HAL archives-ouvertes.fr

On the execution of parallel programs on multiprocessor systems. A queuing theory approach

François Baccelli, Zhen Liu

To cite this version:

François Baccelli, Zhen Liu. On the execution of parallel programs on multiprocessor systems. A queuing theory approach. [Research Report] RR-0833, INRIA. 1988. inria-00075719

HAL Id: inria-00075719
https://hal.inria.fr/inria-00075719

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 833

ON THE EXECUTION OF PARALLEL PROGRAMS ON MULTIPROCESSOR SYSTEMS-A QUEUEING THEORY APPROACH

François BACCELLI
Zhen LIU

AVRIL 1988

RR 833

EXECUTION DE PROGRAMMES PARALLELES SUR ARCHITECTURES MULTIPROCESSEURS: UNE APPROCHE PAR LA THEORIE DES FILES D'ATTENTE

François BACCELLI[1] et Zhen LIU[2]

[1]INRIA Sophia-Antipolis
Valbonne 06565, France

[2]CNET PAA-ATR
Issy les Moulineaux, 92131, France

RESUME

Une nouvelle classe de réseaux de files d'attente est définie pour l'évaluation des performances de systèmes multiprocesseurs multiprogrammés avec parallélisme au niveau des tâches. Le cas considéré est celui où les programmes soumis ont tous la même structure de graphe de tâches et où l'ordonnancement est Premier Arrivé, Premier Servi.

Des équations trajectorielles sont établies pour ces réseaux, tenant compte à la fois des mécanismes d'attente dûs à la compétition des tâches pour les processeurs et des mécanismes de synchronisation traduisant les contraintes de précédence entre tâches.

Les conditions de stabilité de ces réseaux sont d'abord établies sous des hypothèses statistiques générales, ce qui détermine le débit maximal du système multiprocesseurs ou encore le taux maximum de soumissions de programmes garantissant la stabilité d'un tel système. La méthode employée est fondée sur des techniques de théorie des files d'attente et de théorie ergodique.

Des équations intégrales de base caractérisant le régime stationnaire de ces réseaux sont ensuite établies. De nombreux critères de performance tels que la charge stationnaire des divers processeurs ou les temps de réponse stationnaires des programmes se déduisent de la solution de ces équations. Un schéma iteratif convergeant vers la solution de ces équations intégrales est proposé ainsi que diverses bornes supérieures et inférieures qui sont dérivées de théorèmes d'ordonnancement stochastique.

Mots clés: Evaluation de performances, graphes de tâches, multiprogrammation, architectures multiprocesseurs, synchronisation, réseaux de files d'attente, condition de stabilité, théorie ergodique, ordonnancement stochastique.

PAPIER RÉCUPERÉ ET RECYCLÉ

On the Execution of Parallel Programs on Multiprocessor Systems — A Queueing Theory Approach

François BACCELLI

INRIA
Centre Sophia Antipolis
06565 Valbonne Cedex
France

Zhen LIU

CNET PAA-ATR
38 Avenue General Leclerc
92131 Issy Les Moulineaux
France

April 26, 1988

Abstract

A new class of queueing models, named Synchronized Queueing Networks, is proposed for evaluating the performance of multiprogrammed and multitasked multiprocessor systems in the case where the workload consists of parallel programs of similar task graph structure and the scheduling is First Come First Serve.

Pathwise evolution equations are established for these networks that capture both the queueing mechanisms due to the competition of tasks for processors and the synchronization mechanisms translating precedence constraints between tasks.

A general expression is deduced for the stability condition of such queueing networks under general statistical assumptions, which yields the maximum program throughput of the multiprocessor system or equivalently the maximum rate at which programs can be executed or submitted. The proof is based on queueing and ergodic theory considerations.

Basic integral equations are also derived for the stationary distribution of important performance criteria such as the workload of the queues and the program response times. An iterative numerical schema that converges to this solution is proposed and various upper and lower bounds on the moments of these quantities are derived using stochastic ordering techniques.

Categories and Subject Descriptors: C.1.2 [**Processor Architectures**]: *Multiprocessors-parallel processors*; C.4 [**Performance of Systems**]: *modeling techniques, design studies*; D.1.3 [**Programming Techniques**]: *Concurrent Programming*; D.4.1 [**Operating Systems**]: *Process Management, concurrency, multiprocessing, multiprogramming, scheduling, synchronization*; D.4.8 [**Operating Systems**]: *Performance-modeling and prediction, queueing theory, stochastic analysis.*

General Terms: Design, parallelism, performance, theory.

Additional Key Words and Phrases: Performance Evaluation, task graph, multiprogramming and multitasking, stability condition, ergodic theory, stochastic ordering, associated random variables, waiting times, response times.

1 Introduction

This paper is concerned with the performance analysis of multiprocessor systems running parallel programs. A new queueing network model is introduced that describes multiprogramming and multitasking within this context, under the assumption that all programs have the same task structure and that the allocation scheme that is used to map their tasks on the processors is static.

Following [1,2], we shall consider a parallel program to be a set of sequential tasks, the executions of which are subject to certain precedence constraints. These constraints specify various functional relationships that can exist between the tasks of the program. For instance, the execution of two tasks, one of which requires some data produced by the other, will be sequential, while it might be concurrent if no such relation exists. In parallel langages, these relationships translate into a variety of "synchronization primitives", like Fork-Join or Parbegin-Parend, allowing to control the concurrent execution of the various tasks that compose a program ([3-5]). A parallel program will hence be described by a directed graph, referred to as its task graph, where nodes represent tasks and directed edges represent the precedence relations between the tasks. A task that has several outgoing edges corresponds to simultaneous initializations of further tasks. A task that has several incoming edges corresponds to a synchronization. The execution of this task is enabled only if all its predecessors have finished their execution. Tasks which have no direct or indirect precedence relations might execute concurrently, otherwise they have to be executed sequentially. The type of the task graphs that we consider in this paper are directed acyclic graph which are capable of describing all possible relationships between the tasks [2,6].

In a general purpose multiprocessor system, different programs could clearly have different task graphs. As mentioned above the case we consider in the present paper is the one where all programs accessing the multiprocessor system have the same task graph. Such a situation is frequently encountered in real time processing applications (e.g. real time parallel signal processing [7]), in linear algebra algorithms [8] and in certain telecommunication applications [9,10]. In such a case, the various programs accessing the system may only differ in that corresponding tasks may have different execution times in two different program instanciations. Such a program variability comes for instance from the differences between the sizes or values of the data they operate on.

The multiprocessor systems under consideration here have a generic architecture with a finite number of (homogeneous or heterogeneous) processors, possibly sharing a central memory. Each processor possesses a local memory. The processors can communicate via a communication medium or the central memory.

The multiprocessor system is monoprogrammed if there is only one program being executed at a given time. Multitasking is allowed if the tasks of a program can be processed in parallel on the various processors in a way that preserves their precedence constraints. The multiprocessor system is multiprogrammed and multitasked if two or more independent programs can be executed at the same time and if the tasks of each program are executed concurrently in accordance with the partial order defined on each program.

The present paper is concerned with multiprogrammed and multitasked systems executing a sequence of programs of similar task structure. In such a case several task assignment policies can be used. A first class of assignment policies consists in allocating tasks of a newly arrived program following some adaptive scheme based on the load of the processors. These policies, which are usually referred to as dynamic assignment, will not be considered here. Another possibility consists in allocating the various tasks of each program to predefined host processors, according to a fixed assignment strategy. This may be necessary in the particular case where the various processors have dedicated hardware or software. Redundant task assignment will not be considered here, i.e., a task is assigned to one and only one processor. Each processor, however, may have several resident tasks belonging to the same program. This is for instance the case if the number of processors in the system is smaller than the number of tasks in the graph.

As the multiprocessor system is assumed to be multiprogrammed, different programs may be executed simultaneously in the system. The task assignment policy is assumed to remain the same for each program. Each processor may hence have several resident tasks belonging to the same or to different programs. Tasks allocated to the same processor are queued up for execution. The queueing discipline is FCFS (First Come First Serve) at the level of programs, i.e., tasks of n-th program are queued up after the tasks of $n-1$-th. On each processor, tasks of the same program are queued up in accordance with the precedence constraints described by the task graph. The executions of all these tasks are then synchronized according to the following simple rule: A tagged task begins its execution as soon as 1) all its direct predecessor tasks have completed their executions and, 2) the processor it was allocated to has completed the execution of all tasks that were allocated to the same processor and that precede the tagged task in the FCFS order.

Observe that such a static allocation discipline together with the FCFS discipline at the program level may eventually result in a reduced processor utilization. Compared with the dynamic scheme, static allocation may indeed create situations where one processor is idle, while another processor has several tasks queued up. However, a dynamic assignment scheme requires more information on the state of system than a static one. In addition, dynamic allocation implies a central assignment queue that may be a cause of contention. Similarly, the FCFS assumption concerning programs may create situations where a processor stays idle because of task precedence constraints while there are some other tasks of later programs available for execution on the same processor. The FCFS discipline has however obvious robustness properties that compensate this drawback. It is beyond the scope of the present paper to discuss the relative merits of dynamic and static allocation schemes or to compare the FCFS discipline with other possible scheduling disciplines. The reasons for limiting the discussion to static allocation and FCFS scheduling are both practical and mathematical: this is the simplest possible case, both in terms of implementation and in terms of performance analysis.

The communications between tasks that are preceding one another in the task graph can be implemented in two different ways. The first one is based on the use of shared variables (variables that can be referenced by more than one task) and the second one on message passing. Both types can be represented in the task graph by adding certain communication tasks to be allocated to specific queues. There exist in the literature a variety of models for analyzing central memory contentions and the interference due to the sharing of the communication medium within this context (for references see [11-14]). In this paper we shall not focus on these problems. Nevertheless these effects can be taken into account by adding to the task graph adequate communication tasks and memory access tasks for each precedence constraint. The communication tasks should be allocated to a specific queue representing the communication medium and the memory access tasks to a specific queue representing the central memory. In fact, neglecting these phenomenons should only result in slight errors. Jones and Schwarz's experiences in [15] show that idleness of processors due to shared variable reference represent less than one percent of time if a data locking mechanism is used. In the case of message passing, it is enough to introduce for each precedence constraint a communication task that requires the services of the interprocessor communication medium and which should then be allocated to a specific queue representing this medium. Bianchini and Shen claimed in [16] that for many mission-oriented multiprocessor systems interprocessor communication is deterministic.

We now survey briefly the various probabilistic models that have been proposed in the literature and that are related to the problem considered here. If the multiprocessor system is monoprogrammed but multitasking is allowed, the evaluation of the statistics of the execution time of a program can be accomplished by means of the PERT techniques [17-19]. If, however, the multiprocessor system is multiprogrammed and multitasked, the analysis of the performance behavior is much more complex. Initially, most attention focussed on the simplest case, namely static allocation, programs with fixed task structure and as many processors as there are tasks in one graph. The first attempt to analyze this problem was made on the case where the task graph has K nodes without precedence constraints, like for instance in the case of a program of the form Parbegin $T_1, ..., T_K$ Parend, where $T_1, ..., T_K$ are K tasks that can execute concurrently. In this case, exact solutions have been provided for $K = 2$ in [20,21]. Approximate solutions, bounds and logarithmic asymptotics have been derived on the mean program response times for arbitrary values of K in [22-25] and conditions for stability have been presented in [22,26]. More recently, the class of acyclic Fork-Join queueing networks has been studied in [27]. This last paper generalizes the results of [25] to the case where the structure of the tasks graph is acyclic. Conditions for stability are established as well as bounds on the response times of programs under general statistical assumptions.

For the more realistic cases where the number of processors is smaller than the number of tasks in a program, only approximate models have been considered in the literature (see [28-30]). The approach we propose in this paper is hence a first attempt towards an exact model within this context. To the best authors knowledge, all of the more complex problems like dynamic allocation or programs with variable structure are completely open apart from the case of systems with infinitely many processors ([31-34]). For instance, [33] focuses on the case of random graph structure, deterministic and fixed task execution times and infinitely many processors.

The paper is organized as follows. Section 2 is devoted to the definition of a new class of Synchronized Queueing Networks (SQN), which describes the parallel processing systems introduced above. In this model, the processors (and eventually the communication medium and the memory) are all represented by single server queues. Customers attended by these queues represent tasks (or communications). The service of the customers are subject to the precedence constraints specified by the task graph which is assumed to be acyclic. The main result of Section 2 consists in the derivation of the basic evolution equations governing the behavior of these networks. The interarrival of the parallel programs and the service requirements of the tasks will be assumed to be generaly distributed possibly correlated random variables. This allows to represent asynchronous program submissions and some uncertainty on the actual value of the tasks execution times (or of communication times) as it is usually done in the modeling of computer systems.

Section 3 provides the necessary and sufficient stability conditions for such queueing networks under these general statistical assumptions. In this section the SQN is first decomposed into certain subnetworks according to the structure of the task graph, and it is shown that the stability condition of the SQN reduces to the stability conditions of its subnetworks. A general expression is then established for this condition. This determines the maximum rate at which programs can be executed within this context or equivalently, the maximum intensity of program submissions that preserves the system stability. This result is new to the best authors knowledge.

Within this context, the waiting (resp. response) time of a task is defined as the delay between the program arrival date and the date when the task begins (resp. completes) its execution. The response time of a parallel program is defined as the delay between the program arrival time and the time when all its tasks have completed their executions. This last response time can also be expressed as the maximum of the response times of its tasks. Waiting and response times are important performance criteria of such multiprocessor systems. Sections 4,5 and 6 are devoted to the analysis of the transient and stationary behavior of these quantities. In Section 4, basic integral equations are established for the joint distribution function of the waiting times under classical renewal assumptions. The exact analytical solution of these integral functional equations seem very difficult to obtain. However, an iterative numerical schema which converges to this solution is provided. Sections 5 and 6 are devoted to the derivation of various bounds on the solution of these equations. Section 5 provides lower bounds based on the notion of stochastic convex ordering and Section 6 upper bounds based on the notion of associated random variables. Various numerical algorithms are also provided for the computation of both types of bounds.

2 Definitions and Evolution Equations

The original program task graph will be assumed to be acyclic and will be represented by the couple $G_o = (V_o, E_o)$, where V_o is the set of nodes corresponding to the tasks and E_o the set of directed edges indicating precedence relations between tasks. Observe that G_0 depends on the algorithm of the program to be executed only and not on the system architecture or configurations of the multiprocessor systems under consideration.

We introduce now a new graph $G = (V, E)$ that takes into account the number of processors in the system and the task assignment strategy. Let K denote the number of processors in the system. Let $A : V_o \to \{1, \cdots, K\}$ denote the assignment strategy: $A(v)$ is the index of the processor to which task $v \in V_o$ is allocated.

$$B_k = \{v | v \in V_o, A(v) = k\}, \quad 1 \le k \le K$$

denotes the set of tasks allocated to processor k. $G = (V, E)$ is obtained from the initial task graph $G_o = (V_o, E_o)$ as follows:

$$\begin{aligned} V &= V_o \\ E &= E_o \bigcup E' \end{aligned}$$

where E' is a set of edges indicating certain additional precedence relations required for establishing a total order on every B_k, $1 \le k \le K$. These total orders should be compatible with the partial order defined by G_o. In fact there exist several different ways of constructing such total orders. More precisely, the set of all total orders on B_k, $1 \le j \le K$ represents the set of all possible task scheduling policies that are compatible with A and G_o. The choice of these execution orders may influence strongly the performances. The comparison of all these partial scheduling strategies will be the object of another paper ([35]). In the present paper, it will be assumed that the transformation $G_o \rightarrow G$ is given. It will also be assumed that there are no redundant edge in G, i.e., if $i \rightarrow j \in E$, there are no paths of G, other than the edge $i \rightarrow j$ from i to j. Note that since G_o is acyclic G is also acyclic.

At this point, new notations are introduced on graph G, that will be used throughout the paper. $P_G(i)$ will denote the set of immediate predecessors of task i in G :

$$P_G(i) = \{j | j \in V, j \rightarrow i \in E\}$$

and $\Pi_G(i)$ the set of predecessors of task i in G :

$$\Pi_G(i) = \{j | j \in V, \text{ there exists a path of } G \text{ from } j \text{ to } i\}$$

In other words,

$$\Pi_G(i) = \bigcup_{n=1}^{|V|} P_G^n(i)$$

where $P_G^n(i)$ denotes $\underbrace{P_G(P_G(\cdots P_G}_{n}(i)\cdots))$. Similarly $\Pi_G^n(i)$ will denote the n-th iterate of $\Pi_G(i)$: $\Pi_G^n(i) = \underbrace{\Pi_G(\Pi_G(\cdots \Pi_G}_{n}(i)\cdots))$.

For $i, j \in V$, $\mathcal{P}(i,j)$ will denote the set of paths from i to j in G, if any:

$$\mathcal{P}(i,j) = \{v_1, \cdots, \ v_m \in V, \ 1 \leq m \leq |V| \ | i = v_1 \rightarrow v_2, \cdots, v_{m-1} \rightarrow v_m = j \in E\}.$$

We define the length of a path in G to be the sum of the execution times of the tasks situated on the path.

For reasons which will become apparent later on, it will be convenient to order the vertices of G in function of the processors they are allocated to and their level in G. Define the *level* of task i, denoted by $lev(i)$, as follows :

$$lev(i) = \max_{j \in P_G(i)} lev(j) + 1$$

where by definition, $lev(i) = 1$ if $P_G(i) = \emptyset$. $N = \max_{i \in V} lev(i)$ will denote the *level* of G.

Let t_j^k, $1 \leq k \leq K$, $1 \leq j \leq N$ denote the task of level j allocated to processor k if any. Let t_b^k (resp. t_e^k) denote the task of G allocated to processor k ($1 \leq k \leq K$) with the smallest (resp. largest) level value.

Generally speaking, the additional index $n \geq 0$ added to one of the objects defined above indicates a reference to the instanciation of this object in the n-th program. For instance, $t_{j,n}^k$ is task t_j^k in the n-th program and $B_{k,n}$ ($1 \leq k \leq K$) is the set of tasks of program n that are allocated to processor k.

The SQN associated with G consists of K queues, one per processor. The behavior of the SQN is determined by the following three rules:

(i) There is a single external arrival stream with pattern $a_0 = 0 < a_1 < \cdots < a_n < \cdots \in R^+$. The n-th date of this external arrival stream triggers the arrival of a bulk of customers to queue j ($1 \leq j \leq K$). The set of customers of this bulk is precisely $B_{j,n}$.

(ii) The service discipline of each queue is FCFS in the sense that customers of $B_{k,n}$ ($1 \leq k \leq K$, $n \geq 1$) are allowed to be serviced iff all the customers in $B_{k,n-1}$ have been serviced. In addition to that FCFS rule, customer $t_{i,n}^k \in B_{k,n}$ ($1 \leq i \leq |V|$, $n \geq 0$) can only be serviced if all the tasks which are its predecessors in G have been serviced. Customer $t_{i,n}^k$ requires $\sigma_{i,n}^k \in R^+$ units of processing time.

(iii) There is a single output stream out of this network. Its n-th date coincides with the latest of the service completions $t_{i,n}^k$, $1 \leq i \leq |V|$, $1 \leq k \leq K$.

Before establishing the evolution equations of the SQN, we introduce a new network which will be proved to have an equivalent time behavior.

From task graph G we construct a new task graph, denoted by $\underline{G} = (\underline{V}, \underline{E})$, by filling up G with new fictive tasks and by introducing new fictive processors. A fictive task requires no service, the associated service requirement sequence is thus a series of zeros. A fictive processor receives only fictive tasks as its resident tasks. $\underline{G}$ is constructed by the following algorithm, where K' is the number of fictive processors and t_u^k denotes the task allocated to processor $1 \leq k \leq K$ with task level $1 \leq u \leq N$ in G, if it exists.

1. $K' \leftarrow 0$; $\underline{E} \leftarrow \emptyset$;

2. $\underline{V} \leftarrow V$, where every task in $\underline{V}$ receives the same service requirement sequence and the same assignment strategy as in G.

3. **for** every edge $t_u^k \rightarrow t_{u'}^{k'}$ of E,
   if $u' = u + 1$
   then $\underline{E} \leftarrow \underline{E} \bigcup \{t_u^k \rightarrow t_{u'}^{k'}\}$
   otherwise
   if $k = k'$
   then
   $$\underline{V} \leftarrow \underline{V} \bigcup \{t_{u+1}^k, t_{u+2}^k, \cdots, t_{u'-1}^k\}$$
   and
   $$\underline{E} \leftarrow \underline{E} \bigcup \{t_u^k \rightarrow t_{u+1}^k, t_{u+1}^k \rightarrow t_{u+2}^k, \cdots, t_{u'-1}^k \rightarrow t_{u'}^k\}$$
   where $t_{u+1}^k, \cdots, t_{u'-1}^k$ are new fictive tasks allocated to processor k.
   otherwise
   $$K' \leftarrow K' + 1$$
   $$\underline{V} \leftarrow \underline{V} \bigcup \{t_{u+1}^{K+K'}, t_{u+2}^{K+K'}, \cdots, t_{u'-1}^{K+K'}\}$$
   and
   $$\underline{E} \leftarrow \underline{E} \bigcup \{t_u^k \rightarrow t_{u+1}^{K+K'}, t_{u+1}^{K+K'} \rightarrow t_{u+2}^{K+K'}, \cdots, t_{u'-1}^{K+K'} \rightarrow t_{u'}^{k'}\}$$
   where $t_{u+1}^{K+K'}, \cdots, t_{u'-1}^{K+K'}$ are new fictive tasks allocated to fictive processor $K + K'$.
   endfor

4. Let $t_{u_1}^k, t_{u_1+1}^k, \cdots, t_{u_2}^k$ be the set of (possibly fictive) tasks allocated to processor (or fictive processor) k, $1 \leq k \leq K + K'$, $1 \leq u_1 \leq u_2 \leq N$.
   for every k, $1 \leq k \leq K + K'$,
   $$\underline{V} \leftarrow \underline{V} \bigcup \{t_1^k, t_2^k, \cdots, t_{u_1-1}^k, t_{u_2+1}^k, t_{u_2+2}^k, \cdots, t_N^k\};$$
   $$\underline{E} \rightarrow \underline{E} \bigcup \{t_1^k \rightarrow t_2^k, t_2^k \rightarrow t_3^k, \cdots, t_{u_1-1}^k \rightarrow t_{u_1}^k, t_{u_2}^k \rightarrow t_{u_2+1}^k, t_{u_2+1}^k \rightarrow t_{u_2+2}^k, \cdots, t_{N-1}^k \rightarrow t_N^k\};$$
   where $t_1^k, \cdots, t_{u_1-1}^k, t_{u_2+1}^k, \cdots, t_N^k$ are new fictive tasks allocated to processor (or fictive processor) k.
   endfor

In the sequel, the notations that were defined initially for G will be used similarly for $\underline{G}$. The index G (resp. $\underline{G}$) will be added to mention which graph is referenced. For instance, $\mathcal{P}_{\underline{G}}(i, j)$ denotes the set of paths from i to j in $\underline{G}$. The following properties are easily proved:

Lemma 1

The task graph $\underline{G}$ has the following properties :

1. *For every u, $1 \leq u \leq N$, and every k, $1 \leq k \leq K + K'$, there exists one and only one task of level u allocated to processor k.*

2. *Every non-fictive task in $\underline{G}$ is allocated on the same processor as in G.*

3. *Every non-fictive task in $\underline{G}$ receives the same service requirement sequence as in G.*

4. *Every non-fictive task in $\underline{G}$ has the same level value as in G.*

5. *If $v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_f$ ($m > 0, v_0, v_1, \cdots, v_m \in \underline{V}$) forms a path of $\underline{G}$, and if $v_1, \cdots, v_{m-1}$ are fictive tasks and v_0, v_m are non-fictive ones, then $v_1, \cdots, v_{m-1}$ are allocated to a same fictive processor, and furthermore, the fictive tasks allocated to this fictive processor, other than v_1, have no non-fictive tasks as their immediate predecessors in $\underline{G}$, Symmetrically, the fictive tasks allocated to this fictive processor, other than v_{m-1}, have no non-fictive tasks as their immediate successors in $\underline{G}$.*

6. *A fictive task may have at most one non-fictive task and at most one fictive task as its immediate successors in $\underline{G}$. Symmetrically, a fictive task may have at most one non-fictive task and at most one fictive task as its immediate predecessors in $\underline{G}$.*

7. *There is a path $P1$ in $\underline{G}$ from non-fictive task i to non-fictive task j, if and only if there is a path $P2$ in G form i to j, where $P2$ is constructed by removing all the fictive tasks from $P1$.*

8. *Precedence relations in $\underline{G}$ exist only between consecutive levels, i.e.,*

$$t_u^k \rightarrow t_{u'}^{k'} \in \underline{E} \Rightarrow u' - u = 1$$

The SQN associated with $\underline{G}$ is defined similarly to the one associated with G. SQN-$\underline{G}$ is composed of $K + K'$ queues, one per server (or fictive server). Let $t_{u,n}^k$ denote the unique (see property 1 of Lemma 1) task of level u, $1 \leq u \leq N$, allocated by the n-th program, $n \geq 0$ to processor k, $1 \leq k \leq K+K'$. Let $\{\sigma_{u,n}^k\}_{n=0}^{\infty}$ denote the service requirement of task $t_{u,n}^k$ ($1 \leq u \leq N,\ 1 \leq k \leq K+K',\ n \geq 0$). Observe that $\sigma_{u,n}^k = 0$ ($n = 0, 1, 2, \cdots$) if t_u^k is a fictive task. The behavior of SQN-$\underline{G}$ is determined by the following simplified rules :

(i') The n-th bulk arrival to queue k ($1 \leq k \leq K + K'$) is composed of customers $t_{1,n}^k, t_{2,n}^k, \cdots, t_{N,n}^k$.

(ii') Customer $t_{u,n}^k$ ($1 \leq u \leq N,\ n \geq 0,\ 1 \leq k \leq K + K'$) requires $\sigma_{u,n}^k \in R^+$ units of processing time and begins its service as soon as, for $n > 0$, all customers $t_{u,n-1}^k$, $1 \leq u \leq N$ and for $n > 0$, all customers $t_{u-1,n}^i$, $1 \leq i \leq K$, being its predecessors in $\underline{G}$ have been serviced.

(iii') The n-th ($n \geq 0$) departure epoch coincides with the latest service completions of the tasks $t_{N,n}^k$, $1 \leq k \leq K + K'$.

For $n = 0, 1, 2, \cdots$, let τ_n be the n-th interarrival of the external stream : $\tau_n = a_{n+1} - a_n$, $\underline{w}_{i,n}^k$ be the workload of queue k seen by customer $t_{i,n}^k$ $1 \leq i \leq N$, $1 \leq k \leq K + K'$. Similarly, let $\underline{w}_n^k$ be the workload of queue k at the n-th external arrival, $1 \leq k \leq K + K'$, $\underline{r}_n^k$ be the n-th response time of queue k defined as the delay between the n-th bulk arrival to queue k and the latest departure of the bulk customers, and $\underline{r}_n$ be the n-th network response time defined as the delay between the n-th arrival and the n-th departure of the system. Within this context, the workload of a queue is understood as the time it takes to clear this queue of all its customers when stopping further external arrivals. This includes the service times of the customers present in the queue and the synchronization delays due to the precedence constraints indicated in the task graphs.

Theorem 1

Assume the SQN-$\underline{G}$ is empty at time 0. For every n, i, and k ($n \geq 0$, $1 \leq i \leq N-1$, $1 \leq k \leq K + K'$),

$$\underline{w}_0^k = 0 \quad (2.1)$$

$$\underline{w}_{1,n}^k = \underline{w}_n^k \quad (2.2)$$

$$\underline{w}_{i+1,n}^k = \max_{\{j \mid t_i^j \in P_{\underline{G}}(t_{i+1}^k)\}} (\underline{w}_{i,n}^j + \sigma_{i,n}^j) \quad (2.3)$$

$$\underline{w}_{n+1}^k = \max(0, \underline{w}_{N,n}^k + \sigma_{N,n}^k - \tau_n) \quad (2.4)$$

where $P_{\underline{G}}(t^k_{i+1})$ denotes the set of immediate predecessors of task t^k_{i+1} in $\underline{G}$. The n-th response time of queue k ($1 \le k \le K+K'$) is given by

$$\underline{r}^k_n = \underline{w}^k_{N,n} + \sigma^k_{N,n} \tag{2.5}$$

and the n-th network response time $\underline{r}_n$ by

$$\underline{r}_n = \max_{1 \le k \le K+K'} \underline{r}^k_n \tag{2.6}$$

Proof

The boundary condition (2.1) follows from the initial condition assumption. The fact that tasks of level 1 have no predecessors in $\underline{G}$ validates equation (2.2). From property 8) of Lemma 1, all the predecessors of t^k_{i+1} in $\underline{G}$ have the form t^j_i, $1 \le j \le K+K'$. Therefore the right hand side of equation (2.3) gives the workload of queue k seen by customer $t^k_{i+1,n}$ upon its arrival. Notice that the set $\{j | t^j_i \in P_{\underline{G}}(t^k_{i+1})\}$ is never empty from the very definition of the level function.

Observe that $\underline{w}^k_{N,n} + \sigma^k_{N,n} + a_n$ is the date where the last customer of the n-th bulk completes its service in queue k. Hence, we have

$$\underline{w}^k_{n+1} = \max(0, \underline{w}^k_{N,n} + \sigma^k_{N,n} + a_n - a_{n+1})$$

which is exactly the same expression as (2.4) since $\tau_n = a_{n+1} - a_n$

Equations (2.5) and (2.6) follow from the definition. □

Theorem 1bis

Assume that SQN-$\underline{G}$ is empty at time 0. For every n and k ($n \ge 0$, $1 \le k \le K+K'$),

$$\underline{w}^k_0 = 0 \tag{2.7}$$

$$\underline{w}^k_{n+1} = \max(0, \max_{\{j | t^j_1 \in \Pi_{\underline{G}}(t^k_N)\}} (\underline{w}^j_n + \underline{l}^{j,k}_n) - \tau_n) \tag{2.8}$$

$$\underline{r}^k_n = \max_{\{j | t^j_1 \in \Pi_{\underline{G}}(t^k_N)\}} (\underline{w}^j_n + \underline{l}^{j,k}_n) \tag{2.9}$$

$$\underline{r}_n = \max_{1 \le k \le K+K'} \underline{r}^k_n \tag{2.10}$$

where $\underline{l}^{j,k}_n$ denotes the maximum of the lengths of the paths from t^j_1 to t^k_N in $\underline{G}$:

$$\underline{l}^{j,k}_n = \max_{\{(t^j_1,\, t^{k_{N-2}}_2, \ldots,\, t^{k_1}_{N-1},\, t^k_N) \in \mathcal{P}_{\underline{G}}(t^j_1, t^k_N)\}} (\sigma^j_{1,n} + \sigma^{k_{N-2}}_{2,n} + \cdots + \sigma^{k_1}_{N-1,n} + \sigma^k_{N,n}) \tag{2.11}$$

and $\Pi_{\underline{G}}(t^k_N)$ is the set of predecessors of task t^k_N in $\underline{G}$.

Proof

Equation (2.8) can be derived from (2.2)-(2.4) as follows. From (2.3), we get

$$\underline{w}^k_{N,n} = \max_{\{k_1 | t^{k_1}_{N-1} \in P_{\underline{G}}(t^k_N)\}} (\underline{w}^{k_1}_{N-1,n} + \sigma^{k_1}_{N-1,n})$$

$$\underline{w}^{k_1}_{N-1,n} = \max_{\{k_2 | t^{k_2}_{N-2} \in P_{\underline{G}}(t^{k_1}_{N-1})\}} (\underline{w}^{k_2}_{N-2,n} + \sigma^{k_2}_{N-2,n})$$

$$\vdots$$

$$\underline{w}^{k_{N-2}}_{2,n} = \max_{\{k_{N-1} | t^{k_{N-1}}_1 \in P_{\underline{G}}(t^{k_{N-2}}_2)\}} (\underline{w}^{k_{N-1}}_{1,n} + \sigma^{k_{N-1}}_{1,n})$$

Notice that none of the sets $\{j|t_{N-j}^{k_j} \in P_{\underline{G}}(t_{N-j+1}^{k_{j+1}})\}$ is empty, from the definition of the level function. Therefore, we get by replacement,

$$\begin{aligned}
\underline{w}_{N,n}^{k} + \sigma_{N,n}^{k} &= \max_{\{k_1|t_{N-1}^{k_1} \in P_{\underline{G}}(t_N^k)\}} (\underline{w}_{N-1,n}^{k_1} + \sigma_{N-1,n}^{k_1}) + \sigma_{N,n}^{k} \\
&= \max_{\{k_1|t_{N-1}^{k_1} \in P_{\underline{G}}(t_N^k)\}} (\underline{w}_{N-1,n}^{k_1} + \sigma_{N-1,n}^{k_1} + \sigma_{N,n}^{k}) \\
&= \max_{\{k_1|t_{N-1}^{k_1} \in P_{\underline{G}}(t_N^k)\}} (\max_{\{k_2|t_{N-2}^{k_2} \in P_{\underline{G}}(t_{N-1}^{k_1})\}} \\
&\quad (\underline{w}_{N-2,n}^{k_2} + \sigma_{N-2,n}^{k_2}) + \sigma_{N-1,n}^{k_1} + \sigma_{N,n}^{k}) \\
&= \max_{t_{N-1}^{k_1} \in P_{\underline{G}}(t_N^k)} \max_{t_{N-2}^{k_2} \in P_{\underline{G}}(t_{N-1}^{k_1})} \\
&\quad (\underline{w}_{N-2,n}^{k_2} + \sigma_{N-2,n}^{k_2} + \sigma_{N-1,n}^{k_1} + \sigma_{N,n}^{k}) \\
&= \cdots
\end{aligned}$$

and we obtain finally

$$\begin{aligned}
\underline{w}_{N,n}^{k} + \sigma_{N,n}^{k} &= \max_{\{k_1|t_{N-1}^{k_1} \in P_{\underline{G}}(t_N^k)\}} \max_{\{k_2|t_{N-2}^{k_2} \in P_{\underline{G}}(t_{N-1}^{k_1})\}} \cdots \max_{\{k_{N-1}|t_1^{k_{N-1}} \in P_{\underline{G}}(t_2^{k_{N-2}})\}} \\
&\quad (\underline{w}_{1,n}^{k_{N-1}} + \sigma_{1,n}^{k_{N-1}} + \cdots + \sigma_{N-1,n}^{k_1} + \sigma_{N,n}^{k})
\end{aligned}$$

Observe that

$$\{k_{N-1}|t_1^{k_{N-1}} \in P_{\underline{G}}(t_2^{k_{N-2}}), t_2^{k_{N-2}} \in P_{\underline{G}}(t_3^{k_{N-3}}), \cdots, t_{N-1}^{k_1} \in P_{\underline{G}}(t_N^k)\} = \{j|t_1^j \in \Pi_{\underline{G}}(t_N^k)\}$$

The previous equation can be rewritten as

$$\begin{aligned}
\underline{w}_{N,n}^{k} + \sigma_{N,n}^{k} &= \max_{\{j|t_1^j \in \Pi_{\underline{G}}(t_N^k)\}} \max_{\{(t_1^j, t_2^{k_{N-2}}, \ldots, t_{N-1}^{k_1}, t_N^k) \in \mathcal{P}_{\underline{G}}(t_1^j, t_N^k)\}} \\
&\quad (\underline{w}_{1,n}^{j} + \sigma_{1,n}^{j} + \sigma_{2,n}^{k_{N-2}} + \cdots + \sigma_{N-1,n}^{k_1} + \sigma_{N,n}^{k})
\end{aligned} \tag{2.12}$$

In view of the definition of $\underline{l}_n^{j,k}$, (2.12) can be rewritten as

$$\underline{w}_{N,n}^{k} + \sigma_{N,n}^{k} = \max_{\{j|t_1^j \in \Pi_{\underline{G}}(t_N^k)\}} (\underline{w}_n^j + \underline{l}_n^{j,k}) \tag{2.13}$$

where the set $\{j|t_1^j \in \Pi_{\underline{G}}(t_N^k)\}$ is never empty. Furthermore, using (2.4) and (2.13), we get

$$\begin{aligned}
\underline{w}_{n+1}^{k} &= \max(0, \underline{w}_{N,n}^{k} + \sigma_{N,n}^{k} - \tau_n) \\
&= \max(0, \max_{\{j|t_1^j \in \Pi_{\underline{G}}(t_N^k)\}} (\underline{w}_{1,n}^j + \underline{l}_n^{j,k}) - \tau_n)
\end{aligned}$$

and using equation (2.2), we get

$$\underline{w}_{n+1}^{k} = \max(0, \max_{\{j|t_1^j \in \Pi_{\underline{G}}(t_N^k)\}} (\underline{w}_n^j + \underline{l}_n^{j,k}) - \tau_n)$$

This proves equation (2.8). Equation (2.9) follows from (2.5) and (2.12). □

The fictive tasks and fictive processors in SQN-$\underline{G}$ are useful in the formulation of the model. But, as one may expect, they should not introduce additional synchronization overhead to SQN-G, as it is established in Theorem 2 below.

Lemma 2

For all $K+1 \leq j \leq K+K'$ and $1 \leq k \leq K$, such that there exists a path of $\underline{G}$ from t_1^j to t_N^k, then there exists j_0, $1 \leq j_0 \leq K$, such that there exists a path of $\underline{G}$ from $t_1^{j_0}$ to t_N^k, and for all $n \geq 0$,

$$\underline{l}_n^{j_0,k} \geq \underline{l}_n^{j,k}$$

and

$$\underline{w}_n^{j_0} \geq \underline{w}_n^j$$

Proof

Let j and k be as defined in the lemma. And let $v_0 \to v_1 \to \cdots \to v_m$ $(m > 0, v_0, v_1, \cdots, v_m \in \underline{V})$ form a path of $\underline{G}$, where $v_1, \cdots, v_{m-1}$ are fictive tasks allocated to (fictive) processor j, and v_0, v_m are non-fictive tasks. According to property 5 of Lemma 1, v_0 and v_m are uniquely defined and v_0 (respectively v_m) is the only task not allocated to processor j having an edge directed to (resp. originated from) one of the fictive tasks of j.

Observe that all paths from t_1^j to t_N^k include node v_m, and have necessarily the form:

$$t_1^j \to \cdots \to v_1 \to \cdots \to v_{m-1} \to v_m \to \cdots \to t_N^k$$

Let j_0 $(1 \leq j_0 \leq K)$ be the index of the processor to which task v_0 is allocated. It is easy to see that the following path from $t_1^{j_0}$ to t_N^k:

$$t_1^{j_0} \to \cdots \to v_0 \to v_1 \to \cdots \to v_{m-1} \to v_m \to \cdots \to t_N^k$$

is also a path of $\underline{G}$. Hence we reach the conclusion that the relation

$$\underline{l}_n^{j_0,k} \geq \underline{l}_n^{j,k}$$

holds for all $n \geq 0$.

We show now the inequality

$$\underline{w}_n^{j_0} \geq \underline{w}_n^j$$

by induction on n.

It is obvious that the property is satisfied for $n = 0$. Suppose

$$\underline{w}_n^{j_0} \geq \underline{w}_n^j$$

holds for some n.

Let $lev(v_0) = u$. The fact that as soon as v_0 terminates its execution, all its successors allocated to fictive processor j are enabled and complete their execution immediately, entails

$$w_{N,n}^j = w_{N-1,n}^j \cdots = w_{u+1,n}^j \leq w_{u+1,n}^{j_0} \leq w_{u+2,n}^{j_0} \leq \cdots \leq w_{N,n}^{j_0}$$

Hence

$$\begin{aligned}
\underline{w}_{n+1}^{j_0} &= \max(0, \underline{w}_{N,n}^{j_0} + \sigma_{N,n}^{j_0} - \tau_n) \\
&\geq \max(0, \underline{w}_{N,n}^{j} + \sigma_{N,n}^{j} - \tau_n) \\
&= \underline{w}_{n+1}^j
\end{aligned}$$

By induction, the inequality

$$\underline{w}_n^{j_0} \geq \underline{w}_n^j$$

holds for all $n \geq 0$. □

We are now in position to derive the evolution equations of SQN-G. For SQN-G, denote as $w_{i,n}^k$ the workload of queue k seen by customer $t_{i,n}^k$ $1 \leq i \leq N$, $1 \leq k \leq K$ if it exists. Similarly, let w_n^k be the workload of queue k at the n-th external arrival, $1 \leq k \leq K$, r_n^k be the n-th response time of queue k and r_n the n-th network response time.

Owing to Lemma 1 and Lemma 2, we get for SQN-G the same type of results as for SQN-$\underline{G}$.

Theorem 2

Assume that SQN-G is empty at time 0. Then, for every n and k ($n \geq 0$, $1 \leq k \leq K$),

$$\begin{aligned} w_0^k &= 0 && (2.14)\\ w_{n+1}^k &= \max(0, \max_{\{j \in A(\Pi_G(t_e^k))\}} (w_n^j + l_n^{j,k}) - \tau_n) && (2.15)\\ r_n^k &= \max_{j \in A(\Pi_G(t_e^k))} (w_n^j + l_n^{j,k}) && (2.16)\\ r_n &= \max_{1 \leq k \leq K} r_n^k && (2.17) \end{aligned}$$

where $A(X)$ *denotes the set of processors to which tasks of set* X *are allocated,* $l_n^{j,k}$ *the maximum of the lengths of the paths from* t_b^j *to* t_e^k *in* G*, namely*

$$\begin{aligned} l_n^{j,k} &= \sigma_{b,n}^j + \max_{\{(t_{u_1}^{k_1}, \cdots, t_{u_m}^{k_m}) \in \mathcal{P}(t_b^j, t_e^k)\}} (\sigma_{u_2,n}^{k_2} + \cdots \sigma_{u_{m-1},n}^{k_{m-1}}) + \sigma_{e,n}^k \\ &= \max_{\{(t_{u_1}^{k_1}, \cdots, t_{u_m}^{k_m}) \in \mathcal{P}(t_b^j, t_e^k)\}} (\sigma_{u_1,n}^{k_1} + \cdots \sigma_{u_m,n}^{k_m}) && (2.18) \end{aligned}$$

and $\Pi_G(t_e^k)$ *is the set of predecessors of task* t_e^k *in* G.

Proof

Observe first that for all $n \geq 0$ and $1 \leq j, k \leq K$,

$$l_n^{j,k} = \underline{l}_n^{j,k}$$

Indeed, for all $1 \leq j, k \leq K$, equation (2.18) is obtained from (2.11) by removing the fictive tasks in $\mathcal{P}_{\underline{G}}(t_1^j, t_N^k)$.

Lemma 2 implies that for all $1 \leq k \leq K$,

$$\begin{aligned} \underline{w}_{n+1}^k &= \max(0, \max_{\{j | t_1^j \in \Pi_{\underline{G}}(t_N^k)\}} (\underline{w}_n^j + \underline{l}_n^{j,k}) - \tau_n) \\ &= \max(0, \max_{\{j | 1 \leq j \leq K, t_1^j \in \Pi_{\underline{G}}(t_N^k)\}} (\underline{w}_n^j + \underline{l}_n^{j,k}) - \tau_n) \\ &= \max(0, \max_{\{j | t_b^j \in \Pi_G(t_e^k)\}} (\underline{w}_n^j + \underline{l}_n^{j,k}) - \tau_n) \end{aligned}$$

or equivalently, for all $1 \leq k \leq K$,

$$\underline{w}_{n+1}^k = \max(0, \max_{\{j | t_b^j \in \Pi_G(t_e^k)\}} (\underline{w}_n^j + l_n^{j,k}) - \tau_n) \tag{2.19}$$

We now show by induction on n that for all $1 \leq k \leq K$, the workload vector in G, w_n^k is given by

$$w_n^k = \underline{w}_n^k \tag{2.20}$$

It is obvious that (2.20) holds for $n = 0$. Assume it holds for $n \geq 0$.

We prove first by induction that for all $1 \leq k \leq K$, and all $1 \leq u \leq N$ such that t_u^k exists in G,

$$w_{u,n}^k \leq \underline{w}_{u,n}^k \tag{2.21}$$

so that

$$w_{n+1}^k \leq \underline{w}_{n+1}^k \tag{2.22}$$

For $1 \leq k \leq K$, let $t_b^k = t_{u_1}^k$. From the fact that tasks $t_1^k, t_2^k, \cdots, t_{u_1-1}^k$ are all fictive tasks allocated to processor k with no other precedence constraints than $t_u^k \rightarrow t_{u+1}^k$, $1 \leq u \leq u_1$ and from (*ii*), (*ii'*), it follows immediately that $w^k_{u_1,n} = \underline{w}^k_{u_1,n}$, so that property (2.21) holds for all $t^k_b \in G$, $1 \leq k \leq K$. Assume now that (2.21) holds for all predecessors of task t^k_u. Owing to properties 3 and 7 of Lemma 1 and to (ii), (ii'), it follows that (2.21) holds for t^k_u, which completes the proof of (2.21) and (2.22).

On the other hand, observe that

$$w^k_{n+1} \geq \max(0, \max_{\{j | t^j_b \in \Pi_G(t^k_e)\}} (\underline{w}^j_n + l^{j,k}_n) - \tau_n) \tag{2.23}$$

Indeed, for all j such that $t^j_b \in \Pi_G(t^k_e)$ and owing to the induction assumption, processor k cannot become available for attending customer $t^k_{b,n+1}$ before date $a_n + \underline{w}^j_n + l^{j,k}_n$, which readily implies (2.23). Comparing (2.19) and (2.23) yields

$$w^k_{n+1} \geq \underline{w}^k_{n+1} \tag{2.24}$$

so that equations (2.22)-(2.24) complete the proof. □

3 Stability Condition

This section is devoted to the construction of the stationary regime of the networks under consideration. In particular, we provide a general expression for the stability condition. The discussion will be organized in three steps. First, we define the decomposition of a given SQN into a set of subnetworks. It is then established that the stability condition of the SQN reduces to the intersection of the stability conditions for the subnetworks. Lastly, we analyze the stability condition for the subnetworks, when the service requirements of the tasks and the interarrival times are stationary and ergodic stochastic sequences.

Throughout this section, a SQN will be assumed to be given, characterized by its graph G and its allocation policy A in relation with a K processor-multiprocessor system.

The Processor Graph (PG) associated with these data is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where

$$\mathcal{V} = \{1, 2, \cdots, K\}$$

and

$$\mathcal{E} = \{(k_1, k_2) | \exists i_1 \rightarrow i_2 \in E, i_1, i_2 \text{ are respectively allocated to processors } k_1, k_2\}$$

Observe that although G is acyclic, $\mathcal{G}$ can be cyclic.

Consider now the decomposition of $\mathcal{G}$ into its maximal strongly connected subgraphs. Recall that a strongly connected graph is a directed graph in which the existence of a directed path from vertice v_1 to vertice v_2 implies the existence of another path from v_2 to v_1. A maximal strongly connected subgraph of a graph G is a strongly connected subgraph of G such that no other subgraph of G covering it is strongly connected ([36]). Let g be the number of the maximal strongly connected subgraphs in $\mathcal{G}$, and $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, $\cdots$, $\mathcal{G}_g = (\mathcal{V}_g, \mathcal{E}_g)$ be the set of all these subgraphs. It is easy to prove that the above set of subgraphs is uniquely defined and that

$$\mathcal{V}_1 \bigcup \cdots \bigcup \mathcal{V}_g = \mathcal{V}, \tag{3.1}$$

$$\mathcal{E}_1 \bigcup \cdots \bigcup \mathcal{E}_g \subseteq \mathcal{E}, \tag{3.2}$$

and for every i and j, $1 \leq i < j \leq g$,

$$\mathcal{V}_i \bigcap \mathcal{V}_j = \emptyset, \tag{3.3}$$

$$\mathcal{E}_i \bigcap \mathcal{E}_j = \emptyset. \tag{3.4}$$

As we shall see later on, if i and $j \in \mathcal{V}$ belong to the same strongly connected subgraph of $\mathcal{G}$, then certain services in queues i and j of SQN-G are constrained by one another. On the contrary, if i and j belong to different strongly connected subgraphs, the constraint is oneway. These properties will be made precise in Lemma 3 and Lemma 7 below.

Define the System Graph (SG) of G, which is denoted by $\tilde{\mathcal{G}}$, to be $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$, where

$$\tilde{\mathcal{V}} = \{1, 2, \cdots, g\}$$

(g is the number of maximal strongly connected subgraphs of $\mathcal{G}$ defined above) and

$$\tilde{\mathcal{E}} = \{(v_1, v_2) | v_1, v_2 \in \tilde{\mathcal{V}},\ \exists k_1 \to k_2 \in \mathcal{E},\ k_1 \in \mathcal{G}_{v_1},\ k_2 \in \mathcal{G}_{v_2}\}$$

The SG describes the relations between the maximal strongly connected subgraphs.

Lemma 3

$\tilde{\mathcal{G}}$ is acyclic.

Proof

The proof follows immediately from the definition of the strong connectedness. See [37] for details. □

Let $G_1 = (V_1, E_1), \cdots, G_g = (V_g, E_g)$ be the subgraphs of G composed of the tasks respectively allocated to the set of processors $\mathcal{V}_1, \cdots, \mathcal{V}_g$; more precisely

$$V_h = \{i | i \in V, i \text{ is allocated to processor } k, k \in \mathcal{V}_h\}$$

$$E_h = \{(i,j) | i, j \in V_h,\ i \to j \in E\}$$

where $1 \le h \le g$.

Similar to (3.1)-(3.2), we have

$$V_1 \bigcup \cdots \bigcup V_g = V, \quad (3.5)$$

$$E_1 \bigcup \cdots \bigcup E_g \subseteq E, \quad (3.6)$$

and for every i and j, $1 \le i < j \le g$,

$$V_i \bigcap V_j = \emptyset, \quad (3.7)$$

$$E_i \bigcap E_j = \emptyset. \quad (3.8)$$

Let SQN-G_1, $\cdots$, SQN-G_g respectively denote the SQN's associated with the graphs $G_1, \cdots, G_g$. In these new networks, the tasks receive the same service requirement sequence as in SQN-G and have the same arrival pattern $\{a_n\}_0^\infty$. $w_n^{k,i}$ ($1 \le k \le K$, $n = 0, 1, 2, \cdots$) denotes the workload of queue k ($k \in \mathcal{V}_i, 1 \le i \le g$) in SQN-$G_i$ at the n-th arrival.

Corollary 1

Assume that SQN-G_i is empty at time 0. For every n ($n \ge 0$) and k ($k \in \mathcal{V}_i$),

$$w_0^{k,i} = 0 \quad (3.9)$$

$$w_{n+1}^{k,i} = \max(0, \max_{j \in \Pi_{\mathcal{G}_i}(k)} (w_n^{j,i} + l_n^{j,k}) - \tau_n) \quad (3.10)$$

where $l_n^{j,k}$ is given by (2.18).

Proof

In view of Theorem 2, the only property to be shown is that for $k \in \mathcal{V}_i$ and $j \in \Pi_{\mathcal{G}_i}(k)$, the values of $l_n^{j,k}$ coincide when computed in G and G_i, namely, with obvious notations, $\mathcal{P}_G(j,k) = \mathcal{P}_{G_i}(j,k)$. Observe first that for k and j as above, j belongs necessarily to $\mathcal{V}_i$. Hence, the existence of a vertice v of G such that v belongs to one of the paths of $\mathcal{P}(j,k)$ implies that $v \in \mathcal{V}_i$, which completes the proof. □

Since $\Pi_{\mathcal{G}_i}(k) = A(\Pi_{G_i}(t_e^k)) \subseteq A(\Pi_G(t_e^k))$, we get immediately

Corollary 2

For all $n \geq 0$ and $k \in \mathcal{V}_i$

$$w_n^k \geq w_n^{k,i}$$

Proof

The proof is by induction on n. □

Definition 1
In the sequel, a queue, say queue k, of the SQN-G (resp. G_i) is said to be stable if w_n^k (resp. $w_n^{k,i}$) converges weakly to a finite Random Variable (RV) w_∞^k (resp. $w_\infty^{k,i}$) when n goes to ∞. The network is said to stable iff all its queues are stable.

We are now going to establish the relation between the stability condition of SNQ-G and that of SNQ-$G_1, \cdots,$ SQN-G_g. This relation is stated in Theorem 3 below which indicates that for analyzing the stability condition of SQN-G, it is enough to consider the networks SQN-$G_1, \cdots,$ SQN-G_g in isolation. The stability condition of SQN-G is simply the intersection of the stability conditions of all these networks. The discussion is organized in preliminary lemmas of independent interest that will be referenced in the sequel.

Throughout the section, the following assumption will be made as the basis for our discussion on stability conditions.

> A_1 : *The sequence $\{\tau_n, \sigma_n^i, 1 \leq i \leq |V|\}_0^\infty$ on $(R^+)^{1+|V|}$ forms a stationary and ergodic sequence of integrable RV's on the probability space (Ω, F, P).*

The basic idea for analyzing the stability conditions of the SQN will consist in generalizing the schema of Loynes for the response time of a $G/G/1$ queue [38], to the waiting times w_n^k, $1 \leq k \leq K$, in SQN-G and $w_n^{k,i}$, $k \in \mathcal{V}_i$ in SQN-G_i. We assume that the sequence $\{\tau_n, \sigma_n^i, 1 \leq i \leq |V|\}_0^\infty$ is the right half of a certain bi-infinite sequence $\{\tau_n, \sigma_n^i, 1 \leq i \leq |V|\}_{-\infty}^\infty$ on (Ω, F, P). (Ω, F, P) is assumed to be the canonical space of these sequences. Let θ denote the leftshift operator on this canonical space. Within this framework, our stationarity assumptions translate into the hypothesis that P is θ-invariant (stationarity) and θ-ergodic. For more details on this formalism, see [39].

Let $\tau = \tau_0$, and $l^{j,k} = l_0^{j,k}$. Consider the schemas $\{M_n^k\}_{n=0}^\infty$, $1 \leq k \leq K$ and $\{M_n^{k,i}\}_{n=0}^\infty$, $k \in \mathcal{V}_i$ defined by

$$M_0^k = 0 \tag{3.11}$$

$$M_{n+1}^k \circ \theta = \max(0, \max_{j \in A(\Pi_G(t_e^k))} (M_n^j + l^{j,k} - \tau)) \tag{3.12}$$

and, for $k \in \mathcal{V}_i$, $1 \leq i \leq g$,

$$M_0^{k,i} = 0 \tag{3.13}$$

$$M_{n+1}^{k,i} \circ \theta = \max(0, \max_{j \in A(\Pi_{G_i}(t_e^k))} (M_n^{j,i} + l^{j,k} - \tau)) \tag{3.14}$$

Lemma 4

For every k, $(1 \leq k \leq K)$, the sequences $\{M_n^k\}_{n=0}^\infty$ and $\{M_n^{k,i}\}_{n=0}^\infty$ $(k \in \mathcal{V}_i)$ are increasing in n.

Proof

We prove the assertion by induction on n. It is clear that for every k, $(1 \leq k \leq K)$, $M_1^k \geq 0 = M_0^k$. Assume now that for some $n \geq 1$, $M_n^k \geq M_{n-1}^k$ holds for every k, $(1 \leq k \leq K)$. Then for any k, $(1 \leq k \leq K)$,

$$\begin{aligned} M_{n+1}^k \circ \theta &= \max(0, \max_{j \in A(\Pi_G(t_e^k))} (M_n^j + l^{j,k} - \tau)) \\ &\geq \max(0, \max_{j \in A(\Pi_G(t_e^k))} (M_{n-1}^j + l^{j,k} - \tau)) = M_n^k \circ \theta \end{aligned}$$

By induction, the RV's $\{M_n^k\}_{n=0}^{\infty}$ are increasing in n. The proof for $\{M_n^{k,i}\}_{n=0}^{\infty}$ is similar. □

Similar to Corollary 2, we have

Corollary 3

For every n and k, $n \geq 0$, $1 \leq k \leq K$, $k \in \mathcal{V}_i$,

$$M_n^k \geq M_n^{k,i}$$

Proof

The proof is by induction on n. □

Lemma 5

For every n and k, $n \geq 0$, $1 \leq k \leq K$,

$$w_n^k = M_n^k \circ \theta^n$$

and

$$w_n^{k,i} = M_n^{k,i} \circ \theta^n$$

Proof

The proof is by induction on n. It is based on the fact that $\tau_n = \tau \circ \theta^n$, and $l_n^{j,k} = l^{j,k} \circ \theta^n$ for $n \geq 0$. □

Lemma 6

For every n and k, $n \geq 1$, $1 \leq k \leq K$,

$$M_n^k = \max(0, \max_{1 \leq m \leq n} (H_m^k - \sum_{i=1}^{m} \tau \circ \theta^{-i})) \tag{3.15}$$

where

$$H_m^k = \max_{1 \leq j \leq K} H_m^{j,k} \tag{3.16}$$

and

$$H_m^{j,k} = \max_{\{(i_1,\cdots,i_{m+1}) | i_1 = j,\ i_{m+1} = k,\ i_s \in A(\Pi_G(t_e^{i_s+1}))\}} (\sum_{s=1}^{m} l^{i_s,i_{s+1}} \circ \theta^{-s}) \tag{3.17}$$

Proof

We show the property by induction on n. For $n = 1$, (3.15) follows from (3.11) and (3.12). Suppose it holds for n. Then, we get from equation (3.12)

$$M_{n+1}^k = \max(0, \max_{j \in A(\Pi_G(t_e^k))} (M_n^j \circ \theta^{-1} + l^{j,k} \circ \theta^{-1} - \tau \circ \theta^{-1}))$$

And by the inductive assumption, we obtain

$$\begin{aligned}
M^k_{n+1} &= \max(0, \max_{j\in A(\Pi_G(t^k_e))} \max(0, \max_{1\le m\le n} \\
&\quad (H^j_m \circ \theta^{-1} - \sum_{i=2}^{m+1} \tau\circ\theta^{-i})) + l^{j,k}\circ\theta^{-1} - \tau\circ\theta^{-1})) \\
&= \max(0, \max_{j\in A(\Pi_G(t^k_e))} \max_{1\le m\le n} \\
&\quad (H^j_m \circ \theta^{-1} + l^{j,k}\circ\theta^{-1} - \sum_{i=2}^{m+1} \tau\circ\theta^{-i} - \tau\circ\theta^{-1}), l^{j,k}\circ\theta^{-1} - \tau\circ\theta^{-1}) \\
&= \max(0, \max_{1\le m\le n} \max_{j\in A(\Pi_G(t^k_e))} \\
&\quad (H^j_m \circ \theta^{-1} + l^{j,k}\circ\theta^{-1} - \sum_{i=1}^{m+1} \tau\circ\theta^{-i}), l^{j,k}\circ\theta^{-1} - \tau\circ\theta^{-1}) \\
&= \max(0, \max_{1\le m\le n} (H^k_{m+1} - \sum_{i=1}^{m+1} \tau\circ\theta^{-i}), H^k_1 - \tau\circ\theta^{-1}) \\
&= \max(0, \max_{1\le m\le n+1} (H^m_k - \sum_{i=1}^{m} \tau\circ\theta^{-i}))
\end{aligned}$$

Therefore the equation holds for $n+1$, which proves the lemma. □

Let M^k_∞ (resp. $M^{k,i}_\infty$) be the limiting value of the increasing sequence M^k_n (resp. $M^{k,i}_n$) when n goes to ∞.

From Lemma 4 and equations (3.12) and (3.14), we get the pathwise equations satisfied by the limiting variables M^k_∞ and $M^{k,i}_\infty$:

$$M^k_\infty \circ\theta = \max(0, \max_{j\in A(\Pi_G(t^k_e))} (M^j_\infty + l^{j,k} - \tau)) \tag{3.18}$$

and, for $k \in \mathcal{V}_i$, $1\le i\le g$,

$$M^{k,i}_\infty \circ\theta = \max(0, \max_{j\in A(\Pi_{G_i}(t^k_e))} (M^{j,i}_\infty + l^{j,k} - \tau)) \tag{3.19}$$

Furthermore, from Lemma 6 we have

Corollary 4

For each k, $1\le k\le K$, the event $\{M^k_\infty = \infty\}$ (resp. the event $\{M^{k,i}_\infty = \infty\}$) is θ-invariant.

Proof

The fact that $M^k_n \uparrow \infty$ is equivalent to

$$\forall X>0 \;\; P[\bigcup_{n\ge 0}\bigcap_{m\ge n} \{M^k_m > X\}] = 1$$

which, according to the expression of M^k_n given by Lemma 6, is equivalent to

$$\forall X>0 \;\; P[\bigcup_{n\ge 0}\bigcap_{m\ge n} \{M^k_m \circ\theta > X\}] = 1$$

in view of the θ-invariance of P. □

Corollary 4 together with the ergodicity assumption on P immediately imply

Corollary 5

For each k, $1 \leq k \leq K$, the event $\{M_\infty^k = \infty\}$ (resp. the event $\{M_\infty^{k,i} = \infty\}$) is either of probability 0 or 1.

From equations (3.18) and (3.19) we also obtain the following two corollaries:

Corollary 6

For every i, $1 \leq i \leq g$, either $M_\infty^k < \infty$ for all k, $k \in \mathcal{V}_i$, or $M_\infty^k = \infty$ for all k, $k \in \mathcal{V}_i$.

Proof

Suppose there is a k, $k \in \mathcal{V}_i$, $1 \leq i \leq g$, such that $M_\infty^k = \infty$, then for all h, $h \in \mathcal{V}_i$, we have

$$\begin{aligned} M_\infty^h \circ \theta &= \max(0, \max_{j \in A(\Pi_G(t_e^h))} (M_\infty^j + l^{j,h} - \tau)) \\ &\geq \max(0, (M_\infty^k + l^{k,h} - \tau)) \end{aligned}$$

so that $M_\infty^h \circ \theta = \infty$. Corollary 4 yields $M_\infty^h = \infty$, which completes the proof. □

Corollary 7

For every i, $1 \leq i \leq g$, either $M_\infty^{k,i} < \infty$ for all k, $k \in \mathcal{V}_i$, or $M_\infty^{k,i} = \infty$ for all k, $k \in \mathcal{V}_i$.

Proof

The proof is similar to that of the preceding corollary. □

Lemma 7

Assume A_1 holds. Then, in SQN-G, for every i, $1 \leq i \leq g$, either all the queues $i_k \in \mathcal{V}_i$ are stable, or they are all unstable.

Proof

The proof is based on Corollaries 5 and 6. Let $k \in \mathcal{V}_i$. Owing to Corollary 5, either $P[M_\infty^k = \infty] = 1$ or $P[M_\infty^k = \infty] = 0$.

If $P[M_\infty^k = \infty] = 1$, Corollary 6 entails $P[\bigcap_{h \in \mathcal{V}_i} \{M_\infty^h = \infty\}] = 1$, so that almost surely (a.s.), M_n^h tends to infinity with n. This together with Lemma 5 and the assumed θ-invariance of P show that none of the RV's w_n^h, $h \in \mathcal{V}_i$ converges weakly.

If $P[M_\infty^k < \infty] = 1$, it follows from Corollary 6 that $P[\bigcap_{h \in \mathcal{V}_i} \{M_\infty^h < \infty\}] = 1$, which in turn entails that all queues of $\mathcal{V}_i$ are stable, since for all $h \in \mathcal{V}_i$, w_n^h is equivalent in law to M_n^h which converges a.s. to a finite limit. □

Corollary 8

Assume A_1 holds. Then, in every SQN-G_i, $1 \leq i \leq g$, either all the queues of SQN-G_i are stable, or they are all unstable.

Proof

The proof follows from Lemma 5 and Corollary 7. □

Lemma 8

Let i, $1 \leq i \leq g$ be fixed. Assume that for all $k \in \mathcal{V}_i$ and for all j, $j \in A(\Pi_G(t_e^k) - \Pi_{G_i}(t_e^k))$, $M_\infty^j < \infty$. If for all $k \in \mathcal{V}_i$, $M_\infty^{k,i} < \infty$, then $M_\infty^k < \infty$.

Proof

If $A(\Pi_G(t_e^k) - \Pi_{G_i}(t_e^k)) = \emptyset$, the conclusion is obvious since $M_n^k = M_n^{k,i}$ holds for all $n \geq 0$ and for all $k \in \mathcal{V}_i$.

If $A(\Pi_G(t_e^k) - \Pi_{G_i}(t_e^k)) \neq \emptyset$, we prove the lemma by reduction to absurdity. Suppose $M_\infty^k = \infty$. By Lemma 4 and Corollary 6, $M_n^h \uparrow \infty$ holds true for all h, $h \in \mathcal{V}_i$.

By assumption, $M_n^j \uparrow M_\infty^j < \infty$ holds for all j, $j \in A(\Pi_G(t_e^k) - \Pi_{G_i}(t_e^k))$. On the other hand, $A(\Pi_{G_i}(t_e^k)) \subseteq \mathcal{V}_i$. Therefore there exists a Z such that for every $n \geq Z$, and every $h \in \mathcal{V}_i$,

$$M_n^j + l^{j,h} < M_n^k + l^{j,h}, \qquad j \in A(\Pi_G(t_e^k) - \Pi_{G_i}(t_e^k))$$

Thus for every $h \in \mathcal{V}_i$, and $n > Z$,

$$\begin{aligned} M_{n+1}^h \circ \theta &= \max(0, \max_{j \in A(\Pi_G(t_e^h))} (M_n^j + l^{j,h} - \tau)) \\ &= \max(0, \max_{j \in A(\Pi_{G_i}(t_e^h))} (M_n^j + l^{j,h} - \tau)) \end{aligned}$$

Let $U = \max_{h \in \mathcal{V}_i} (M_Z^h - M_Z^{h,i})$, we get

$$\begin{aligned} M_{Z+n+1}^h \circ \theta &= \max(0, \max_{j \in A(\Pi_{G_i}(t_e^h))} (M_{Z+n}^j + l^{j,h} - \tau)) \\ &\leq \max(0, \max_{j \in A(\Pi_{G_i}(t_e^h))} (M_{Z+n}^{j,i} + U + l^{j,h} - \tau)) \\ &\leq U + \max(0, \max_{j \in A(\Pi_{G_i}(t_e^h))} (M_{Z+n}^{j,i} + l^{j,h} - \tau)) \\ &= U + M_{Z+1}^{h,i} \circ \theta \end{aligned}$$

so that

$$M_{Z+1}^h \circ \theta \leq U + M_{Z+1}^{h,i} \circ \theta$$

We can easily prove by induction that for $n \geq 0$ and $h \in \mathcal{V}_i$,

$$M_{Z+n}^h \circ \theta^n \leq U + M_{Z+n}^{h,i} \circ \theta^n$$

so that for all $X > 0$

$$\begin{aligned} P[M_{Z+n}^h > X] &\leq P[U + M_{Z+n}^{h,i} \circ \theta^n > X] \\ &\leq P[U > \frac{X}{2}] + P[M_{Z+n}^{h,i} > \frac{X}{2}] \end{aligned}$$

Letting n go to ∞ in the preceding relation yields

$$P[M_\infty^h > X] \leq P[U > \frac{X}{2}] + P[M_\infty^{h,i} > \frac{X}{2}]$$

where we have used the increasingness of the Loynes' schemas to permute the limits and the expectations. Owing to the assumption that $M_\infty^h = \infty$ a.s., it follows that $\lim_{X \to \infty} P[M_\infty^h > X] = 1$. Similarly, the finiteness of the RV U entails $\lim_{X \to \infty} P[U > \frac{X}{2}] = 0$, so that taking the limit in X in the last equation yields

$$P[M_\infty^{h,i} = \infty] = \lim_{X \to \infty} P[M_\infty^{h,i} > X] = 1$$

which contradicts our assumption that $M_n^{k,i} \uparrow M_\infty^{k,i} < \infty$ a.s..

Hence we reach the conclusion that $M_\infty^k < \infty$ for every $k \in \mathcal{V}_i$. □

Corollary 9

If for every k, $1 \le k \le K$, $M_\infty^{k,i} < \infty$, then for every k, $1 \le k \le K$, $M_\infty^k < \infty$.

Proof

According to Lemma 3, $\tilde{\mathcal{G}}$ is acyclic. We can therefore label the nodes of $\tilde{\mathcal{G}}$ as $1, \cdots, g_0, g_0+1, \cdots, g$ in such a way that $i \to j \in \tilde{\mathcal{E}}$ implies $i < j$, and nodes $1, \cdots, g_0$ have no predecessor in $\tilde{\mathcal{G}}$.

The proof of the corollary is by induction on i, $1 \le i \le g$.

Consider all i, $1 \le i \le g_0$. Since i has no predecessor in $\tilde{\mathcal{G}}$, for every $k \in \mathcal{V}_i$, k has thus no other predecessor than the elements of $\mathcal{V}_i$ in the processor graph $\mathcal{G}$. Therefore $\Pi_{\mathcal{G}}(t_e^k) = \Pi_{\mathcal{G}_i}(t_e^k)$. From Lemma 8, we obtain $M_\infty^k < \infty$.

Consider now i, $g_0 < i \le g$, assume that for all j, $j < i$, $M_\infty^k < \infty$ is true for all $k \in \mathcal{V}_i$. Then the fact that for all $k \in \mathcal{V}_i$, $M_\infty^k < \infty$ is an immediate consequence of Lemma 8 due to the fact that $A(\Pi_{\mathcal{G}}(t_e^k) - \Pi_{\mathcal{G}_i}(t_e^k)) \subseteq \{1, 2, \cdots, i-1\}$. The assertion is thus proved. □

Now we are in position to prove the following important result.

Theorem 3

Assume A_1 holds. Then SQN-G is stable iff for all $1 \le i \le g$, SQN-G_i is stable.

Proof

The assertion of the theorem follows immediately from Corollary 9, Corollary 4 and Lemma 5. □

It is well known ([39]) that, under the foregoing assumptions, if SQN-G_i consists of a single queue, say queue k, then the stability condition reads

$$E[\sum_{\{j \mid A(j)=k\}} \sigma^j] < E[\tau]$$

where A(j) denotes the index of the processor to which task j is allocated. From this we get

Corollary 10

Assume A_1 holds. If $\mathcal{G}$ is acyclic, and for every k, $1 \le k \le K$,

$$E[\sum_{\{j \mid A(j)=k\}} \sigma^j] < E[\tau],$$

then w_n^k converges weakly to a finite and integrable RV w_∞^k when n goes to ∞.

The remainder of this section is concerned with the stability condition of SQN's whose processor graphs are strongly connected. Owing to Theorem 3, such stability condition will provide immediately the stability condition of SQN-G. Without loss of generality, it will be assumed that $\mathcal{G}$ is strongly connected.

Let $L^{j,k}$ $(1 \le j, k \le K)$ be defined as

$$L^{j,k} = \max_{\{i_s \mid s=0,\cdots,K, i_0=j, i_K=k, i_s \in A(\Pi_{\mathcal{G}}(t_e^{i_s+1}))\}} \sum_{s=0}^{K-1} l^{i_s, i_{s+1}} \circ \theta^{-s} \tag{3.20}$$

Let

$$Q_n = E[\max_{1 \le v_1, \cdots, v_{n+1} \le K} (\sum_{i=1}^{n} L^{v_i, v_{i+1}} \circ \theta^{-iK})] \tag{3.21}$$

where the finiteness of Q_n follows from the integrability assumption on the service times (use the fact that $\max(a,b) \leq a+b$), and

$$U_n = \frac{1}{n} Q_n. \quad (3.22)$$

Lemma 9

U_n tends to a limit γ when n goes to ∞ :

$$\exists \lim_{n \to \infty} U_n = \gamma \quad (3.23)$$

Proof
For all $n \geq 1$, and all $p, q \geq 1$ such that $p+q=n$, we have

$$\begin{aligned}
Q_n &= E[\max_{1 \leq v_1, \cdots, v_{n+1} \leq K} (\sum_{i=1}^{n} L^{v_i, v_{i+1}} \circ \theta^{-iK})] \\
&\leq E[\max_{1 \leq v_1, \cdots, v_{p+1} \leq K} (\sum_{i=1}^{p} L^{v_i, v_{i+1}} \circ \theta^{-iK})] \\
&\quad + E[\max_{1 \leq v_{p+1}, \cdots, v_{n+1} \leq K} (\sum_{i=p+1}^{n} L^{v_i, v_{i+1}} \circ \theta^{-iK})] \\
&= E[\max_{1 \leq v_1, \cdots, v_{p+1} \leq K} (\sum_{i=1}^{p} L^{v_i, v_{i+1}} \circ \theta^{-iK})] \\
&\quad + E[\max_{1 \leq v_1, \cdots, v_{q+1} \leq K} (\sum_{i=1}^{q} L^{v_i, v_{i+1}} \circ \theta^{-iK}) \circ \theta^{-pK}] \\
&= E[\max_{1 \leq v_1, \cdots, v_{p+1} \leq K} (\sum_{i=1}^{p} L^{v_i, v_{i+1}} \circ \theta^{-iK})] \\
&\quad + E[\max_{1 \leq v_1, \cdots, v_{q+1} \leq K} (\sum_{i=1}^{q} L^{v_i, v_{i+1}} \circ \theta^{-iK})] \\
&= Q_p + Q_q
\end{aligned}$$

which yields

$$Q_n \leq Q_p + Q_q$$

The function Q_n is hence sub-additive. It is well known that this property entails that $U_n = Q_n n^{-1}$ tends to a limit when n goes to ∞, as stated in (3.23). For a simple proof, see for instance [33]. □

Now we are in position to prove the main results of this section.

Theorem 4

•

Assume A_1 holds, and that the processor graph of G, $\mathcal{G}$, is strongly connected. If

$$\gamma < KE[\tau]$$

where γ is defined in (3.23). then the distribution functions of the RV's $\vec{w}_n$ converge weakly to a finite RV $\vec{w}_\infty$ when n goes to ∞. If

$$\gamma > KE[\tau]$$

then, the RV's $\vec{w}_n$ converge a.s. to ∞.

Proof

In order to prove the first part of the theorem, it is enough to prove that under the first condition, the RV's $\vec{M}_n$ increase pathwise to a finite limit $\vec{M}_\infty$ when n goes to ∞ ($\vec{w}_n$ and $\vec{M}_n$ are equivalent in law for all $n \geq 0$).

From Lemma 7, either $M_n^k \uparrow M_\infty^k < \infty$ a.s. for all $1 \leq k \leq K$, or $M_n^k \uparrow \infty$ a.s. for all $1 \leq k \leq K$.

Assume we are in the later case. From (3.12), we get the following relation holding for all $N > 0$ and $n \geq 1$

$$M_{N+n}^k = \max(0, \max_{j \in A(\Pi_G(t_e^k))} (M_{N+n-1}^j \circ \theta^{-1} + l^{j,k} \circ \theta^{-1}) - \tau \circ \theta^{-1})$$

Owing to the assumption $M_n^k \uparrow \infty$, there exists an N_0 such that for all $n > 0$ and $1 \leq k \leq K$, the last relation reduces to

$$M_{N_0+n}^k = \max_{j \in A(\Pi_G(t_e^k))} (M_{N_0+n-1}^j \circ \theta^{-1} + l^{j,k} \circ \theta^{-1} - \tau \circ \theta^{-1})$$

After simple manipulations on the above equation, we get by induction on n that

$$M_{N_0+n}^k = \max_{j \in A(\Pi_G^n(t_e^k))} (M_{N_0}^j \circ \theta^{-n} + H_n^{j,k}) - \sum_{i=1}^n \tau \circ \theta^{-i}$$

for all $n > 0$, where the function Π_G^n was defined at the beginning of section 2 and $H_n^{j,k}$ in Lemma 6.

As $\mathcal{G}$ is strongly connected, we have, for all $1 \leq k \leq K$,

$$A(\Pi_G^n(t_e^k)) = \{1, \cdots, K\}$$

when $n \geq K$. so that for $n \geq K$

$$M_{N_0+n}^k = \max_{1 \leq j \leq K} (M_{N_0}^j \circ \theta^{-n} + H_n^{j,k}) - \sum_{i=1}^n \tau \circ \theta^{-i}$$

This implies that

$$\max_{1 \leq k \leq K} M_{N_0+n}^k \leq \max_{1 \leq j \leq K} M_{N_0}^j \circ \theta^{-n} + \max_{1 \leq j,k \leq K} H_n^{j,k} - \sum_{i=1}^n \tau \circ \theta^{-i}$$

which can be rewritten as

$$\max_{1 \leq j,k \leq K} H_n^{j,k} - \sum_{i=1}^n \tau \circ \theta^{-i} \geq \max_{1 \leq k \leq K} M_{N_0+n}^k - \max_{1 \leq j \leq K} M_{N_0}^j \circ \theta^{-n}$$

Hence

$$\max_{1 \leq j,k \leq K} H_n^{j,k} - \sum_{i=1}^n \tau \circ \theta^{-i} \geq \limsup_{N \to \infty} (\max_{1 \leq k \leq K} M_{N+n}^k - \max_{1 \leq j \leq K} M_N^j \circ \theta^{-n})$$

One gets from (3.12) that for all $N > 0$,

$$\max_{1 \leq k \leq K} M_{N+1}^k \leq \max_{1 \leq j \leq K} M_N^j \circ \theta^{-1} + \max_{1 \leq j,k \leq K} l_N^{j,k} \circ \theta^{-1}$$

which implies that for all $n \geq 1$, the RV's

$$\{\max_{1 \leq k \leq K} M_{N+n}^k - \max_{1 \leq j \leq K} M_N^j \circ \theta^{-n}\}_{N=0}^\infty$$

are uniformly bounded in N by an integrable RV, so that owing to Lebesgue's Theorem,

$$\limsup_{N \to \infty} E[(\max_{1 \leq k \leq K} M_{N+n}^k - \max_{1 \leq j \leq K} M_N^j \circ \theta^{-n})] \leq E[\limsup_{N \to \infty} (\max_{1 \leq k \leq K} M_{N+n}^k - \max_{1 \leq j \leq K} M_N^j \circ \theta^{-n})]$$

Owing to the integrability of the RV's M_n^j, we also have

$$E[(\max_{1\le k\le K} M_{N+n}^k - \max_{1\le j\le K} M_N^j \circ \theta^{-n})] = E[(\max_{1\le k\le K} M_{N+n}^k - \max_{1\le j\le K} M_N^j)] \ge 0$$

where the last inequality follows from the increasingness of M_n^k for all $1 \le k \le K$. hence,

$$\begin{aligned} 0 &\le \limsup_N E[\max_{1\le k\le K} M_{N+n}^k - \max_{1\le j\le K} M_N^j \circ \theta^{-n}] \\ &\le E[\limsup_N(\max_{1\le k\le K} M_{N+n}^k - \max_{1\le j\le K} M_N^j \circ \theta^{-n})] \\ &\le E[\max_{1\le j,k\le K} H_n^{j,k} - \sum_{i=1}^n \tau \circ \theta^{-i}] \end{aligned}$$

We obtain finally

$$E[\max_{1\le j,k\le K} H_n^{j,k} - \sum_{i=1}^n \tau \circ \theta^{-i}] \ge 0 \tag{3.24}$$

Consider now the expression of $\max_{1\le j,k\le K} H_n^{j,k}$. Let $n = mK + c$, where $m \ge 0$, $0 \le c < K$. $\max_{1\le j,k\le K} H_n^{j,k}$ can be rewritten as

$$\begin{aligned} \max_{1\le j,k\le K} H_n^{j,k} &= \max_{\{i_s|s=1,\cdots,n+1, i_s\in A(\Pi_G(t_e^{i_s+1}))\}} \sum_{s=1}^n l^{i_s,i_{s+1}} \circ \theta^{-s} \\ &= \max_{\{i_s|s=1,\cdots,n+1, i_s\in A(\Pi_G(t_e^{i_s+1}))\}} (\sum_{s=1}^{mK} l^{i_s,i_{s+1}} \circ \theta^{-s} + \sum_{s=mK+1}^{n} l^{i_s,i_{s+1}} \circ \theta^{-s}) \end{aligned}$$

Observe that

$$\max_{1\le j,k\le K} H_{mK}^{j,k} \le \max_{1\le j,k\le K} H_n^{j,k} \le \max_{1\le j,k\le K} H_{(m+1)K}^{j,k} \tag{3.25}$$

and that $\max_{1\le j,k\le K} H_{mK}^{j,k}$ is expressed as

$$\max_{1\le j,k\le K} H_{mK}^{j,k} = \max_{\{i_s|s=1,\cdots,mK+1, i_s\in A(\Pi_G(t_e^{i_s+1}))\}} (\sum_{h=0}^{m-1} \sum_{s=hK+1}^{(h+1)K} l^{i_s,i_{s+1}} \circ \theta^{-s})$$

Using the fact that for all $1 \le h \le m$,

$$\{1, \cdots, K\} = A(P_G^K(t_e^{i_{(h+1)K}}))$$

we can rewrite $\max_{1\le j,k\le K} H_{mK}^{j,k}$ as follows

$$\begin{aligned} \max_{1\le j,k\le K} H_{mK}^{j,k} &= \max_{\{\{i_s|s=1,\cdots,mK+1, i_s\in A(\Pi_G(t_e^{i_s+1}))\}} \sum_{s=1}^{mK} l^{i_s,i_{s+1}} \circ \theta^{-s} \\ &= \max_{\{1\le i_1, i_{K+1},\cdots,i_{mK+1}\le K\}} \max_{\{\{i_s|s=hK+1,\cdots,(h+1)K,\ h=0,\cdots,m-1,\ i_s\in A(\Pi_G(t_e^{i_s+1}))\}} \\ &\quad \sum_{h=0}^{m-1} \sum_{s=hK+1}^{(h+1)K} l^{i_s,i_{s+1}} \circ \theta^{-s} \\ &= \max_{\{1\le i_1, i_{K+1},\cdots,i_{mK+1}\le K\}} \sum_{h=0}^{m-1} \\ &\quad \max_{\{\{i_s|s=hK+1,\cdots,(h+1)K,\ h=0,\cdots,m-1,\ i_s\in A(\Pi_G(t_e^{i_s+1}))\}} \sum_{s=hK+1}^{(h+1)K} l^{i_s,i_{s+1}} \circ \theta^{-s} \\ &= \max_{\{1\le v_0,v_1,\cdots,v_m\le K\}} \sum_{h=0}^{m-1} L^{v_h,v_{h+1}} \circ \theta^{-hK} \end{aligned}$$

so that

$$E[\max_{1\le j,k\le K} H^{j,k}_{mK}] = Q_m$$

Using (3.24) and this lastequation, we get

$$\begin{aligned}\lim_{n\to\infty} \frac{1}{n} E[\max_{1\le j,k\le K} H^{j,k}_{n}] &= \lim_{m\to\infty} \frac{1}{mK} E[\max_{1\le j,k\le K} H^{j,k}_{mK}] \\ &= \lim_{m\to\infty} \frac{1}{mK} Q_m \\ &= \frac{\gamma}{K}\end{aligned}$$

where γ is defined in (3.23).

Using the θ-invariance of P in equation (3.24) yields

$$E[\max_{1\le j,k\le K} H^{j,k}_{n}] - \sum_{i=1}^{n} E[\tau] \ge 0$$

Dividing by n each side of the last inequality, we get

$$\frac{E[\max_{1\le j,k\le K} H^{j,k}_{n}]}{n} - E[\tau] \ge 0$$

This inequality is preserved when n tends to ∞:

$$\lim_{n\to\infty} \frac{1}{n} E[\max_{1\le j,k\le K} H^{j,k}_{n}] \ge E[\tau]$$

which entails

$$\gamma \ge KE[\tau].$$

Now taking the contrapositive of this argument, we see that

$$\gamma < KE[\tau]$$

is sufficient to have $\vec{M}_\infty$, the limiting value of random vector $\vec{M}_n$, finite almost everywhere. This completes the proof of the first part of the theorem.

As for the second part of the theorem, assume

$$\gamma > KE[\tau]$$

Let $\delta = \frac{\gamma}{K} - E[\tau] > 0$ and $\epsilon = \frac{\delta}{4} > 0$. From Lemma 9, there exists integer Z_1 such that for all $n \ge Z_1$,

$$|\frac{Q_n}{Kn} - \frac{\gamma}{K}| < \epsilon$$

Let Z_2 be the smallest integer satisfying the relation

$$\frac{2}{Z_2} \cdot \frac{\gamma}{K} \le \epsilon$$

or equivalently

$$Z_2 \ge \frac{2\gamma}{\epsilon K}$$

Now let $Z = \max(Z_1, Z_2)$, and $H^{j,k}_n$ be defined as (3.17). Then for all $1 \le k \le K$,

$$\begin{aligned}\frac{EH^{k,k}_{K(Z+2)}}{K(Z+2)} &= \frac{1}{K(Z+2)} E \max_{1\le v_0,\cdots,v_Z\le K} (L^{k,v_0} + L^{v_0,v_1} \circ \theta^{-K} + \cdots + L^{v_{Z-1},v_Z} \circ \theta^{-ZK} \\ &\quad + L^{v_Z,k} \circ \theta^{-(Z+1)K})\end{aligned}$$

$$
\begin{aligned}
&> \frac{1}{K(Z+2)} E \max_{1\le v_0,\cdots,v_Z\le K} (L^{v_0,v_1}\circ\theta^{-K} + \cdots + L^{v_{Z-1},v_Z}\circ\theta^{-ZK}) \\
&= \frac{Q_Z}{K(Z+2)} \\
&= \frac{Q_Z}{KZ}\cdot\frac{Z}{Z+2} \\
&> \frac{\gamma}{K} - 2\epsilon \\
&= \frac{\delta}{2} + E[\tau]
\end{aligned}
$$

or simply

$$
\frac{E[H^{k,k}_{K(Z+2)}]}{K(Z+2)} - E[\tau] > \frac{\delta}{2} > 0 \tag{3.26}
$$

In addition, using Lemma 6, we get

$$
M^k_n \ge H^k_n - \sum_{i=1}^{n} \tau\circ\theta^{-i}
$$

which implies that for all $n \ge 1$,

$$
\begin{aligned}
M^k_{nK(Z+2)} &\ge H^k_{nK(Z+2)} - \sum_{i=1}^{nK(Z+2)} \tau\circ\theta^{-i} \\
&= \max_{1\le v_0,\cdots,v_{nK(Z+2)}\le K} \sum_{i=0}^{nK(Z+2)} L^{v_i,v_{i+1}}\circ\theta^{-iK} - \sum_{i=1}^{nK(Z+2)} \tau\circ\theta^{-i} \\
&\ge \sum_{h=0}^{n-1} H^{k,k}_{hK(Z+2)}\circ\theta^{-hK(Z+2)} - \sum_{i=1}^{nK(Z+2)} \tau\circ\theta^{-i}
\end{aligned}
$$

or

$$
\frac{M^k_{nK(Z+2)}}{nK(Z+2)} \ge \frac{1}{nK(Z+2)} \sum_{h=0}^{n-1} H^{k,k}_{hK(Z+2)}\circ\theta^{-hK(Z+2)} - \frac{1}{nK(Z+2)} \sum_{i=1}^{nK(Z+2)} \tau\circ\theta^{-i} \tag{3.27}
$$

Owing to the ergodicity assumption, we get

$$
\lim_{n\to\infty} \frac{1}{nK(Z+2)} \sum_{h=0}^{n-1} H^{k,k}_{hK(Z+2)}\circ\theta^{-hK(Z+2)} = \frac{1}{K(Z+2)} E[H^{k,k}_{K(Z+2)}]
$$

and

$$
\lim_{n\to\infty} \frac{1}{nK(Z+2)} \sum_{i=1}^{nK(Z+2)} \tau\circ\theta^{-i} = E[\tau]
$$

Together with (3.26) and (3.27) we get

$$
\lim_{n\to\infty} M^k_{nK(Z+2)} = \infty
$$

The monotone increasingness of M^k_n entails in turn

$$
\lim_{n\to\infty} M^k_n = \infty
$$

which implies

$$
\lim_{n\to\infty} w^k_n = \infty \quad a.s.
$$

The proof is therefore completed. □

In case the service requirements of the tasks are deterministic, there is a finite expression for γ.

Theorem 5

Assume that the service pattern $\{\sigma_n^i,\ 1 \leq i \leq |V|\}_{n=0}^{\infty}$ *is deterministic, namely* $\sigma_n^i = \sigma^i,\ 1 \leq i \leq |V|,\ n \geq 0$. *Then*

$$\gamma = \nu$$

where

$$\nu = \max_{1\leq d\leq K}\left(\frac{1}{d}\max_{1\leq i_1,\cdots,i_d\leq K}(L^{i_1,i_2} + \cdots + L^{i_{k-1},i_k} + L^{i_d,i_1})\right)$$

Proof
See Appendix 1. □

4 Distribution Functions of Waiting and Response Times

This section is devoted to the analytical characterization of the transient and stationary distributions of the waiting and response times. Throughout this section, it will be assumed that

A_2:
The RV's $\{\tau_n\}_{n=0}^{\infty}$ *are i.i.d.*
The RV's $\{\sigma_n^i\}_{n=0}^{\infty}$ *are i.i.d. for all* $1 \leq i \leq |V|$.
$\{\tau_n, \sigma_n^i, 1 \leq i \leq |V|\}_{n=0}^{\infty}$ *is a set of mutually independent RV's.*

Denote as Σ^i $(1 \leq i \leq |V|)$ and T the respective common distribution functions of the RV's σ_n^i and τ_n. Let

$$\Lambda = \Lambda(y_{1,1}, \cdots, y_{j,k}, \cdots, y_{K,K})$$

be the joint distribution function in R^Q of the Q RV's $l_n^{j,k}$ $(1 \leq j,k \leq K,\ j \in A(\Pi_G(t_e^k)))$, where $Q = \sum_{1\leq k\leq K} |\{j,\ 1 \leq j \leq K,\ j \in A(\Pi_G(t_e^k))\}|$. Note that this joint distribution function can be obtained by PERT techniques. We give below a simple integral representation for Λ:

$$\Lambda(y_{1,1}, \cdots, y_{j,k}, \cdots, y_{K,K}) = \int \cdots \int_{\Phi} d\Sigma^1(u_1) \cdots d\Sigma^{|V|}(u_{|V|})$$

where Φ is the following convex subset of ${R^+}^{|V|}$:

$$\Phi = \left(\bigcap_{i=1}^{|V|} u_i \geq 0\right) \bigcap \left(\bigcap_{\{j,k|\mathcal{P}(t_b^j,t_e^k)\neq\emptyset\}} \ \bigcap_{\{v_{p_1},\cdots,v_{p_m}\in\mathcal{P}(t_b^j,t_e^k)\}} \left(\sum_{i=1}^{m} u_{p_i} \leq y_{j,k}\right)\right)$$

This integral representation can be proved as in Theorem 6 below.

For sake of simplicity, it will be assumed that all the distribution functions Σ^i, $(1 \leq i \leq |V|)$ have a density on R^{+*} and have no mass at the origin. Observe that in view of the preceding integral representation, this entails that Λ has a density on ${R^{+*}}^Q$ and no mass on the boundaries of this domain. It is important to notice that the various distribution functions that were introduced so far are defined as distribution functions on the whole real line (or space) although their support is actually on the positive part of the line (or the positive orthant).

Now let

$$W_n = W_n(x_1, \cdots, x_K)$$

and

$$D_n = D_n(z_1, \cdots, z_K)$$

denote the distribution functions on R^K defined by the the following set of integral equations

$$W_0(x_1,\cdots,x_K) = U(x_1,\cdots,x_K) \tag{4.1}$$

$$W_{n+1}(x_1,\cdots,x_K) = U(x_1,\cdots,x_K)\cdot\int_{-\infty}^{\infty}(\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}W_n(b_1,\cdots,b_K)d\Lambda)dT \tag{4.2}$$

$$D_n(x_1,\cdots,x_K) = U(x_1,\cdots,x_K)\cdot\int_{-\infty}^{\infty}(\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}W_n(c_1,\cdots,c_K)d\Lambda)dT \tag{4.3}$$

where U denotes the step function:

$$U(t_1,\cdots,t_K)=0, \qquad \exists j:\ t_j<0$$
$$U(t_1,\cdots,t_K)=1, \qquad \forall j,\ t_j\geq 0$$

and

$$d\Lambda = \frac{\partial}{\partial y_{1,1}}\cdots\frac{\partial}{\partial y_{j,k}}\cdots\frac{\partial}{\partial y_{K,K}}\Lambda(y_{1,1},\cdots,y_{j,k},\cdots,y_{K,K})$$
$$dT = dT(u)$$

and for $1\leq j\leq K$,

$$b_j = \min_{\{k|j\in A(\Pi_G(t_e^k))\}}(x_k - y_{j,k}+u)$$
$$c_j = b_j - u$$

Theorem 6

For all $n\geq 0$, W_n and D_n are the joint distribution functions of the RV's $(w_n^1,\cdots,w_n^K)$ and $(r_n^1,\cdots,r_n^K)$, respectively.

Proof

We prove the assertion by induction .
For $n=0$, (2.14) and (4.1) yields that W_0 is the joint distribution function of RV's $(w_0^1,\cdots,w_0^K)$.

Assume for some $n\geq 0$, W_n obtained from (4.1) and (4.2) is effectively the joint distribution function of the RV's $(w_n^1,\cdots,w_n^K)$. Then for all $l_n^{j,k}, \tau_n>0$ $(1\leq j,k\leq K)$, and all $x_1,\cdots,x_K\geq 0$, from (2.15)

$$\begin{aligned}
&P[w_{n+1}^1\leq x_1,\cdots,w_{n+1}^K\leq x_K] \\
&= P[\bigcap_{k=1}^{K}\max(0,\max_{\{j\in A(\Pi_G(t_e^k))\}}(w_n^j+l_n^{j,k}-\tau_n))\leq x_k] \\
&= P[\bigcap_{k=1}^{K}\max_{\{j\in A(\Pi_G(t_e^k))\}}(w_n^j+l_n^{j,k}-\tau_n)\leq x_k] \\
&= P[\bigcap_{k=1}^{K}\bigcap_{\{j\in A(\Pi_G(t_e^k))\}}(w_n^j+l_n^{j,k}-\tau_n)\leq x_k] \\
&= P[\bigcap_{j=1}^{K}\bigcap_{\{k|j\in A(\Pi_G(t_e^k))\}}(w_n^j+l_n^{j,k}-\tau_n)\leq x_k] \\
&= P[\bigcap_{j=1}^{K}\bigcap_{\{k|j\in A(\Pi_G(t_e^k))\}}(w_n^j\leq x_k+l_n^{j,k}-\tau_n)] \\
&= P[\bigcap_{j=1}^{K}(w_n^j\leq b_j)]
\end{aligned}$$

which implies that W_{n+1} defined by (4.2) is also the joint distribution function of the RV's $(w_{n+1}^1, \cdots, w_{n+1}^K)$ Therefore, for all $n \geq 0$, the function W_n defined by (4.1)-(4.2) is the joint distribution function of the RV's $(w_n^1, \cdots, w_n^K)$.

Using this fact and the similar arguments, we can prove that for all $n \geq 0$, D_n is the joint distribution function of the RV's $(r_n^1, \cdots, r_n^K)$. □

Remark

Observe that Theorem 4 actually provides the condition under which the random *vector* $\vec{w}_n$ converges weakly to a finite limit $\vec{w}_\infty$ when n tends to ∞. Indeed, under the conditions specified there, $\vec{w}_n =_{st} \vec{M}_n$ where $\vec{M}_n$ converges a.s. to a finite limit. This entails that the distribution functions W_n and D_n respectively converge weakly to finite distribution functions W_∞ and D_∞ when n goes to ∞.

The remainder of this section is devoted to the characterization of the limit distribution functions.

Theorem 7

Assume A_2 holds, and that the stability condition of Theorem 4 is satisfied. Then, the distribution function on R^K of the random vector $\vec{w}_\infty$, which will be denoted as $W_\infty(x_1, \cdots, x_K)$ satisfies the integral equations

$$W_\infty(x_1, \cdots, x_K) = U(x_1, \cdots, x_K) \cdot \int_{-\infty}^{\infty} (\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} W_\infty(b_1, \cdots, b_K) d\Lambda) dT \qquad (4.4)$$

$$D_\infty(x_1, \cdots, x_K) = U(x_1, \cdots, x_K) \cdot \int_{-\infty}^{\infty} (\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} W_\infty(c_1, \cdots, c_K) d\Lambda) dT \qquad (4.5)$$

where U denotes the step function:

$$U(x_1, \cdots, x_K) = 0, \qquad \exists j : x_j < 0$$
$$U(x_1, \cdots, x_K) = 1, \qquad \forall j, x_j \geq 0$$

and

$$d\Lambda = \frac{\partial}{\partial y_{1,1}} \cdots \frac{\partial}{\partial y_{j,k}} \cdots \frac{\partial}{\partial y_{K,K}} \Lambda(y_{1,1}, \cdots, y_{j,k}, \cdots, y_{K,K})$$
$$dT = dT(u)$$

and for $1 \leq j \leq K$,

$$b_j = \min_{\{k | j \in A(\Pi_G(t_e^k))\}} (x_k - y_{j,k} + u)$$

Similarly, the the distribution function on R^K of the random vector $\vec{r}_\infty$, which will be denoted as $D_\infty(x_1, \cdots, x_K)$ is given by

$$D_\infty(x_1, \cdots, x_K) = U(x_1, \cdots, x_K) \cdot \int_{-\infty}^{\infty} (\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} W_\infty(c_1, \cdots, c_K) d\Lambda) dT \qquad (4.6)$$

where

$$c_j = b_j - u$$

Proof

Using the pathwise equations satisfied by the limit RV's

$$w_\infty^k \circ \theta = \max(0, \max_{j \in A(\Pi_G(t_e^k))} (w_\infty^j + l^{j,k} - \tau))$$
$$r_\infty^k \circ \theta = \max(0, \max_{j \in A(\Pi_G(t_e^k))} (w_\infty^j + l^{j,k}))$$

and the arguments of the proof of Theorem 6 yield immediately the results. □

Remarks

1. Observe that the distribution functions $W_n(x_1, \cdots, x_K)$, which are given by the recursive integral schema (4.1), (4.2) converge weakly to $W_\infty(x_1, \cdots, x_K)$ when n goes to ∞. This provides a direct numerical schema for computing the solution of (4.6).

2. The distribution function of the n-th program response time is given by $R_n(z) = D_n(z, \cdots, z)$. Similarly, the stationary distribution function of programs response times is given by $R_\infty(z) = D_\infty(z, \cdots, z)$.

3. Owing to Little's formula, at the steady state, the mean number of programs in queue k at program arrival epochs, $E[N_k]$, is given by

$$E[N_k] = \frac{E[r_\infty^k]}{E[\tau]} \tag{4.7}$$

5 Lower Bounds for Waiting and Response Times Based on Convex Ordering

Exact analytical solutions to the basic integral functional equation (4.6) seem to be rather difficult to obtain. However, computational lower and upper bounds can be derived on the solution of (4.6) as well as on the recursive system (4.1)-(4.5) using simple stochastic ordering techniques. In this section we discuss lower bounds on waiting and response times based on convex ordering.

Recall that two non-negative and integrable RV's b_1 and b_2 are stochastically ordered in the convex increasing sense:

$$b_1 \leq_{ci} b_2$$

iff for all convex increasing functions $f : R^+ \to R^+$ such that the expectation exists,

$$E[f(b_1)] \leq E[f(b_2)].$$

For more detail on the notion of convex ordering, see [40].

Consider the SQN-G described by rules (i)-(iii). We assume that all the RV's $\{\tau_n\}_{n=0}^\infty$ and $\{l_n^{j,k}, 1 \leq j, k \leq K\}_{n=0}^\infty$ are defined on the probability space (Ω, F, P) and are all integrable, and that A_1 holds.

Let $\{\tilde{\tau}_n\}_{n=0}^\infty$ and $\{\tilde{l}_n^{j,k}, 1 \leq j, k \leq K\}_{n=0}^\infty$ be the set of "smoother" inter-arrival and service requirement processes on (Ω, F, P) in the sense that there exists a sub σ-algebra say $\mathcal{H}$ of F such that for all $n \geq 0$ and all $1 \leq j, k \leq K$,

$$\tilde{\tau}_n = E[\tau_n|\mathcal{H}] \qquad a.s.$$
$$\tilde{l}_n^{j,k} = E[l_n^{j,k}|\mathcal{H}] \qquad a.s.$$

These new varibles are smoother than the initial ones in the sense that they have the same first moment as the initial ones but higher moments are always smaller for the new ones than for the initial ones. Indeed let $\tilde{b}$ and b be two non-negative and integrable RV's on (Ω, F, P) such that

$$\tilde{b} = E[b|\mathcal{H}] \qquad a.s.$$

then, for each convex increasing functions $f : R^+ \to R^+$ such that the expectation exists, we have, owing to Jensen's inequality for conditional expectations,

$$f(\tilde{b}) = f(E[b|\mathcal{H}]) \leq E[f(b)|\mathcal{H}] \qquad a.s.$$

which entails

$$E[f(\tilde{b})] \leq E[f(b)].$$

Observe that the above relation implies that for all $n \geq 0$ and $1 \leq j,k \leq K$,

$$\tilde{\tau}_n \leq_{ci} \tau_n$$

and

$$\tilde{l}_n^{j,k} \leq_{ci} l_n^{j,k}$$

Let $\tilde{w}_n^k$, $\tilde{r}_n^k$ and $\tilde{r}_n$ be the waiting time and response time variable obtained with the new arrival and service pattern $\{\tilde{\tau}_n\}_{n=0}^{\infty}$ and $\{\tilde{l}_n^{j,k}, 1 \leq j,k \leq K\}_{n=0}^{\infty}$.

It is immediate from the evolution equations (2.14)-(2.17) that $\tilde{w}_n^k$, $\tilde{r}_n^k$ and $\tilde{r}_n$ are integrable for all $n \geq 0$ and $1 \leq j,k \leq K$. The following lemma establishes the basis of our discussion in this section.

Lemma 10

For all $n \geq 0$ and $1 \leq j,k \leq K$,

$$\tilde{w}_n^k \leq E[w_n^k|\mathcal{H}] \quad a.s. \tag{5.1}$$

$$\tilde{r}_n^k \leq E[r_n^k|\mathcal{H}] \quad a.s. \tag{5.2}$$

$$\tilde{r}_n \leq E[r_n|\mathcal{H}] \quad a.s. \tag{5.3}$$

Proof

As inequalities (5.2), (5.3) are direct consequences of (5.1), we only show inequality (5.1). The proof is by induction on n.

Consider the case $n = 0$. For all $1 \leq k \leq K$, it is clear that $\tilde{w}_0^k = 0 = w_0^k = E[w_0^k|\mathcal{H}]$.

Assume that (5.1) holds for some $n+1$. Then applying Jensen's theorem for conditional expectations to (2.15) yields

$$\begin{aligned} E[w_{n+1}^k|\mathcal{H}] &\geq \max(0, \max_{j \in A(\Pi_G(t_e^k))}(E[w_n^j|\mathcal{H}] + E[l_n^{j,k}|\mathcal{H}] - E[\tau_n|\mathcal{H}]) \\ &\geq \max(0, \max_{j \in A(\Pi_G(t_e^k))}(\tilde{w}_n^j + E[l_n^{j,k}|\mathcal{H}] - E[\tau_n|\mathcal{H}]) \\ &= \tilde{w}_{n+1}^k \end{aligned}$$

Thus by induction (5.1) holds for all $n \geq 0$. □

Remark
Lemma 10 remains true under the weaker assumptions

$$\begin{array}{rcc} \tilde{\tau}_n \geq E[\tau_n|\mathcal{H}] & n \geq 0 & a.s. \\ \tilde{l}_n^{j,k} \leq E[l_n^{j,k}|\mathcal{H}] & n \geq 0, 1 \leq j,k \leq K & a.s. \end{array}$$

Corollary 11

For all $n \geq 0$ and $1 \leq k \leq K$,

$$\tilde{w}_n^k \leq_{ci} w_n^k \tag{5.4}$$

$$\tilde{r}_n^k \leq_{ci} r_n^k \tag{5.5}$$

$$\tilde{r}_n \leq_{ci} r_n \tag{5.6}$$

The next corollary shows that the transient bounds (5.4)-(5.6) extend to steady state in the sense of Theorem 3 and Theorem 4.

Corollary 12

Assume that A_1 holds for both $\{\tau_n, l_n^{j,k}, 1 \le j,k \le K\}_{n=0}^{\infty}$ and $\{\tilde{\tau}_n, \tilde{l}_n^{j,k}, 1 \le j,k \le K\}_{n=0}^{\infty}$, and that for all $1 \le j,k \le K$, w_n^k and $\tilde{w}_n^k$ converge weakly to finite RV's w_∞^k and $\tilde{w}_\infty^k$ respectively. Then for all $1 \le k \le K$,

$$\tilde{w}_\infty^k \le_{ci} w_\infty^k \tag{5.7}$$
$$\tilde{r}_\infty^k \le_{ci} r_\infty^k \tag{5.8}$$
$$\tilde{r}_\infty \le_{ci} r_\infty \tag{5.9}$$

Proof

Let $M_n^k = w_n^k \circ \theta^{-n}$ and $\tilde{M}_n^k = \tilde{w}_n^k \circ \theta^{-n}$ (see Lemma 4, Lemma 5). The weak convergence of w_n^k and $\tilde{w}_n^k$ to finite RV's yields

$$\tilde{M}_n^k \le \tilde{M}_\infty^k < \infty \qquad a.e.$$
$$M_n^k \le M_\infty^k < \infty \qquad a.e.$$

Let $f : R^+ \to R^+$ be a convex increasing function. Assume $f(M_\infty^k)$ and $f(\tilde{M}_\infty^k)$ are integrable. Then it is easy to show that $f(w_n^k)$ and $f(\tilde{w}_n^k)$ are both integrable for all $n \ge 0$. Hence Corollary 11 entails that

$$E[f(\tilde{M}_n^k)] = E[f(\tilde{w}_n^k)] \le E[f(w_n^k)] = E[f(M_n^k)] \le E[f(M_\infty^k)]$$

Letting n goes to ∞ in the inequality

$$E[f(\tilde{M}_n^k)] \le E[f(M_\infty^k)]$$

yields

$$E[f(\tilde{M}_\infty^k)] \le E[f(M_\infty^k)]$$

which implies (5.7).

It is clear that under the above assumptions, the RV's $r_n^k, \tilde{r}_n^k, r_n, \tilde{r}_n$ converge weakly to the finite RV's $r_\infty^k, \tilde{r}_\infty^k, r_\infty, \tilde{r}_\infty$ respectively. (5.8) and (5.9) follow directly from (5.7). □

Let

$$\bar{\tau} = E[\tau], \qquad n \ge 0$$
$$\bar{l}^{j,k} = E[l_n^{j,k}], \qquad n \ge 0, 1 \le j,k \le K$$

Let $\bar{w}_n^k, \bar{r}_n^k, \bar{r}_n$ $(n \ge 0, 1 \le k \le K)$ be defined as follows

$$\bar{w}_0^k = 0 \tag{5.10}$$
$$\bar{w}_{n+1}^k = \max(0, \max_{j \in A(\Pi_G(t_e^k))} (\bar{w}_n^j + \bar{l}^{j,k}) - \bar{\tau}) \tag{5.11}$$
$$\bar{r}_n^k = \max_{j \in A(\Pi_G(t_e^k))} (\bar{w}_n^j + \bar{l}^{j,k}) \tag{5.12}$$
$$\bar{r}_n = \max_{1 \le k \le K} \bar{r}_n^k \tag{5.13}$$

and $\breve{w}_n^k, \breve{r}_n^k, \breve{r}_n$ $(n \ge 0, 1 \le k \le K)$ be defined as follows

$$\breve{w}_0^k = 0 \tag{5.14}$$
$$\breve{w}_{n+1}^k = \max(0, \max_{j \in A(\Pi_G(t_e^k))} (\breve{w}_n^j + \bar{l}^{j,k}) - \tau_n) \tag{5.15}$$
$$\breve{r}_n^k = \max_{j \in A(\Pi_G(t_e^k))} (\breve{w}_n^j + \bar{l}^{j,k}) \tag{5.16}$$
$$\breve{r}_n = \max_{1 \le k \le K} \breve{r}_n^k \tag{5.17}$$

Lemma 10 and Corollary 11 entails

Theorem 8

Assume the set of RV's $\{\tau_n\}_{n=0}^{\infty}$ *and the set of RV's* $\{\sigma_n^i, 1 \leq i \leq |V|\}_{n=0}^{\infty}$ *are mutually independent. Then for all* $n \geq 0$ *and* $1 \leq k \leq K$,

$$\bar{w}_n^k \leq_{ci} \breve{w}_n^k \leq_{ci} w_n^k \tag{5.18}$$
$$\bar{r}_n^k \leq_{ci} \breve{r}_n^k \leq_{ci} r_n^k \tag{5.19}$$
$$\bar{r}_n \leq_{ci} \breve{r}_n \leq_{ci} r_n \tag{5.20}$$

Proof

Observe that the difference between $\bar{w}_n^k$ and $\breve{w}_n^k$ originates from differences between interarrival times only. Applying Lemma 10 and Corollary 11 to the SQN generated by (5.14)-(5.17) with $\mathcal{H}$ equal to the trivial σ-algebra (with $\mathcal{H} = \sigma(\bar{l}^{j,k}, 1 \leq j, k \leq K, j \in A(\Pi_G(t_e^k)))$) yields immediately that for all $n \geq 0$ and $1 \leq k \leq K$,

$$\bar{w}_n^k \leq_{ci} \breve{w}_n^k$$
$$\bar{r}_n^k \leq_{ci} \breve{r}_n^k$$
$$\bar{r}_n \leq_{ci} \breve{r}_n$$

Similarly applying Lemma 10 and Corollary 11 to the SQN generated by (2.14)-(2.17) with $\mathcal{H} = \sigma(\{\tau_n\}_{n=0}^{\infty})$ yields:

$$\breve{w}_n^k \leq_{ci} w_n^k$$
$$\breve{r}_n^k \leq_{ci} r_n^k$$
$$\breve{r}_n \leq_{ci} r_n$$

The proof is thus completed. □

Corollary 13

Under the assumption of Theorem 8, if for all $1 \leq k \leq K$, *the RV's* w_n^k *converges weakly to a finite RV* w_∞^k, *then for all* $1 \leq k \leq K$,

$$\bar{w}_\infty^k \leq_{ci} \breve{w}_\infty^k \leq_{ci} w_\infty^k \tag{5.21}$$
$$\bar{r}_\infty^k \leq_{ci} \breve{r}_\infty^k \leq_{ci} r_\infty^k \tag{5.22}$$
$$\bar{r}_\infty \leq_{ci} \breve{r}_\infty \leq_{ci} r_\infty \tag{5.23}$$

Proof

The proof is similar to that of Corollary 12. □

Theorem 8 and Corollary 13 provide lower bounds for waiting and response times. It is obvious that the stability condition for these lower bounds is weaker than the initial one. Indeed, let $\bar{\gamma}$ be defined by (3.23) with $l^{j,k} o \theta^{-n}$ replaced by $\bar{l}^{j,k}$, it is easy to see that $\bar{\gamma} \leq \gamma$ using the same type of techniques as above.

The remainder of this section focuses on the computation of such bounds. As $\bar{r}_n^k$ and $\bar{r}_n$ can be obtained immediately from $\bar{w}_n^k$ $(1 \leq k \leq K)$, we will only discuss the method for computing $\bar{w}_n^k$.

Lemma 11

Assume A_1 *holds and that* $\mathcal{G}$ *is strongly connected. Assume in addition that the RV's* $\bar{w}_n^k$ $(1 \leq k \leq K)$ *converge weakly to finite RV's* $\bar{w}_\infty^k$. *Then there exists a deterministic integer* N_0 *such that for all* $n \geq N_0$,

$$\bar{w}_n^k = \bar{M}_{N_0}^k \tag{5.24}$$

where

$$\bar{M}^k_{N_0} = \max(0, \max_{1\le m\le N_0}(\bar{H}^k_m - mE[\tau]))$$

and

$$\bar{H}^k_m = \max_{\{i_s|s=1,\cdots,m,i_{m+1}=k,i_s\in A(\Pi_G(t_s^{i_s+1}))\}}(\sum_{s=1}^{m}\bar{l}^{i_s,i_{s+1}})$$

Proof

From Theorem 4 and Theorem 5, the weak convergence of the RV's $\bar{w}^k_n$ to the finite RV's $\bar{w}^k_\infty$ is ensured by the condition

$$\begin{aligned}\nu &= \max_{1\le d\le K}\frac{1}{d}\max_{1\le i_1,\cdots,i_d\le K}(\bar{L}^{i_1,i_2}+\cdots+\bar{L}^{i_{k-1},i_k}+\bar{L}^{i_d,i_1})\\ &< KE[\tau]\end{aligned}$$

where $\bar{L}^{i_s,i_{s+1}}$ is defined in (3.21) with $l^{j,k}\circ\theta^{-n}$ replaced by $\bar{l}^{j,k}$.

Let $\bar{M}^k_n$ be defined by (3.11)-(3.12) with $l^{j,k}=\bar{l}^{j,k}$ and $\tau=\bar{\tau}$, i.e. $\bar{M}^k_n=\bar{w}^k_n\circ\theta^{-n}$ $(1\le k\le K, n\ge 0)$. From Lemma 4 we know that $\bar{M}^k_n\uparrow\bar{M}^k_\infty<\infty$. Using Lemma 6, $\bar{M}^k_n$ can be rewritten as

$$\bar{M}^k_n = \max(0, \max_{1\le m\le n}(\bar{H}^k_m - mE[\tau]))$$

Let

$$\chi^k_m = \bar{H}^k_m - mE[\tau]$$

Then

$$\bar{M}^k_n = \max(0, \max_{1\le m\le n}\chi^k_m)$$

Let N_0 be the integer satisfying the relation

$$N_0 = [\frac{\nu + K\cdot\max_{1\le u,v\le K}\bar{L}^{u,v}}{KE[\tau]-\nu}+1]\cdot K$$

where $[x]$ denotes the integer part of the real number x. For all $m\ge N_0$, let $m=m_1K+m_2$, where $0\le m_2<K$, we get

$$\begin{aligned}\bar{H}^k_m &\le \max_{1\le k\le K}\bar{H}^k_m\\ &< \max_{1\le k\le K}\bar{H}^k_{(m_1+1)K}\\ &= \max_{1\le v_0,\cdots,v_{m_1+1}\le K}\sum_{s=0}^{m_1}\bar{L}^{v_s,v_{s+1}}\end{aligned}$$

It follows from Appendix 1 (equation (8.3)) that

$$\max_{1\le v_0,\cdots,v_{m_1+1}\le K}\sum_{s=0}^{m_1}\bar{L}^{v_s,v_{s+1}}\le(m_1+1)\nu+K\cdot\max_{1\le u,v\le K}\bar{L}^{u,v}$$

Since $m\ge N_0$, we obtain

$$m_1\ge\frac{\nu+K\cdot\max_{1\le u,v\le K}\bar{L}^{u,v}}{KE[\tau]-\nu}$$

in other words

$$(m_1+1)\nu+K\cdot\max_{1\le u,v\le K}\bar{L}^{u,v}\le m_1KE[\tau]$$

Thus

$$\bar{H}^k_m<\max_{\vec{v}_{m_1}}H(\vec{v}_{m_1})\le m_1KE[\tau]\le mE[\tau]$$

which entails

$$\chi_m^k < 0, \qquad m \geq N_0$$

Hence for all $m \geq N_0$, and all $1 \leq k \leq K$,

$$\bar{M}_m^k = \bar{M}_{N_0}^k$$

As $\bar{w}_n^k =_{st} \bar{M}_n^k$ (where $=_{st}$ denotes equality in law), we get for all $n \geq N_0$,

$$\bar{w}_n^k =_{st} \bar{M}_n^k = \bar{M}_{N_0}^k$$

Due to the fact that the right most side of the above equations is a constant, the proof of the lemma is thus completed. □

Remarks:
1. In the above proof we have in fact shown that for all $m \geq N_0$

$$\chi_m = \max_{1 \leq k \leq K} \chi_m^k < 0$$

2. Intuitively, Lemma 11 indicates that if the processor graph $\mathcal{G}$ is strongly connected and if SQN is stable, then $\bar{w}_n^k$, $\bar{r}_n^k$ and $\bar{r}_n$ converge to constant values when $n \to \infty$. Furthermore these values can be reached within bounded time, where the bound on time is given by N_0.

The end of this section is devoted to various extensions of Lemma 11. The first extension is concerned with the case of non strongly connected $\mathcal{G}$.

When $\mathcal{G}$ is not strongly connected, let $\mathcal{G}_1, \cdots, \mathcal{G}_g$ be its maximal strongly connected subgraphs. Lemma 11 provides an algorithm for computing lower bounds for the waiting times in SQN-G_1, $\cdots$, SQN-G_g defined in Section 3. Let $\bar{w}_k^{k,i}$ ($k \in \mathcal{V}_i$) be the lower bound of $w_n^{k,i}$ obtained with the interarrival and service pattern τ and $\bar{l}^{j,k}$ respectively. According to Lemma 11, there exists an integer N_0^i ($1 \leq i \leq g$) such that for all $k \in \mathcal{V}_i$, $n \geq N_0^i$,

$$\bar{w}_n^{k,i} = \bar{M}_{N_0^i}^{k,i}$$

where $\bar{M}_n^{k,i}$ is defined by (3.13)-(3.14) with $l^{j,k} = \bar{l}^{j,k}$ and $\tau = \bar{\tau}$.

As indicated in Lemma 3, $\tilde{\mathcal{G}}$ is always acyclic, let the nodes of $\tilde{\mathcal{G}}$ $(1, \cdots, g_0, g_0+1, \cdots, g)$ be ordered in such a way that if (i,j) is an edge in $\tilde{\mathcal{G}}$ then $i < j$, and that nodes $1, \cdots, g_0$ have no predecessors in $\tilde{\mathcal{G}}$.

For $1 \leq i \leq g$, let

$$\nu_i = \max_{1 \leq d \leq |\mathcal{V}_i|} \frac{1}{d} \max_{v_1, \cdots, v_d \in \mathcal{V}_i} (\bar{L}^{v_1,v_2} + \cdots + \bar{L}^{v_{d-1},v_d} + \bar{L}^{v_d,v_1})$$

and

$$\xi = \max_{1 \leq u,v \leq K} \bar{L}^{u,v}$$

and for $1 \leq i \leq g$,

$$N_i' = [\frac{\nu_i + (|\mathcal{V}_i| + 1)\xi}{|\mathcal{V}_i| E[\tau] - \nu_i} + 1] \cdot |\mathcal{V}_i|$$

Note that ξ is computed for the whole set $\{1, \cdots, K\}$, and not for $\mathcal{V}_i$, so that $N_i' \geq N_0^i$.

Finally denote as N_i the quantity

$$N_i = \begin{cases} N_i', & 1 \leq i \leq g_0, \\ N_i' + \max_{i_0 \in \Pi_{\tilde{\mathcal{G}}}(i)} N_{i_0}, & g_0 + 1 \leq i \leq g \end{cases} \tag{5.25}$$

Theorem 9

Assume A_1 holds and that the stability condition for SQN-G is satisfied. Then for every i and k, $1 \leq i \leq g$, $k \in \bar{V}_i$, all $n \geq N_i$,

$$\bar{w}_n^k = \bar{M}_{N_i}^k \tag{5.26}$$

where

$$\bar{M}_{N_i}^k = \max(0, \max_{1 \leq m \leq N_i}(\chi_m^k))$$

$$\chi_m^k = \bar{H}_m^k - mE[\tau]$$

and

$$\bar{H}_m^k = \max_{\{i_s | s=1,\cdots,m, i_{m+1}=k, i_s \in A(\Pi_G(t_e^{i_s+1}))\}} (\sum_{s=1}^{m} \bar{l}^{i_s, i_{s+1}})$$

Proof

We are going to show that for all $k \in \mathcal{V}_i$ and all $m \geq 1$

$$\chi_m^k \leq \max_{1 \leq n \leq N_i'} \chi_n^k$$

This property, which entails (5.26), is proved by induction on i $(1 \leq i \leq g)$.

From Lemma 11, it is clear that for $1 \leq i \leq g_0$ the above assertion holds (In fact, from the proof of Lemma 11, we get the stronger assertion that $\chi_m^k < 0$ holds for all $k \in \mathcal{V}_i$, $1 \leq i \leq g_0$, and all $m \geq N_i = N_i' \geq N_0^i$).

Assume it holds for some i, $1 \leq i \leq g-1$. Then for every $k \in \mathcal{V}_{i+1}$, consider

$$\bar{H}_m^k = \max_{\{v_s | s=1,\cdots,m, v_{m+1}=k, v_s \in A(\Pi_G(t_e^{v_s+1}))\}} (\sum_{s=1}^{m} \bar{l}^{v_s, v_{s+1}})$$

Examin the series $v_1, \cdots, v_{m+1}$. Due to the fact that $\tilde{\mathcal{G}}$ is acyclic, and that all $\mathcal{G}_i$ $(i = 1, \cdots, g)$ are strongly connected, we can decompose this series in such a way that

$$\begin{aligned} v_1, \cdots, v_{h_1} &\in \mathcal{V}_{d_1} \\ v_{h_1+1}, \cdots, v_{h_2} &\in \mathcal{V}_{d_2} \\ &\vdots \\ v_{h_{q-1}+1}, \cdots, v_{h_q} &\in \mathcal{V}_{d_q} \end{aligned}$$

where

$$1 \leq h_1 < h_2 < \cdots < h_q = m+1$$

and

$$1 \leq d_1 < d_2 < \cdots < d_q = i+1$$

$$d_p \in P_{\tilde{\mathcal{G}}}(d_{p+1}), \qquad p = 1, \cdots, q-1$$

which entails

$$d_{q-1} < d_q = i+1$$

From the inductive assumption we get that

$$\chi_{h_{q-1}}^j \leq \max_{1 \leq n \leq N_{d_{q-1}}} \chi_n^j$$

holds for all $j \in \mathcal{V}_{d_{q-1}}$.

Let

$$\max_{1\le n\le N_{d_{q-1}}} \chi_n^j = \chi_{c_j}^j = \bar{l}^{x_0,x_1} + \cdots + \bar{l}^{x_{c_j-1},x_{c_j}} - c_j E[\tau]$$

where $c_j \le N'_{d_{q-1}}$, and $x_{c_j} = j \in \mathcal{V}_{d_{q-1}}$.

Let

$$\max_{1\le n\le N'_{i+1}} \chi_n^k = \chi_{b_k}^k = \bar{l}^{y_0,y_1} + \cdots + \bar{l}^{y_{b_k-1},y_{b_k}} - b_k E[\tau]$$

where $b_k \le N_{i+1}$, and $y_{b_k} = k \in \mathcal{V}_{i+1}$.

Observe that $v_{h_{q-1}+1}, \cdots, v_{h_q} \in \mathcal{V}_{i+1}$. Similarly to the proof of Lemma 11 we can show that

$$\begin{aligned}
\sum_{u=h_{q-1}}^{h_q} \bar{l}^{v_u,v_{u+1}} &= \bar{l}^{v_{h_{q-1}},v_{h_{q-1}+1}} + \sum_{u=h_{q-1}+1}^{h_q} \bar{l}^{v_u,v_{u+1}} \\
&\le \xi + ([\frac{h_q-h_{q-1}}{|\mathcal{V}_{i+1}|}+1])\nu_{i+1} + |\mathcal{V}_{i+1}| \sum_{u,v\in\mathcal{V}_{i+1}} \bar{l}^{u,v} \\
&\le ([\frac{h_q-h_{q-1}}{|\mathcal{V}_{i+1}|}+1])\nu_{i+1} + (|\mathcal{V}_{i+1}|+1)\xi
\end{aligned}$$

With the same manipulations as in the proof of Lemma 11, we get that for all $h_q - h_{q-1} \ge N'_{i+1}$,

$$\sum_{u=h_{q-1}}^{h_q} \bar{l}^{v_u,v_{u+1}} - (h_q - h_{q-1})E[\tau] < 0$$

Therefore

$$\sum_{u=h_{q-1}}^{h_q} \bar{l}^{v_u,v_{u+1}} - (h_q - h_{q-1})E[\tau] \le \chi_{b_k}^k$$

holds for all $h_q - h_{q-1} \ge 1$.

Hence for all $m \ge 1$

$$\begin{aligned}
&\chi_m^k \\
&= \bar{H}_m^k - mE[\tau] \\
&= \max_{\{v_s|s=1,\cdots,m,v_{m+1}=k,v_s\in A(\Pi_G(t_e^{v_s+1}))\}} (\sum_{s=1}^{m} \bar{l}^{v_s,v_{s+1}} - mE[\tau]) \\
&= \max_{\{v_s|s=1,\cdots,m,v_{m+1}=k,v_s\in A(\Pi_G(t_e^{v_s+1}))\}} \\
&\quad (\sum_{u=1}^{h_{q-1}-1} (\bar{l}^{v_s,v_{s+1}}) - h_{q-1}E[\tau]) + (\sum_{u=h_{q-1}}^{h_q-1} (\bar{l}^{v_s,v_{s+1}}) - (h_q - h_{q-1})E[\tau]) \\
&\le \max_{x_0,\cdots,x_{c_j}=y_0,y_1,\cdots,y_{b_k}=k} \\
&\quad (\bar{l}^{x_0,x_1} + \cdots + \bar{l}^{x_{c_j-1},x_{c_j}} + \bar{l}^{y_0,y_1} + \cdots + \bar{l}^{y_{b_k-1},y_{b_k}} - (c_j + b_k)E[\tau] \\
&\le \max_{1\le n\le N'_{i+1}} \chi_n^k
\end{aligned}$$

The assertion is thus proved to be true for $i+1$. Hence, the assertion holds for all $1 \le i \le g$. □

Remarks

1. The above theorem shows that if the SQN is stable, $\bar{w}_n^k$, $\bar{r}_n^k$ and $\bar{r}_n$ converge to constant values. Furthermore these values can be computed within bounded time.

2. Lemma 11 and Theorem 9 generate a very simple algorithm for computing these bounds which consists in calculating the bounds by (5.26) within the time defined by (5.25).

The results of Lemma 11 and Theorem 9 can be also be extended to the systems $\breve{w}_n^k$, $\breve{r}_n^k$, and $\breve{r}_n$ described by equations (5.14)-(5.17) in case the "noise" on interarrival times is bounded.

Corollary 14

Assume A_1 holds, that $\mathcal{G}$ is strongly connected, and that SQN-G is stable. Let $\tau_n = E[\tau] + \alpha_n$, $n \geq 0$. We assume in addition that there exists an integer n_0 and a real $D > 0$ such that for all $m \geq n_0$

$$|\sum_{n=1}^{m} \alpha_n| \leq D$$

Then there exists an integer N_0 such that for all $n \geq N_0$,

$$\breve{w}_n^k =_{st} \breve{M}_{N_0}^k \tag{5.27}$$

where

$$\breve{M}_{N_0}^k = \max(0, \max_{1 \leq m \leq N_0} (\bar{H}_m^k - \sum_{n=1}^{m} \tau_n))$$

and

$$\bar{H}_m^k = \max_{\{i_s | s=1,\cdots,m, i_{m+1}=k, i_s \in A(\Pi_G(t_s^{i_s+1}))\}} (\sum_{s=1}^{m} \bar{l}^{i_s, i_s+1})$$

Proof

The idea of the proof is similar to that of Lemma 11, except that we take

$$N_0 \geq \max(n_0, [\frac{\nu + K \cdot \max_{1 \leq u,v \leq K} \bar{L}^{u,v}}{KE[\tau] - \nu} + 1] \cdot K)$$

Then for all $m \geq N_0$, let $m = m_1 K + m_2$, where $0 \leq m_2 < K$, we get

$$\begin{aligned} H_m^k &\leq \max_{1 \leq k \leq K} H_m^k \\ &< \max_{1 \leq k \leq K} H_{(m_1+1)K}^k \\ &= \max_{1 \leq v_0, \cdots, v_{m_1+1} \leq K} \sum_{s=0}^{m_1} \bar{L}^{v_s, v_s+1} \\ &\leq (m_1 + 1)\nu + K \cdot \max_{1 \leq u,v \leq K} \bar{L}^{u,v} \end{aligned}$$

Let $\breve{M}_n^k$ be defined by (3.11)-(3.12) with $\bar{l}_n^{j,k}$ and τ_n. Using Lemma 6, we get

$$\breve{M}_n^k = \max(0, \max_{1 \leq m \leq n} (\bar{H}_m^k - \sum_{n=1}^{m} \tau_n))$$

Let

$$\chi_m^k = \bar{H}_m^k - \sum_{n=1}^{m} \tau_n$$

Then

$$\breve{M}_n^k = \max(0, \max_{1 \leq m \leq n} \chi_m^k)$$

Since $m \geq N_0$, we obtain

$$(m_1 + 1)\nu + K \cdot \max_{1 \leq u,v \leq K} \bar{L}^{u,v} \leq m_1 K E[\tau] - D$$

Thus

$$H_m^k <\leq m_1 K E[\tau] - D < mE[\tau] - D \leq \sum_{n=1}^{m} \tau_n$$

which entails ,

$$\chi_m^k < 0, \qquad m \geq N_0$$

Hence for all $m \geq N_0$, and all $1 \leq k \leq K$,

$$\check{M}_m^k = \check{M}_{N_0}^k$$

The last equation yields (5.27). □

Remark

The strong connectedness assumption on $\mathcal{G}$ can be easily removed from the above corollary by using the idea in the proof of Theorem 9.

6 Upper Bounds for Waiting and Response Times Based on Association

The main concern of this section is to provide for a simplified version of the basic integral equation (4.6) the solution of which is also an upper bound to the solution of (4.6) in some stochastic sense. The discussion of these upper bounds is based on the notion of associated RV's.

Definition 2 [41]

Real valued RV's $a_1, \cdots, a_n$ are said to be associated if

$$cov[h(a_1, \cdots, a_n), g(a_1, \cdots, a_n)] \geq 0$$

for all pairs of increasing functions $h, g : R^n \rightarrow R$.

Lemma 12 [41]

The association of RV's entails the following properties:

1. *Any subset of associated RV's are associated.*
2. *Increasing functions of associated RV's are associated.*
3. *Independent RV's are associated.*
4. *If two sets of associated RV's are independent of one another, then their union forms a set of associated RV's.*
5. *If $a_1, \cdots, a_n$ are associated RV's, then*

$$P[\max_{1 \leq i \leq n} a_i \leq t] \geq \prod_{i=1}^{n} P[a_i \leq t]$$

for all pairs of increasing functions $h, g : R^n \rightarrow R$.

Definition 3 [40]

Let F and H be the two distribution function on R. F is said to stochastically dominate H, $F \geq_{st} H$, iff

$$F(x) \leq H(x), \qquad \forall x \in R.$$

If a and b are two real valued RV's, we say that $a \geq_{st} b$, whenever

$$P[a \leq x] \leq P[b \leq x], \qquad \forall x \in R.$$

A direct consequence of the above definition and property 5 of Lemma 12 is

Lemma 13

Let $(a_1, \cdots, a_n)$ be a set of associated real valued RV's with respective distribution function $F_1, \cdots, F_n$. Let F be the distribution function of $\max(a_1, \cdots, a_n)$. Then

$$F \leq_{st} \prod_{i=1}^{n} F_i.$$

To terminate the introduction of the basic definition on *associated RV*, we state the following obvious lemma.

Lemma 14

Let $(F_1, \cdots, F_n)$ and $(H_1, \cdots, H_n)$ be two families of distribution functions on R. If for all $1 \leq i \leq n$, $F_i \geq_{st} H_i$, then

$$F_1 \cdot F_2 \cdots F_n = \prod_{i=1}^{n} F_i \geq_{st} \prod_{i=1}^{n} H_i = H_1 \cdot H_2 \cdots H_n$$

and

$$F_1 * F_2 * \cdots * F_n \geq_{st} H_1 * .H_2 * \cdots * H_n$$

where $\cdot$ and $*$ denote the product and the convolution of distribution functions respectively.

We are now in position to derive the upper bounds. If not redefined, the notations are defined in the previous sections. Throughout this section we will assume

A_3 : $\{\tau_n, \sigma_n^i, 1 \leq i \leq |V|\}_{n=0}^{\infty}$ *is a set of mutually independent RV's.*

By the definition of the RV's $l_n^{j,k}$ $(1 \leq j, k \leq K, n \geq 0)$, we can easily prove the following facts:

Lemma 15

Assume A_3 holds. Then

1. *for all $n \geq 0$, $\{l_n^{j,k}, 1 \leq j, k \leq K\}$ is a set of associated RV's,*
2. *for $n = 0, 1, \cdots$, the sets of associated RV's $\{l_n^{j,k}, 1 \leq j, k \leq K\}$ are mutually independent.*
3. *for every n, i, j, k, $(n \geq 0, 1 \leq i, j, k \leq K)$, the three RV's w_n^i, $l_n^{j,k}$ and τ_n are mutually independent.*

Lemma 16

Assume A_3 holds. For all $m \geq 0$,

$$\begin{aligned} S_m &= \{w_n^k, 1 \leq k \leq K, 0 \leq n \leq m\} \bigcup \{r_n^k, 1 \leq k \leq K, 0 \leq n \leq m\} \\ &\quad \bigcup \{l_n^{j,k}, 1 \leq j, k \leq K, n \geq 0\} \bigcup \{-\tau_n, n \geq 0\} \end{aligned}$$

is a set of associated RV's.

Proof

The proof is done by induction on m. The case when $m = 0$ is trivial observing the fact that $w_0^k = 0$ so that, from the previous lemma, $S_0 - \{r_0^k, 1 \le k \le K\}$ is a set of independent RV's. For all $1 \le k \le K$

$$r_0^k = \max_{j \in A(\Pi_G(t_e^k))} (l_0^{j,k})$$

is an increasing function of associated RV's. Properties 2 and 3 of Lemma 12 yield the desired conclusion for $m = 0$.

Assume S_m is a set of associated RV's for some $m \ge 0$. Then for every k, $1 \le k \le K$,

$$w_{m+1}^k = \max(0, \max_{j \in A(\Pi_G(t_e^k))} (w_m^j + l_m^{j,k}) - \tau_m)$$

is an increasing function of associated RV's. $S_{m+1} - \{r_0^k, 1 \le k \le K\}$ is therefore a set of associated RV's. And in turn

$$r_{m+1}^k = \max_{j \in A(\Pi_G(t_e^k))} (w_{m+1}^j + l_{m+1}^{j,k})$$

is an increasing function of associated RV's. Hence S_{m+1} is a set of associated RV's.

By induction, the assertion holds for all $m \ge 0$. □

Remark:
Lemma 16 holds under the weaker assumptions :

1. $\{\tau_n\}_{n=0}^{\infty}$ is independent of $\{l_n^{j,k}, 1 \le j, k \le K\}_{n=0}^{\infty}$.
2. $\{\tau_n\}_{n=0}^{\infty}$ is a set of associated RV's.
3. $\{l_n^{j,k}, 1 \le j, k \le K\}_{n=0}^{\infty}$ is a set of associated RV's.

Let W_n^k, R_n^k, R_n, $\Lambda_n^{j,k}$, and T_n^- denote the distribution functions on R of the RV's w_n^k, r_n^k, r_n, $l_n^{j,k}$, and $-\tau_n$ respectively. Note that W_n^k, R_n^k, R_n, $\Lambda_n^{j,k}$ have their support on R^+ and T_n^- on R^-.

We define the sequences Let $\hat{W}_n^k$, $\hat{R}_n^k$ and $\hat{R}_n$ ($1 \le k \le K$, $n \ge 0$) of distribution functions on R by the following recursion:

$$\begin{aligned}
\hat{W}_0^k &= U && (6.1)\\
\hat{W}_{n+1}^k &= U \cdot \prod_{j \in A(\Pi_G(t_e^k))} (\hat{W}_n^j * \Lambda_n^{j,k} * T_n^-) && (6.2)\\
\hat{R}_n^k &= \prod_{j \in A(\Pi_G(t_e^k))} (\hat{W}_n^j * \Lambda_n^{j,k}) && (6.3)\\
\hat{R}_n &= \prod_{1 \le k \le K} \hat{R}_n^k && (6.4)
\end{aligned}$$

where U denotes the step function:

$$\begin{aligned}
U(t) &= 0, \quad t < 0\\
U(t) &= 1, \quad t \ge 0
\end{aligned}$$

It is easy to see that the distribution functions $\hat{W}_n^k$, $\hat{R}_n^k$ and $\hat{R}_n$ have their support on R^+.

Theorem 10

Assume A_3 holds. Then for all $n \ge 0$, $1 \le k \le K$, we have

$$\begin{aligned}
W_n^k &\le_{st} \hat{W}_n^k && (6.5)\\
R_n^k &\le_{st} \hat{R}_n^k && (6.6)\\
R_n &\le_{st} \hat{R}_n && (6.7)
\end{aligned}$$

Proof

Let $df(a)$ denote the distribution function of RV a. We show (6.5) by induction on n. For $n = 0$, and all $1 \le k \le K$,

$$\hat{W}_0^k = U = df(w_0^k) = W_0^k$$

Hence (6.5) is true for $n = 0$.

Now assume (6.5) holds for some $n \ge 0$. Then from Lemma 14,

$$\begin{aligned}\hat{W}_{n+1}^k &= U \cdot \prod_{j \in A(\Pi_G(t_e^k))} (\hat{W}_n^j * \Lambda_n^{j,k} * T_n^-) \\ &\ge_{st} U \cdot \prod_{j \in A(\Pi_G(t_e^k))} (W_n^j * \Lambda_n^{j,k} * T_n^-)\end{aligned}$$

Using Lemma 16 and property 3 of Lemma 15 as well as Lemma 19 yields

$$\begin{aligned}\hat{W}_{n+1}^k &\ge_{st} U \cdot \prod_{j \in A(\Pi_G(t_e^k))} (W_n^j * \Lambda_n^{j,k} * T_n^-) \\ &\ge_{st} df(\max(0, \max_{j \in A(\Pi_G(t_e^k))} (w_n^j + l_n^{j,k}) - \tau_n)) \\ &= W_{n+1}^k\end{aligned}$$

By induction we have thus shown that (6.5) holds for all $n \ge 0$ and all $1 \le k \le K$.

With similar arguments, for all $n \ge 0$ and all $1 \le k \le K$,

$$\begin{aligned}\hat{R}_n^k &= \prod_{j \in A(\Pi_G(t_e^k))} (\hat{W}_n^j * \Lambda_n^{j,k}) \\ &\ge_{st} \prod_{j \in A(\Pi_G(t_e^k))} (W_n^j * \Lambda_n^{j,k}) \\ &\ge_{st} df(\max_{j \in A(\Pi_G(t_e^k))} (w_n^j + l_n^{j,k})) \\ &= R_n^k\end{aligned}$$

And finally

$$\begin{aligned}\hat{R}_n &= \prod_{1 \le k \le K} \hat{R}_n^k \\ &\ge_{st} \prod_{1 \le k \le K} R_n^k \\ &\ge_{st} df(\max_{1 \le k \le K} r_n^k) \\ &= R_n\end{aligned}$$

This completes the proof. □

The next result extends the transient bounds of Theorem 10 to steady state provided that $\hat{W}_n^k$ ($1 \le k \le K$) converge weakly to finite distribution functions $\hat{W}_\infty^k$ when n goes to ∞.

Corollary 15

Assume A_2 holds, and that for all k, $1 \le k \le K$, $\hat{W}_n^k$ converge weakly to finite distribution functions $\hat{W}_\infty^k$ when n goes to ∞. Then for all k, $1 \le k \le K$, the distribution functions $\hat{R}_n^k$, $\hat{R}_n$, W_n^k, R_n^k, R_n converge *weakly to finite distribution functions* $\hat{R}^k_\infty$, $\hat{R}_\infty$, W^k_∞, R^k_∞, R_∞ *when* n *goes to* ∞, *and*

$$W^k_\infty \leq_{st} \hat{W}^k_\infty \tag{6.8}$$
$$R^k_\infty \leq_{st} \hat{R}^k_\infty \tag{6.9}$$
$$R_\infty \leq_{st} \hat{R}_\infty \tag{6.10}$$

Proof

It was established that under the assumption A_2,

$$W^k_n \leq_{st} \hat{W}^k_n$$

holds for all $n \geq 0$, $1 \leq k \leq K$. Let M^k_n ($n \geq 0$, $1 \leq k \leq K$) be the RV's defined by equations (3.11)-(3.12). It follows from Lemma 5 that

$$W^k_n = df(M^k_n).$$

Hence the fact that $\hat{W}^k_n$ converges weakly to a finite distribution function $\hat{W}^k_\infty$ when n goes to ∞ entails that the increasing sequence RV's M^k_n cannot converge to ∞ almost surely. In other words the distribution functions W^k_n ($1 \leq k \leq K$) converge weakly to finite distribution functions W^k_∞.

The weak convergence of the distribution functions $\hat{R}^k_n$, $\hat{R}_n$, R^k_n, R_n towards finite distribution functions follows directly from the weak convergence of $\hat{W}^k_n$ and W^k_n, and from equations (6.3), (6.4), (2.15) and (2.16).

Equations (6.8)-(6.10) are immediate consequences of equations (6.5)-(6.7). □

The remainder of this section is concerned with computational algorithms for these upper bounds . It will be assumed that assumption A_2 holds. Observe that assumption A_2 implies that the sequence $\{l^{j,k}_n, 1 \leq j,k \leq K\}_{n=0}^\infty$ is i.i.d. Denote as $\Lambda^{j,k}$ and T^- the common distribution functions of the RV's $l^{j,k}_n$ and $-\tau_n$ respectively.

It follows from equations (6.1)-(6.4) that $\hat{W}^k_\infty$, $\hat{R}^k_\infty$ and $\hat{R}_\infty$ satisfy the following functional equations below

$$\hat{W}^k_\infty = U \cdot \prod_{j \in A(\Pi_G(t^k_s))} (\hat{W}^j_\infty * \Lambda^{j,k}_\infty * T^-_\infty) \tag{6.11}$$
$$\hat{R}^k_\infty = \prod_{j \in A(\Pi_G(t^k_s))} (\hat{W}^j_\infty * \Lambda^{j,k}_\infty) \tag{6.12}$$
$$\hat{R}_\infty = \prod_{1 \leq k \leq K} \hat{R}^k_\infty \tag{6.13}$$

The set of functional equations (6.11) can be solved as follows.

If $\hat{W}^k_n$ converges weakly to the limit distribution function $\hat{W}^k_\infty$, equations (6.1)-(6.5) entail that the following numerical schema converges towards the solution of (6.11).

$$\hat{W}^k_0 = U, \quad 1 \leq k \leq K,$$
$$\hat{W}^k_{n+1}(t) = U(t) \cdot \int_{-\infty}^t F(t-u)\,dT^-(u)$$

where U is unit function and

$$F(t) = \prod_{j \in A(\Pi_G(t^k_s))} \int_{-\infty}^t \hat{W}^j_n(t-u)\,d\Lambda^{j,k}(u)$$

Observe that this upper is based on K unknown functions of one real variable $W^1_\infty, \cdots, W^K_\infty$, to be compared with the initial equation (4.6) where the basic unknown function W_∞ is a function of K real variables.

7 Conclusions

In this paper, a new class of queueing models was introduced for evaluating the performance of multiprogrammed and multitasked multiprocessor systems under simple workload and scheduling assumptions.

In this model, task graphs are represented as general acyclic graphs, which allows the description of sequential or parallel execution, synchronization and spawning of tasks. Tasks execution times and programs interarrival times are represented as generally distributed stationary and ergodic sequences of random variables, which allows the description of asynchronism in program submission and uncertainty on the actual value of tasks execution times. The multiprocessor systems considered have generic architecture with a finite number of processors possibly sharing a central memory.

It was established that the evolution of the system can be characterized by a set of state variables that satisfy a system of stochastic recursive equations. These evolution equations capture two types of mechanisms that are characteristic of parallel processing: queueing mechanisms that are due the competition of all tasks for a limited number of processors and synchronization mechanisms that translate the precedence constraints between tasks.

The first result of the paper consists in a general expression for the stability condition of such systems under mild statistical assumptions that only require that the program inter-arrival times and execution times be stationary and ergodic random sequences. For this, the SQN was decomposed into a set of subnetworks that are determined by the structure of the Processor Graph. The stability condition of the SQN was shown to reduce to the intersection of the stability conditions of these subnetworks, which were derived explicitly. It is important to notice that this condition actually yields the maximum program throughput of the system or equivalently the maximum rate at which programs can be executed or submitted.

The second type of results concern the statistics of the stationary behavior of such systems. Basic integral equations were derived for the stationary joint distribution of the state variables. Important performance criteria such as stationary program response times or stationary queue sizes can be derived from the solution of this integral equation. An iterative numerical schema that converges to this solution was proposed. In addition to this, various upper and lower bounds were derived on the statistics of these quantities together with simple computational algorithms.

The practically important particular case where the service requirements of the tasks are deterministic was shown to be also of theoretical interest since the simple lower bounds that were derived are based on a deterministic version of the initial probabilistic problem. In this special case, the stationary program response times and queue sizes were shown to be equal to certain constants that can be obtained with a simple algorithm of known complexity.

To the best authors knowledge, the analysis presented in this paper is the first attempt towards an exact model within this context, and the results obtained here are new. Further research topics consist in studying analytical solutions to the basic integral equations, generalizing the model to other scheduling policies and relaxing the assumption that all task graphs are similar.

8 Acknowledgements

The authors are grateful to Dr. Alain Jean-Marie for his constructive comments and his various suggestions to improve the material contained in this paper.

9 Appendix 1

This appendix contains a proof of Theorem 5.

Observe first that under the assumption of the theorem, $L^{j,k}$ $(1 \le j, k \le K)$ is deterministic. Let

$$Q_n = \max_{1 \le v_1, \cdots, v_{n+1} \le K} \sum_{i=1}^{n} L^{v_i, v_{i+1}} \tag{9.1}$$

For all $1 \le d \le K$, and all $1 \le k \le K$,

$$\begin{aligned} Q_{nd} &= \max_{1 \le v_1, \cdots, v_{nd+1} \le K} \sum_{i=1}^{nd} L^{v_i, v_{i+1}} \\ &\ge n \cdot \max_{1 \le v_1, \cdots, v_{d-1} \le K} (L^{k,v_1} + L^{v_1,v_2} \cdots + L^{v_{d-2},v_{d-1}} + L^{v_{d-1},v_k}) \end{aligned}$$

which entails

$$\gamma \ge \nu \tag{9.2}$$

Now for every vector $\vec{v}_n = (v_1, \cdots, v_{n+1}) \in (1, K)^{n+1}$ given by (8.1), let $Q_n(\vec{v}_n)$ denote

$$Q_n(\vec{v}_n) = \sum_{h=0}^{n} L^{v_h, v_{h+1}}$$

Consider the series $v_0, v_1, \cdots, v_{n+1}$. Scanning its vertices from the left to the right, as soon as we find a vertex equal to one of the precedently scanned vertices, for example,

$$v_p = v_q \qquad p < q$$

we remove the vertices between v_p and v_q (including v_p but excluding v_q) from the series. These vertices will be said to form a cycle, and the length of the cycle is defined as the sum of the corresponding $L^{j,k}$'s. This procedure is iterated and successively found cycles are removed from the series until no cycle can be found. It can easily be shown that there are at most K vertices in each cycle, and there are at most K vertices left in the final series.

Now we group the cycles in such a way that cycles having the same number of vertices are put into a same set. Let $\mathcal{A}_h$ $(h = 1, \cdots, K)$ denote the set of cycles having h vertices, n_h the number of elements of $\mathcal{A}_h$, and $a_{h,j}$ the j-th element of $\mathcal{A}_h$. The length of $a_{h,j}$ is given by

$$\mathcal{L}_{h,j} = L^{i_1,i_2} + \cdots + L^{i_{h-1},i_h} + L^{i_h,i_1} \le h \cdot \nu$$

where $i_1, \cdots, i_h$ are the vertices in cycle $a_{h,j}$, $j = 1, \cdots, n_h$.

We can rewrite $Q_n(\vec{v}_n)$ as

$$Q_n(\vec{v}_n) = \sum_{h=1}^{K} \sum_{s=1}^{n_h} \mathcal{L}_{h,s} + \sum_{s=1}^{m_0} L^{u_s,v_s}$$

and

$$n = \sum_{h=1}^{K} h n_h + m_0$$

where m_0 $(0 \le m_0 \le K)$ is the number of vertices left in the final series.

As a consequence,

$$\begin{aligned} Q_n(\vec{v}_n) &= \sum_{h=1}^{K} \sum_{s=1}^{n_h} \mathcal{L}_{h,s} + \sum_{s=1}^{m_0} L^{u_s,v_s} \\ &\le \sum_{h=1}^{K} \sum_{s=1}^{n_h} h \cdot \nu + m_0 \cdot \max_{1 \le u,v \le K} L^{u,v} \\ &= (n - m_0)\nu + m_0 \cdot \max_{1 \le u,v \le K} L^{u,v} \\ &\le n\nu + K \cdot \max_{1 \le u,v \le K} L^{u,v} \end{aligned}$$

which yields

$$Q_n = \max_{\vec{v}_n} Q_n(\vec{v}_n) \leq n \cdot \nu + K \cdot \max_{1 \leq u,v \leq K} L^{u,v}$$

or

$$Q_n \leq n \cdot \nu + K \cdot \max_{1 \leq u,v \leq K} L^{u,v} \tag{9.3}$$

Dividing the above inequality by n at each side and letting $n \longrightarrow \infty$ yields

$$\gamma \leq \nu \tag{9.4}$$

The result of the theorem is therefore obtained from (8.2) and (8.4). □

References

[1] E.G.Coffman, P.J.Denning, *Operating Systems theory.* Prentice Hall, 1972.

[2] L.Lamport, *Time, Clocks and the Ordering of Events in a Distributed System.* Comm.ACM, Vol.21, No.7, pp.558-565, July 1978.

[3] C.A.R.Hoare, *Communicating Sequential Processes.* Comm. ACM, Vol.21, No.8, pp.666-677, Aug. 1978.

[4] I.C.Pyle, *The Ada Programming Language.* Prentice-Hall International, London, 1981.

[5] G.R.Andrews, F.B.Schneider, *Concepts and Notations for Concurrent Programming.* ACM Computing Surveys, Vol.15, No.1, pp.3-43, March 1983.

[6] J.P.Huang, *Modeling of Software Partition for Distributed Real-Time Applications.* IEEE Trans. on Software Engineering. Vol.SE-11, No.10, pp.1113-1126, 1985.

[7] V.Milutinović, J.A.B.Fortes, and L.H.Jamieson, *A Multimicroprocessor Architecture for Real-Time Computation of a Class of DFT Algorithms.* IEEE Trans. on ASSP, Vol.ASSP-34, No.5, pp.1301-1309, 1986.

[8] T.Duncan, W.H.Huen, *Software Structure of No.5 ESS–A Distributed Telephone Switching System.* IEEE Trans. on Communications, Vol.COM-30, No.6, pp.1379-1385, 1982.

[9] G.J.Battarel, H.F.Savary, *Interprocess Communication System of the MT35 Digital Exchange.* Computer Communication Review, Vol.13, No.2, pp.197-203, 1983.

[10] M.Cosnard, Y.Robert, *Algorithmique Parallèle: Une Etude de Complexité.* Techniques et Sciences Informatiques, Paris, 1987.

[11] M.A.Marsan, F.Gregoretti, *Memory Interference Models for a Multiprocessor System with a Shared Bus and a Single External Common Memory.* North-Holland, Microprocessing and Microprogramming 7, pp.124-133, 1981.

[12] M.A.Marson, G.Balbo, G.Conte, and F.Gregoretti, *Modeling Bus Contention and Memory Interference in a Multiprocessor System.* IEEE Trans. on Computers, Vol.C-32, No.1, pp.60-71, Jan.1983.

[13] M.A.Holliday, M.K.Vernon, *Exact Performance Estimates for Multiprocessor Memory and Bus Interference.* IEEE Trans. on Computers, Vol.C-36, No.1, pp.76-85, Jan.1987.

[14] S.Yalamanchili, J.K.Aggarwal, *A Characterization and Analysis of Parallel Processor Interconnection Networks.* IEEE Trans. on Computers, Vol.C-36, No.6, pp.680-691, June 1987.

[15] A.K.Jones, P.Schwarz, *Experience Using Multiprocessor Systems—A Status Report.* ACM Computing Surveys, Vol.12, No.2, pp.121-165, June 1980.

[16] R.P.Bianchini, J.P.Shen, *Interprocessor traffic Scheduling Algorithm for Multiple-Processor Networks.* IEEE Trans. on Computers, Vol.C-36, No.4, pp.396-409, June 1987.

[17] S.E.Elmaghraly, *On the Expected duration of PERT type networks.* Management Science, Vol.13, pp.469-481, 1967.

[18] B.Dodin, *Bounding the Project Completion Time Distribution in PERT Networks.* Oper. Res., Vol.33, No.4, pp.862-881, (July-August) 1985.

[19] L.P.Devroye, *Inequalities for the Completion Times of Stochastic PERT Networks.* Math. Oper. Res., Vol.4, pp.441-447, 1980.

[20] L.Flatto, S.Hahn, *Two Parallel Queues Created by Arrivals With Two Demands.* SIAM J. Appl. Math., Vol.44, pp.1041-1053, 1984.

[21] F.Baccelli, *Two Parallel Queues Created by Arrivals with Two Demands: The M/G/2 Symmetrical Case.* INRIA Report, No.426, July 1985.

[22] F.Baccelli, A.Makowski, *Single Computable Bounds for the Fork-Join Queue.* Proc. Conf. Inform. Sci. Systems, John Hopkins Univ., pp.436-441, March 1985

[23] R.Nelson, A.N.Tantawi, *Approximate Analysis of Fork/Join Synchronization in Parallel Queues.* IBM Report RC11481, Oct. 1985.

[24] D.Towsley, S.P.Yu, *Bounds for Two Server Fork-Join Queueing Systems.* Submitted to "Operations Research".

[25] F.Baccelli, A.Makowki, A.Shwartz, *Fork-Join Queue and Related Systems with Synchronization Constraints: Stochastic Ordering, Approximations and Computable Bounds.* Electrical Engineering Technical Report, Univ. of Maryland, College Park, Jan. 87. To Appear in the J. A. P..

[26] K.Sigman, *Regeneration in Queues with Regenerative Input.* Submitted to "Queueing Systems"

[27] F.Baccelli, W.A.Massey, D.Towsley, *Acyclic Fork-Join Queueing Networks.* Rapport INRIA No 688, June 87, Submitted for publication to the J.A.C.M..

[28] E.Gelenbe, Z.Liu, *Performance Analysis Approximations for Parallel Processing on Multiprocessor Systems.* Submitted to IFIP Working Conference on Parallel Processing. 1988.

[29] Z.Liu, *Evaluation des Performances d'Applications Paralléles sur des Systèmes Multiprocesseurs.* Univ. of Paris-Sud, Rapport de DEA, Sept. 1986.

[30] W.W.Chu, K.K.Leung, *Module Replication and Assignment for Real-Time Distributed Processing Systems.* Proc. of the IEEE, Vol.75, No.5, May 1987.

[31] E.Gelenbe, E.Montagne, R.Suros, and C.M.Woodside, *A Performance Model of Block Structured Parallel Programs.* ISEM Rapport de recherche No.46, 1986.

[32] E.Gelenbe, P.Nelson, T.Philips, and A.Tantawi, *The Asymptotic Processing Time for a Model of Parallel Processing.* ISEM Rapport de recherche No.48, 1986.

[33] J.N.Tsitsiklis, C.H.Papadimitriou, and P.Humblet, *The Performance of a Precedence-Based Queuing Discipline.* JACM, Vol.33, No.3, July 1986, pp.593-602.

[34] A.Stafylopatis, E.Gelenbe, *Delay Analysis of Resequencing Systems with Partial Ordering.* PERFORMANCE 87, North-Holland (eds. P.-J.Courtois and G.Latouche), Dec. 1987, pp.433-446.

[35] F.Baccelli, Z.Liu, *Scheduling Partially Ordered Tasks in Multiprocessor Systems–Some Optimal Heuristics and Their Performance Comparisons.* In preparation.

[36] W.K.Chen, *Applied Graph Theory.* North-Holland series in *Applied Mathematics and Mechanics*, H.A.Lauwerier and W.T.Koiter eds., 1971.

[37] N.Deo, *Graph Theory with Applications to Engineering and Computer Science.* Prentice-Hall series in *Automatic Computation*, G.Forsythe ed., 1974.

[38] R.M.Loynes, *The Stability of Queues with Non Independent Inter-arrival and Service Times.* Proc. cambridge Ph.Soc.58, pp.497-520, 1962.

[39] F. Baccelli, P. Brémaud, *Palm Probabilities and Stationary queues.* Lecture Notes in Statistics No 41, Springer Verlag, 1987.

[40] D.Stoyan, *Comparison Methods for Queues and Other Stochastic Models.* English translation (D.J. Daley editor), J.Wiley and Sons, New York, 1984.

[41] R.Barlow, F.Proschan, *Statistical Theory of Reliability and Life Testing.* Holt, Rinehart and Winston, 1975.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique