposes, the following question is obviously key: *How does one approximate a fork-join network using results from the previous section?* Theorems 5.1 and 5.2 state that for large n the vector of processes (Z^n, W^n, Q^n, T^n) is well approximated by (Z^*, W^*, Q^*, T^*), where Z^* is (S, μ, Ω, R)RBM, and W^*, Q^*, T^* are simple transformations of Z^*.

Take a fork-join network in heavy traffic with J single-server stations; for ease of exposition let us consider only systems whose interarrival times and service times are statistically independent sequences of i.i.d. random variables. As in Section 2, we denote by λ the average arrival rate of new jobs and by c_a^2 the SCV of interarrival times. Processing times at station j have mean τ_j and SCV c_{sj}^2. Precedence requirements for the execution of tasks are expressed in terms of a precedence matrix P. To approximate this fork-join network, choose n such that $n^{1/2}(1 - \rho_j)$ is of order 1 for each processing station j. For example, one may choose $n = 100$ when ρ_j is near 0.09 for each station j. Viewing n as fixed, the scaled processes (Z^n, W^n, Q^n, T^n) can be approximated by the processes (Z^*, W^*, Q^*, T^*). To obtain an approximation for the original processes (Z, W, Q, T) associated with the network model, one simply "undoes" the scaling. The reader can verify that one arrives at (Z^0, W^0, Q^0, T^0) as the appropriate approximation for (Z, W, Q, T), where Z^0 is (S, μ^0, Ω, R)RBM and W^0, Q^0, T^0 are defined in terms of Z^0 as in Theorems 5.1 and 5.2. The parameters of the RBM (S, μ^0, Ω, R), are computed from the data of the network model via

$$\mu^0 = R(e - \rho)$$

(here e denotes the J vector of 1's), and (4.1), (4.7), (5.1), (5.2) and (5.14).

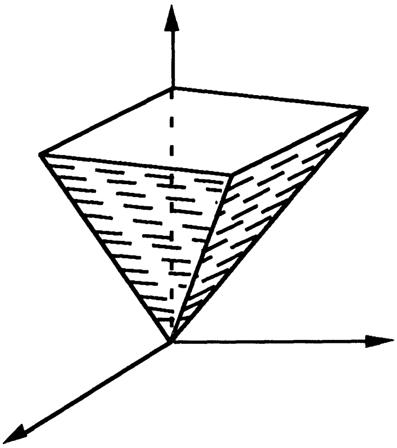

FIG. 6. *A polyhedral cone.*

As an example, consider the network in Figure 5. The parameters of the corresponding RBM are given by $\mu = R(e - \rho)$, $\Omega = R\Gamma R'$, with

$$\rho_j = \lambda\tau_j,$$

$$R = \operatorname{diag}\left(\tau_1^{-1}, \tau_2^{-1}, \tau_3^{-1}\right),$$

$$\Gamma_{ij} = \begin{cases} \lambda\tau_j^2\left(c_{sj}^2 + c_a^2\right), & i = j, \\ \lambda\tau_i\tau_j c_a^2, & i \neq j. \end{cases}$$

The state space associated with this process, shown in Figure 6, is described by $S = \{x \in \mathfrak{R}^3\colon Ax \geq 0\}$ with

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix}.$$

The resulting RBM, like all RBM's that arise as heavy traffic limits of fork-join networks, differ from those associated with heavy traffic limits of traditional queueing networks in that the state space is a nonsimple polyhedral region. It turns out, however, that the extant analytical theory of RBM's in the orthant can be extended to this setting (see Nguyen [24]). Results analogous to those in [16] regarding steady-state characteristics of the RBM can be established. Furthermore, with minor modifications, the numerical method by Dai and Harrison [7] for obtaining steady-state distributions of RBM in the orthant can also be applied to this setting. With this algorithm, one can compute the steady-state distribution of the limiting RBM, from which approximate steady-state performance measures of the processing network can be calculated. Further investigation and clarification of these issues are clearly of prime importance in future research.

APPENDIX

A. The inverse random time change theorem.

THEOREM A.1 (Inverse random time change theorem). *Suppose that* $X_n \in \mathbf{D}$, $\Phi_n \in \mathbf{D}_0$ *and* χ *is the identity function defined by* $\chi(t) = t$. *If* $X_n \circ \Phi_n \Rightarrow X$, $\Phi_n \Rightarrow \chi$ *and* $P\{X \in C\} = 1$, *then* $X_n \Rightarrow X$.

PROOF. By Theorem 15.5 of [6], it suffices to prove convergence under the uniform topology. Denote the uniform metric is denoted by ρ, and the modulus of continuity in the uniform topology by

$$w(x, \delta) = \sup_{|s-t|<\delta} |x(s) - x(t)|.$$

For $f \in \mathbf{D}$ and $h \in \mathbf{D}_0$, consider the following conditions: (a) $\rho(f \circ h, f) > \varepsilon$, (b) $m(f \circ h, \delta) > \varepsilon$ and (c) $\rho(h, \chi) > \delta$. Suppose we can show that if (a) is true, then either (b) or (c) must be true. Then, for all n, δ, ε,

$$\{\omega\colon \rho(X_n \circ \Phi_n, X_n) > \varepsilon\} \subset \{\omega\colon w(X_n \circ \Phi_n, \delta) > \varepsilon\} \cup \{\omega\colon \rho(\Phi_n, \chi) > \delta\}.$$

Since $\chi \in C$, $\rho(\Phi_n, \chi) \Rightarrow 0$. By assumption, $X_n \circ \Phi_n \Rightarrow X$ with $P\{X \in \mathbf{C}\} = 1$, hence $X_n \circ \Phi_n$ is $\mathbf{C}$ tight. Consequently, $\rho(X_n \circ \Phi_n, X_n) \Rightarrow 0$ and by the converging together theorem, $X_n \Rightarrow X$.

To complete the proof, we need to show that (a) implies either (b) or (c). First suppose that (a) is true and (b) is false, that is,

$$\sup_{0\le t\le 1} |f \circ h(t) - f(t)| > \varepsilon \tag{A.1}$$

and

$$|f \circ h(s) - f \circ h(t)| \le \varepsilon, \qquad 0 \le t, s \le 1, |s - t| < \delta. \tag{A.2}$$

These two conditions imply the existence of a t, $0 \le t \le 1$, such that $h(s) \ne t$ for all $t - \delta \le s \le t + \delta$. Since h is continuous and nondecreasing, either $h(t - \delta) > t$ or $h(t + \delta) < t$. In either case,

$$\sup_{0\le t\le 1} |h(t) - t)| > \delta,$$

that is, $\rho(h, \chi) > \delta$ and (c) is true.

Now suppose that (a) is true and (c) is false, that is, we have condition (A.1) and

$$|h(t) - t| \le \delta, \qquad 0 \le t \le 1. \tag{A.3}$$

Fix an arbitrary t. Then for $t - \delta \le s \le t + \delta$, it must be the case that

$h(t-\delta) \le h(s) \le h(t+\delta)$ by the monotonicity of h. But from (A.3),

$$t - 2\delta \le h(t-\delta) \le t,$$
$$t \le h(t+\delta) \le t + \delta.$$

Since h is continuous, there exists an s, $t-\delta \le s \le t+\delta$ such that $h(s) = t$, implying that

$$\sup_{|s-t|<\delta} |f \circ h(t) - f \circ h(s)| \ge \varepsilon$$

by (A.1). Hence (c) is true. □

Acknowledgments. This paper is taken from the author's Ph.D. dissertation, written under the supervision of J. M. Harrison. The author's interest in this area of research is in part due to the influence of M. I. Reiman; the author is indebted to Dr. Reiman for many helpful discussions and clarifications. S. Varma is thanked for making available his early results on the analysis of some similar network models. The editors' and referees' comments on the first draft of this paper significantly improved its presentation. Finally, financial support by Bell Laboratories Graduate Research Program for Women during the author's graduate career is gratefully acknowledged.

REFERENCES

[1] Baccelli, F. and Liu, Z. (1990). On the execution of parallel programs on multiprocessor systems—a queueing theory approach. *J. ACM* **37** 373–414.

[2] Baccelli, F. and Makowski, A. M. (1989). Queueing models for systems with synchronization constraints. *Proceedings of the IEEE* **77** 138–161.

[3] Baccelli, F., Makowski, A. M. and Shwartz, A. (1989). The fork-join queue and related systems with synchronization constraints: stochastic ordering, approximations and computable bounds. *J. Adv. Probab.* **21** 629–660.

[4] Baccelli, F., Massey, W. A. and Towsley, D. (1989). Acyclic fork-join queueing networks. *J. ACM* **36** 615–642.

[5] Benes, V. (1963). *General Stochastic Processes in the Theory of Queues*. Addison-Wesley, Reading, MA.

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

[7] Dai, J. G. and Harrison, J. M. (1992). Reflected Brownian motion in an orthant: Numerical methods for steady-state analysis. *Ann. Appl. Probab.* **2** 65–86.

[8] Dai, J. G., Nguyen, V. and Reiman, M. I. (1992). Sequential bottleneck decomposition: A decomposition approximation for open queueing networks. Unpublished manuscript.

[9] Flatto, L. and Hahn, S. (1984). Two parallel queues created by arrivals with two demands I. *SIAM J. Appl. Math.* **44** 1041–1053.

[10] Flatto, L. and Hahn, S. (1985). Two parallel queues created by arrivals with two demands II. *SIAM J. Appl. Math.* **45** 861–878.

[11] Glynn, P. W. (1990). Diffusion approximations. In *Handbook on OR & MS* **2** (D. P. Heyman and M. J. Sobel, eds.) 145–198. North-Holland, Amsterdam.

[12] Harrison, J. M. (1973). Assembly-like queues. *J. Appl. Probab.* **10** 354–367.

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

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

[15] Harrison, J. M. and Reiman, M. I. (1981). Reflected Brownian motion on an orthant. *Ann. Probab.* **9** 302–308.

[16] Harrison, J. M. and Williams, R. J. (1987). Multidimensional reflected Brownian motions having exponential stationary distributions. *Ann. Probab.* **15** 115–137.

[17] Harrison, J. M. and Williams, R. J. (1987). Brownian models of open queueing networks with homogeneous customer populations. *Stochastics* **22** 77–115.

[18] Hillier, F. S. and Lieberman, G. J. (1986). *Introduction to Operations Research*. Holden-Day, Oakland.

[19] Iglehart, D. L. and Whitt, W. (1969). The equivalence of functional central limit theorems and counting processes and associated partial sums. Technical Report 121, Dept. Operations Research, Stanford Univ.

[20] Iglehart, D. L. and Whitt, W. (1970). Multiple channel queues in heavy traffic, I and II. *Adv. in Appl. Probab.* **2** 150–177; 355–364.

[21] Knessl, C. (1988). On the diffusion approximation to a fork and join queueing model. Applied Math Research Paper AM88, Dept. Mathematics, Statistics and Comp. Sci., Univ. Illinois, Chicago.

[22] Mandelbaum, M. and Avi-Itzhak, B. (1968). Introduction to queueing with splitting and matching. *Israel J. Tech.* **6** 376–382.

[23] Nelson, R. and Tantawi, A. N. (1988). Approximate analysis of fork-join synchronization in parallel queues. *IEEE Trans. Comm.* **37** 739–743.

[24] Nguyen, V. (1990). Heavy traffic analysis of processing networks with parallel and sequential tasks. Ph.D. dissertation, Dept. Operations Research, Stanford Univ.

[25] Peterson, W. P. (1991). Diffusion approximations for networks of queues with multiple customer types. *Math. Oper. Res.* **16** 90–118.

[26] Reiman, M. I. (1982). The heavy traffic diffusion approximation for sojourn times in Jackson networks. in *Applied Probability—Computer Science: The Interface* **2** (R. L. Disney and T. Ott, eds.) 409–422. Birkhäuser, Boston.

[27] Reiman, M. I. (1984). Open queueing networks in heavy traffic. *Math. Oper. Res.* **9** 441–458.

[28] Reiman, M. I. (1988). A multiclass feedback queue in heavy traffic. *Adv. in Appl. Probab.* **20** 179–207.

[29] Schmenner, R. W. (1981). *Production/Operations Management: Concepts and Situations.* Science Research Associates, Chicago.

[30] Varma, S. (1990). Heavy and light traffic approximations for queues with synchronization constraints. Ph.D. dissertation, Dept. Electrical Engineering, Univ. Maryland.

[31] Whitt, W. (1970). Weak convergence of probability measures on the function space $C[0,\infty)$. *Ann. Math. Statist.* **41** 939–944.

[32] Whitt, W. (1971). Weak convergence theorems for priority queues: Preemptive-resume discipline. *J. Appl. Probab.* **8** 74–94.

[33] Whitt, W. (1983). The queueing network analyzer. *Bell. System Tech. J.* **62** 2779–2815.

Sloan School of Management
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139