To appear in Stochastic Models 1998

ALLOCATION OF INTERDEPENDENT RESOURCES FOR MAXIMAL THROUGHPUT

Leandros Tassiulas
Electrical Engineering Dept. and
Institute for Systems Research
University of Maryland
College Park, MD 20742

Partha P. Bhattacharya
IBM Research Division
T.J. Watson Research Center
Yorktown Heights, NY 10598

Abstract

We consider a queueing network that models certain computer and communication systems with interdependent resources. The resources are modeled by a collection of servers that cannot be allocated to the queues independently but subject to certain constraints. The customers have random service times and the service completions of different customers cannot be synchronized. We obtain necessary and sufficient conditions for the stability, that is, finiteness of long run average delays. We also propose a non-preemptive server allocation policy that stabilizes the system whenever it is stabilizable and hence attains maximal throughput. The policy does not need the knowledge of the arrival statistics and has polynomial time complexity for some specific applications.

Keywords: Constrained Queues, stability, throughput

1 Introduction

There are several applications of computer and communication networks that involve resources that are interdependent in the sense that the allocation of one resource may affect that of the other. The resources can only be allocated subject to certain system specified constraints.

We consider the following queueing network as a general model of such systems. The network consists of N nodes and K *distinct* servers. Customers arrive at each one of the N queues according to some arrival process that is unspecified at this time. There are certain constraints on how the servers can be allocated to the nodes and these are specified by a collection $\mathcal{U}$ of feasible server activation schedules. A feasible server activation schedule $u \in \mathcal{U}$ is an $N \times K$ binary matrix and u_{ik}, its $(i,k)^{th}$ element, equals 1 (respectively 0) if server k can (resp. cannot) be allocated to a customer at node i. We assume that a customer needs only one server for service from a node. Therefore, if $u_{ik_1} = u_{ik_2} = 1$ for some schedule $u \in \mathcal{U}$, then under this schedule, one customer is served by each of the servers k_1 and k_2. Also, a server can serve only one customer at a time; so for every server k, there is at most one node i such that $u_{ik} = 1$. After completing service from server k, a customer at node i joins the queue at node j with probability $p(i,k,j)$; clearly $\sum_{j=1}^{N+1} p(i,k,j) = 1$. Set $p(i,k,i) = 0$ for every $i \leq N$, $k \leq K$. This system will be referred to as a Generalized Constrained Queueing System (GCQS) in this paper.

Consider the following example of a multi-processor computer system processing parallelizable jobs. Such a job consists of many independent components that can be processed in parallel. A parallelizable job with i independent components needs i servers concurrently during service; it can begin service only when i servers are available and it keeps these servers busy for the entire duration of its service. Somewhat more generally, a parallelizable job may be processed in phases; in each phase, depending on its parallelism profile, there may be a different number of independent components and hence different number of processors may be needed concurrently for execution. After completing a phase, a job is either completed and it leaves the system or enters another phase requiring further processing, according to a given probability distribution. Suppose the total number of processors is K. Jobs can be processed in parallel as long as each job has the required number of servers and the total number of required servers is less than K.

Scheduling the server assignment in the above system amounts to determining how many jobs from each class will be served at each time period. Jobs may complete service asynchronously, therefore whenever some of the servers become free the scheduler may reassign servers to jobs. If the number of servers that are free at some time instant is less than the number of servers required by the jobs waiting for service, then these servers cannot be allocated and they are forced to idle. For example, when there are only jobs that require $K-1$ servers in the system and $K \geq 3$, then 1 of the K servers necessarily idle.

The problem of scheduling to minimize performance measures like makespan or average delay in a multiserver queueing system is in general a difficult one. For example, the problem of minimizing makespan or delay for a system with a fixed number of customers initially, no arrivals and no feedback, 2 servers, 1 class of customers each requiring 1 server is well known to be NP-complete. The presence of constraints in server allocation only make these problems harder. Because of the forced idle periods mentioned earlier, a work conserving policy has no meaning in these systems and the problem of stabilizability, that is, designing scheduling policies to achieve finite delays, becomes challenging. A good scheduling policy has to incorporate the constraints in an appropriate manner to deliver finite delays.

In this paper we determine necessary and sufficient conditions for stability of systems like the one described above and propose a policy that stabilizes the systems whenever the necessary and sufficient conditions hold. Our setting is in fact considerably more general than the example described above and allows distinct servers, arbitrary constraints in the simultaneous service of different customers (not just a constraint on the total number of servers needed as in the parallel processing application), as well as server dependent routing of the customers through several nodes before exiting the system. This system was motivated originally by multihop radio communication networks where the service constraints arise as interference constraints in the simultaneous transmissions (see Tassiulas et.al. [12]). Then the similarity with models arising in parallel processing and wireless cellular networks was noticed and the general model we consider here has within its scope all these practical systems.

The queueing network studied in this paper consists of $N + 1$ nodes and K distinct servers. Customers arrive at each of the nodes $1, \cdots, N$ for service with $N + 1$ as their destination node. There are certain constraints on how the servers can be allocated to the nodes and we assume that these can be specified by a collection $\mathcal{U}$ of all possible feasible server activation schedules. A feasible server activation schedule $u \in \mathcal{U}$ is an $N \times K$ binary matrix and u_{ik}, its $(i, k)^{th}$ element, equals 1 (respectively 0) if server k can (resp. cannot) be allocated to a customer at node i. We assume that a customer needs only one server for service from a node. Therefore, if $u_{ik_1} = u_{ik_2} = 1$ for some schedule $u \in \mathcal{U}$, then under this schedule, one customer is served by each of the servers k_1 and k_2. Also, a server can serve only one customer at a time; so for every server $k \leq K$, there is at most one node $i \leq N$ such that $u_{ik} = 1$. After completing service from server k, a customer at node i joins the queue at node j with probability $p(i, k, j)$; clearly $\sum_{j=1}^{N+1} p(i, k, j) = 1$. Set $p(i, k, i) = 0$ for every $i \leq N$, $k \leq K$. This system will be referred to as a Generalized Constrained Queueing System (GCQS) in this paper.

It is easy to see how GCQS models the parallel processing application discussed earlier. Node i consists of all jobs that need i servers concurrently at their current phase. Since there can be at most K nodes, without any loss of generality we can assume $N = K$ by setting the exogenous arrival rates to zero for all $K - N$ nodes. While the system has K servers, we need $\tilde{K} := (K + \lfloor K/2 \rfloor + \lfloor K/3 \rfloor + \cdots + 1)$ servers in the GCQS model to represent the constraints. GCQS servers $1, \cdots, K$ are the servers needed by node 1 jobs and therefore can be allocated *only* to node 1. GCQS servers $K + 1, \cdots, K + \lfloor K/2 \rfloor$ represent a combination of two servers needed by node 2 jobs and can be allocated *only* to node 2. Similarly GCQS servers $K + \lfloor K/2 \rfloor + 1, \cdots, K + \lfloor K/2 \rfloor + \lfloor K/3 \rfloor$ represent a combination of three servers and can *only* be allocated to node 3 and so on. A schedule u is a binary $K \times \tilde{K}$ matrix and is feasible, that is, belongs to $\mathcal{U}$ if and only if

$$u_{1k} = 0 \quad \text{for } k \geq K + 1, \tag{1}$$

$$u_{ik} = 0 \quad \text{for } k \leq K + \lfloor K/2 \rfloor + \cdots + \lfloor (K/(i-1)) \rfloor \text{ and}$$
$$\text{for } k \geq K + \lfloor K/2 \rfloor + \cdots + \lfloor K/i \rfloor + 1, \quad i = 2, \cdots, K; \tag{2}$$

and the following holds:

$$\sum_{i=1}^{K} i \left(\sum_{k=1}^{\tilde{K}} u_{ik} \right) \leq K. \tag{3}$$

The final inequality ensures that at nΦmost K actual servers are allocated. The routing probability is independent of the servers.

The distinct identity of the servers and the dependence of the routing probabilities on the servers in GCQS is essentially needed to be able to model multi-hop communication networks, where servers associated with node i are the communication links connecting node i to the other nodes of the network. The links are distinct as they connect node i to different nodes. The routing probabilities in this case depend on the servers as the destination of a message at the next hop depends on the link that it is being transmitted along from a node. The applicability of GCQS to this and another practical situation is considered in section 5.

A feasible scheduling policy for GCQS should be such that the allocation of servers to the queues is always according to some schedule that belongs to the set of feasible schedules $\mathcal{U}$. For example, a feasible non-preemptive policy may schedule, at its decision times, new customers for service as long as they are not conflicting with those being served at that moment and the resulting activation schedule is a feasible one, that is, belongs to $\mathcal{U}$. Let Π denote the class of all possible feasible server allocation policies. Let $Q_i^{\pi}(t)$ denote the queue size at node i at time t under policy $\pi \in \Pi$. We say that *GCQS* is stable under a policy π if the long run average incurred delay under this policy is finite; that is,

$$\sup_{t \geq 0} \frac{1}{t} \int_0^t E \sum_{i=1}^{N} Q_i^{\pi}(s)\, ds \ < \ \infty. \tag{4}$$

We will provide a characterization of the space of arrival rates for which GCQS is stabilizable (see Theorem 1 in section 2) and identify a specific policy $\pi^* \in \Pi$ that stabilizes the system whenever it is stabilizable (see Theorem 2 in section 2). Therefore, policy π^* can sustain the maximal arrival rate, or equivalently, can deliver the maximal throughput.

The stabilizing policy proposed in this paper takes scheduling decisions at certain time instances, at which the next decision instant and the schedule to be followed until the next decision instant are determined. In determining the schedules, an optimization problem over the space of feasible schedules needs to be solved. For the parallel processing application mentioned above, we show in section 2 that the optimization problem reduces to a special integer knapsack problem for which an efficient algorithm of time complexity $O(K^2)$ (K is the number of servers) is available. We have therefore obtained a fairly practical stabilizing policy. There are no known stabilizing policies for this system in the literature that are of polynomial time complexity. Two other applications involving a multi-hop radio network and a cellular network are presented in section 5. For the first application, the time complexity of the stabilizing policy is also polynomial while it is exponential for the latter application. A practical scheduling policy for these applications has to be of low complexity and has to have a distributed implementation. For the multi-hop radio network, a distributed implementation of the policy suggested here may be attempted with a central control station that has the knowledge of all queue lengths and can coordinate feasible transmissions. For the cellular network application however, the policy is not directly applicable except in situations where the size of the problem is small. Nevertheless, the results provided here are of interest since they provide upper bounds to achievable throughput. In view of the complicated nature of queueing systems with constraints, it is often difficult to design good scheduling policies that incorporate the constraints in a meaningful way. The stabilizing policy provided here, even if not directly implementable, provides insights as to how a good policy might work. Finally, in order to achieve efficiency in server allocation, the policy proposed here may not serve customers in the order of their arrivals and therefore is also not directly applicable in situations where it is essential to maintain FCFS service discipline.

Several models of resource sharing systems with the jobs having conflicting resource requirements, have appeared in the literature. Mitra et.al. [7],[8] considered models of locking based concurrency controlled transaction processing systems that are very similar to the one considered here with the exception that transactions in his model cannot be queued and an arriving transaction that finds some the needed items blocked is either discarded or restarted after a random amount of time. There is no queueing or waiting involved. Several interesting quantities like the probability of blocking and the mean number of concurrently active transactions were computed. A similar model was studied by Kelly [5]. Tsitsiklis et.al. [14] considered a model of a queueing system with infinite servers where an arriving transaction has to wait, with a certain probability, for a transaction that is being presently served or queued, to complete. Necessary and sufficient stability conditions were obtained.

The same model under more general statistical and blocking assumptions was considered by Baccelli and Liu [1]. Courcoubetis et.al. [4] studied a model of K servers operating in parallel, each with its own queue. Two types of customers arrive in the system: the first type of customers do not have any resource conflicts while the customers of the second type have to be processed simultaneously by all K servers. Each server serves its own queue in a $FCFS$ manner. Necessary and sufficient conditions for stability were derived. Bambos et.al. [2],[3] studied the stability of a model of parallel processing systems in which an incoming job has a requirement on the number of servers that it needs concurrently for service. Necessary and sufficient conditions for stability were obtained and a stabilizing policy was also given. Besides considering a model with a more general topology, the scheduling policy proposed in this paper results in a simpler policy of lower complexity when specialized to the models considered in [2],[3].

Tassiulas and Ephremides [12] studied a model similar to the one considered here, motivated by radio networks. However, the service times were assumed to be deterministic and identical and the servers were synchronized to start and complete service at the same time. These assumptions are typically valid for slotted networks. Because of the synchronization, all the servers are idle at the decision instants and any schedule can be selected for activation. If the customers have random service times and service is non-preemptive, which is the case considered here, the service completions cannot be synchronized to occur at the same time, and a server may need to be activated while other servers are serving other queues. An inappropriate choice of the server activation schedule may leave some of the servers underutilized and consequently result in a throughput that is smaller than the maximum attainable. This issue was not addressed in [12].

The rest of the paper is organized as follows. The necessary and sufficient conditions for stability (Theorems 1 and 2) and the stabilizing policy for GCQS are presented in section 2. At the end of the section, we show how these results specialize to the parallel processing application discussed earlier and lead to a scheduling policy of polynomial complexity. Sections 3 and 4 contain the proofs of Theorems 1 and 2 respectively. In section 5, we briefly present two other applications that can be modeled by GCQS. Our conclusions and some directions for future research are outlined in section 6.

2 Main Results

In this section, we will first state the necessary and sufficient conditions for stability of GCQS and provide a stabilizing policy. Towards the end, we show how these results specialize to the parallel processing example mentioned in the introduction.

We will need to make the following assumptions.

(A) The collection $\mathcal{U}$ of feasible schedules has the following property: if $u \in \mathcal{U}$ and an $N \times K$ matrix $\tilde{u}$ is such that for every $i \leq N$, $k \leq K$,$\tilde{u}_{ik}$ is $\{0,1\}$ valued and $\tilde{u}_{ik} \leq u_{ik}$, then $\tilde{u} \in \mathcal{U}$ as well.

(B) For every node i, there is a path to the destination, that is, there is a sequence of node-server pairs $\{(i_j, k_j)\}_{j=1}^{n-1}$ with $i_1 = i$, $i_n = N+1$ and $n \leq N+1$ such that $\prod_{j=1}^{n-1} p(i_j, k_j, i_{j+1}) > 0$.

Assumption (A) states that every schedule obtained by idling some of the servers activated by a feasible schedule is also feasible. This is quite natural and can be seen to hold in practice. Assumption (B) states that there is a path of non-zero probability from every node to the destination node. This is quite natural as well since a node not satisfying this property would not be able to sustain any arrival rate at all. We will also assume the arrival processes and the service requirements are mutually independent.

Our first result concerns a necessary condition of stabilizability. We need some additional assumptions on the arrivals and the service requirements of customers at the nodes. Let $A_i(t)$ denote the number of exogenous arrivals to node i by time t. We assume the existence of an arrival rate, that is, the existence of $\lambda_i > 0$ such that $A_i(t)/t \to \lambda_i$ a.s. as $t \to \infty$. We also assume that the service times for every server k at node i are i.i.d with mean β_i; this could be relaxed to a weaker assumption that is similar to the one on arrival processes but this not done for simplicity. Let $\mathbb{R}^{N \times N}$ denote the set of $N \times N$ matrices.

For the network, a flow of customers between nodes is denoted as a $N \times (N+1)$ matrix whose $(i,j)^{th}$ component is the flow from node i to node j. A flow $\phi \in \mathbb{R}^{N \times (N+1)}$ is feasible if flows are conserved at every node; denote the set of feasible flows as

$$\mathcal{F} \;:=\; \left\{ \phi \in I\!R^{N\times(N+1)} \;:\; \lambda_i + \sum_{j=1}^{N} \phi_{ji} \;=\; \sum_{j=1}^{N+1} \phi_{ij},\; i = 1,\cdots,N \right\}. \tag{5}$$

Let the collection of all possible feasible server activation schedules be $\mathcal{U} := \{u^1, \cdots, u^L\}$, where for $l \leq L$, $u^l \in I\!R^{N\times K}$ is the l^{th} feasible schedule. For every $l \leq L$, let $\mathcal{S}^l$ denote a flow that is achieved by employing schedule u^l:

$$\mathcal{S}^l_{ij} \;:=\; \frac{1}{\beta_i} \sum_{k=1}^{K} u^l_{ik} p(i,k,j), \quad i \leq N, \;\; j \leq N+1.$$

The set of flows achieved by mixing all policies in $\mathcal{U}$ is therefore

$$\mathcal{S} \;:=\; \text{convex hull } \left\{\mathcal{S}^1, \cdots, \mathcal{S}^L\right\}. \tag{6}$$

We now state the necessary condition for a stabilizing policy. From the definitions of $\mathcal{F}$ and $\mathcal{S}$, it is immediate that if GCQS is stabilizable by some policy, then the flows achieved by the stabilizing policy must belong to both $\mathcal{F}$ and $\mathcal{S}$. This is our first result.

Theorem 1 *If there exists a stabilizing server allocation policy then $\mathcal{F} \cap \mathcal{S}$ is non-empty.*

It is plausible that if there is a feasible flow (that is, in $\mathcal{F}$) that belongs to the interior of $\mathcal{S}$, then there is a policy that stabilizes GCQS. We now present such a policy. The policy is non-preemptive and determines (possibly) new allocations at certain chosen time instants $\tau_0, \tau_1, ..$, at which all the servers are idle. The reallocation is done based on the system state and in some sense, 'priority' is given to those nodes that have a large number of customers waiting for service. In the specification of the policy, we need the function $\delta : \mathcal{U} \times I\!R^N \mapsto I\!R$ defined as

$$\delta(u,q) \;:=\; \sum_{i=1}^{N} \sum_{k=1}^{K} u_{ik} \Big[q_i - \sum_{j=1}^{N} p(i,k,j) q_j \beta_j / \beta_i \Big]. \tag{7}$$

Suppose that the system starts from from an arbitrary initial state at time $\tau_0 = 0$ at which all the servers are idle. Let $Q(t) := (Q_1(t), \cdots, Q_N(t))^T$ denote the vector of queue sizes at time t under this policy. At time τ_n, $n \geq 0$, compute

$$u^* \;:=\; \arg\max_{u \in \mathcal{U}} \delta\left(u, Q(\tau_n)\right) \tag{8}$$

and follow the server activation schedule u^* for a period of time equal to $[\delta(u^*, Q(\tau_n))]^r$ for some $r < 1$. From this time onwards, servers that become idle are not allocated to the jobs until time $\tilde{\tau}_{n+1}$ at which all the servers become idle. If at least one queue at time $\tilde{\tau}_{n+1}$ is non-empty, then set $\tau_{n+1} := \tilde{\tau}_{n+1}$; else set τ_{n+1} to be the time of the first arrival after $\tilde{\tau}_{n+1}$. At time τ_{n+1}, an appropriate server activation schedule is selected by (8) and the process continues. Let us denote this policy by π^*. Note that the policy does not use the information about the arrival rates at the nodes.

To show that the above policy stabilizes the system, we need more restrictive assumptions on the statistics of the arrival and service processes. Specifically, we will assume that at node i the arrivals occur according to a Poisson stream of rate λ_i and that the service requirements have finite third moment. Let S^o denote the interior of a set S.

Theorem 2 *If $\mathcal{F} \cap \mathcal{S}^o$ is non-empty then π^* stabilizes GCQS.*

The proofs of Theorems 1 and 2 are presented in the next two sections. Note that we have left out the critically balanced case in which $\mathcal{F}$ has an intersection with the boundary of $\mathcal{S}$ but not with its interior, that is, $\mathcal{F} \cap \mathcal{S}^o = \emptyset$ but $\mathcal{F} \cap \mathcal{S} \neq \emptyset$. In this case, the feasible flows can be achieved only by a convex combination of the feasible schedules. The system may be stabilizable in this case and our results do not show whether the policy π^* is stabilizes the system. The analysis of this case requires more assumptions on the arrival and service distributions as evident from the work of Loynes [6] on single server queues. We conjecture that with appropriate additional assumptions, GCQS is not stabilizable in this critically balanced case. If this case does not occur then Theorems 1 and 2 together imply that policy π^* achieves the maximum possible throughput: To see this, let $\Lambda^\pi \subset \mathbb{R}^N$ denote the set of exogenous arrival rates for which GCQS operating under policy $\pi \in \Pi$ is stable. From Theorem 1, we have

$$\bigcup_{\pi \in \Pi} \Lambda^\pi \subset \left\{\lambda \in \mathbb{R}^N : \mathcal{F} \cap \mathcal{S} \neq \emptyset\right\} \tag{9}$$

while Theorem 2 implies

$$\left\{\lambda \in \mathbb{R}^N : \mathcal{F} \cap \mathcal{S}^o \neq \emptyset\right\} \subset \Lambda^{\pi^*}. \tag{10}$$

Combining (9) and (10), we have in this case

$$\Lambda^{\pi^*} = \bigcup_{\pi \in \Pi} \Lambda^{\pi}.$$

We discuss briefly the delay properties of the stabilizing policy π^*. In this paper, we have only shown that the delays are finite. However, the fact that π^* stabilizes the system whenever it is stabilizable (except the critically balanced case described earlier) suggests that π^* has favorable delay properties as well. A property of π^* worth noting is that it tends to avoid buildup of large queues by giving preference to nodes with "larger" queue sizes: this is particularly easy to see in models without feedback where (7) reduces to $\delta(u, q) = \sum_{i=1}^{N} q_i \left(\sum_{k=1}^{K} u_{ik}\right)$. Consequently, a customer at a node will receive service as its queue builds up. There may be some periods in between decision instants during which some of the servers idle if the nodes that they were assigned to at the beginning of the decision instant, run out of customers. Since the schedule is not changed under π^* until the next decision instant, these servers continue to idle up to that time unless new customers arrive, even if there are customers waiting for service at other nodes they could have been assigned to. This may serve as a potential weakness of π^*. Towards this end, we make the following observations. Because of the non-preemptive nature of the system, the services of the customers can not be synchronized to complete simultaneously; so it is not clear in the first place whether an assignment of idle servers to nodes with customers in between decision instants of π^* leads to a more efficient utilization of the resources. More importantly, the duration between decision instants under π^* can be seen to be of the order of the total number of customers at the beginning of the decision instant plus a residual service time and the length of this duration can be controlled by choosing the parameter r in the definition of the policy. Recall that our policy remains stabilizing as long as $0 < r < 1$; so by choosing a small value of r, the time between decision instants can be reduced if necessary thus lessening the possibility of the situation described above. Note that there are no known stabilizing policies for GCQS, let alone policies that minimize the delay. We believe that the stabilizing policy provided here is a starting point for the discovery of policies with good delay properties.

In the rest of this section, we consider the parallel processing application mentioned in the introduction and show how the results described in this section translate to this important special case. Recall the GCQS model for this application which is described in the introduction. Briefly, the GCQS model consists of $K+1$ nodes and $\tilde{K}$ servers; customers at node i, $i \leq K$, need i servers concurrently and after service, is routed to node j with probability $p(i,j)$ (here $p(i,i) = 0$ for all i). Node $K+1$ is the destination node. Recall also the description of the set $\mathcal{U}$ of feasible schedules as given by (1), (2) and (3). For this example, the situation simplifies if we define the flow through a node (rather than the flow from one node to another). With P denoting the $K \times K$ routing probability matrix with elements $p(i,j)$ and $\lambda := (\lambda_1, \cdots, \lambda_K)^T \in I\!R^K$ denoting the vector of exogenous arrival rates to the nodes, define the vector of flows through the nodes: $\tilde{\lambda} := \left(\tilde{\lambda}_1, \cdots, \tilde{\lambda}_K\right)^T$ as a solution to the equations

$$\lambda_i \; + \; \sum_{j=1}^{N} \tilde{\lambda}_j p(j,i) \; = \; \sum_{j=1}^{N+1} \tilde{\lambda}_i p(i,j), \quad i = 1, \cdots, N. \tag{11}$$

The solution is given by $\tilde{\lambda} := \left(I - P^T\right)^{-1} \lambda$. It is easy to check from the definition of $\mathcal{F}$ that the matrix $\phi \in I\!R^{K \times (K+1)}$ with elements $\phi_{ij} = \tilde{\lambda}_i p(i,j)$ is the unique member of $\mathcal{F}$. The set $\mathcal{S}^l$ of flows achieved by a schedule $u^l \in \mathcal{U}$ is the $K \times (K+1)$ matrix whose elements are given by

$$\mathcal{S}^l_{ij} \; = \; \frac{1}{\beta_i} p(i,j) \left(\sum_{k=1}^{\tilde{K}} u^l_{ik} \right), \quad i \leq K, \;\; j \leq K+1. \tag{12}$$

The set $\mathcal{S}$ is the convex hull of the matrices $\mathcal{S}^l$ as above. It follows that the necessary stabilizability condition $\mathcal{F} \cap \mathcal{S} \neq \emptyset$ is then equivalent to

$$\begin{pmatrix} \tilde{\lambda}_1 \\ \tilde{\lambda}_2 \\ \vdots \\ \tilde{\lambda}_K \end{pmatrix} \in \text{convex hull} \left\{ \begin{pmatrix} \frac{1}{\beta_1} \sum_{k=1}^{\tilde{K}} u_{1k} \\ \frac{1}{\beta_2} \sum_{k=1}^{\tilde{K}} u_{2k} \\ \vdots \\ \frac{1}{\beta_K} \sum_{k=1}^{\tilde{K}} u_{Kk} \end{pmatrix} \; : \; u \in \mathcal{U} \right\}; \tag{13}$$

for sufficiency the vector $\tilde{\lambda}$ needs to belong to the interior of the set on right-hand-side of (13). The condition (13) admits a simple interpretation. Observe that $\sum_{k=1}^{\tilde{K}} u_{ik}$ is the number of node i customers receiving service under schedule u so that $\frac{1}{\beta_i}(\sum_{k=1}^{\tilde{K}} u_{ik})$ is the departure rate from node i under a schedule u. Also $\tilde{\lambda}_i$ is the flow of customers through node i. Therefore (13) simply states that for stability, the vector of flows through the nodes should belong to the convex hull of the set of departure rates under feasible server activation schedules.

We now turn to the stabilizing policy π^*. From (3), it easy to see that in this example a schedule can be represented more concisely by the vector $(n_1, n_2, \cdots, n_K)$ where $n_i = \sum_{k=1}^{\tilde{K}} u_{ik}$ is the number of node i customers receiving service. The set of feasible schedules consists of all schedules $(n_1, n_2, \cdots, n_K)$ that satisfy the constraint

$$\sum_{i=1}^{K} i n_i \leq K; \quad n_i \text{ nonnegative integer}, \quad i = 1, \cdots, K. \tag{14}$$

Define

$$\tilde{Q}_i(t) := Q_i(t) - \sum_{j=1}^{K} p(i,j) Q_j(t) \beta_j / \beta_i, \quad i = 1, \cdots, K.$$

During $[\tau_n, \tau_{n+1}]$, policy π^* uses the schedule $(n_1, \cdots, n_K)$ that solves

$$\max \left\{ \sum_{i=1}^{K} \tilde{Q}_i(\tau_n) n_i \ : \ \sum_{i=1}^{K} i n_i \leq K, \quad n_i \text{ nonnegative integer } \forall\, i \right\}, \tag{15}$$

which corresponds to solving the optimization problem defined by (8) for this problem. Fortunately, (15) is a favorable instance of the well-known integer knapsack problem. Although NP-complete in general, there exists pseudo-polynomial algorithms and for our special case, in fact an optimal solution can be easily constructed in $O(K^2)$ time following problem 2-22 in p.55 of Parker and Rardin [11]. We have therefore obtained a simple implementable stabilizing policy for the general model of a parallel processing computer system. We note that this is an improvement over policies existing in the literature (see Bambos et al [2],[3]) for this type of problems.

3 Proof of Theorem 1

We will show that if the system is stabilizable, that is, if there exists a policy π such that

$$\sup_{t \geq 0} \frac{1}{t} \int_0^t E \sum_{i=1}^{N} Q_i^{\pi}(s)\, ds < \infty \tag{16}$$

then $\mathcal{F} \cap \mathcal{S} \neq \emptyset$. The idea is straightforward. Let $T_{ij}^{\pi}(t)$ denote the number of transitions from node i to node j (that is, the number of departures from node i that joined node j) by time t. The limit of $T_{ij}^{\pi}(t)/t$ is shown to be a flow in $\mathcal{S}$. Via the 'customer conservation' equation (see (22) below), we then show that this flow belongs to $\mathcal{F}$ as well. The proof provided here is a more direct one than that in [12] where some results of max-flow problem in combinatorial optimization were used.

We first study the process $T_{ij}^{\pi}(t)/t$. Some notation is needed. Recall from section 2 that $D_{ik}^{\pi}(t)$ denotes the number of departures from node i via server k by time t under policy π. Let $t^{l,\pi}$ denote the total amount of time up to t during which π followed schedule u^l. Without any loss of generality, we can assume that the times $t^{l,\pi}$ are defined so that all the activated servers during $t^{l,\pi}$ (that is, all k such that $u_{ik}^l = 1$ for some i) are serving customers and never idle. If a server k remains idle, say during the time interval $[t, t']$ while schedule u^l is followed, then assume that in fact policy $u^{l'}$ is being followed during $[t, t']$, where $u^{l'}$ is defined as $u_{im}^{l'} = u_{im}^l$ for $m \neq k$ and $u_{ik}^{l'} = 0$ for all i. That $u^{l'}$ is feasible follows from assumption (A) in the previous section. The advantage of this definition is that from the assumption on service time distributions in the previous section, we have that for every l, i, k such that $u_{ik}^l = 1$:

$$\left(\frac{\beta_i D_{ik}^{\pi}(t)}{t^{l,\pi}}\right) \to 1 \quad \text{on} \quad \left\{t^{l,\pi} \to \infty\right\}. \tag{17}$$

Finally, define $\hat{p}^t(i,k,j)$ to be the following estimate of the routing probabilities:

$$\hat{p}^t(i,k,j) \;:=\; \frac{1}{D_{ik}^{\pi}(t)} \sum_{n=1}^{D_{ik}^{\pi}(t)} 1\left\{\text{customer } n \text{ joined node } j \text{ after service by server } k \text{ at node } i\right\}. \tag{18}$$

We can write the total number of transitions from node i to node j as

$$T_{ij}^{\pi}(t) \;=\; \sum_{l=1}^{L}\sum_{k=1}^{K} u_{ik}^l D_{ik}^{\pi}(t)\hat{p}^{t^{l,\pi}}(i,k,j). \tag{19}$$

Dividing through (19) by t, we obtain

$$T_{ij}^{\pi}(t)/t \;=\; \sum_{l=1}^{L}\left(\frac{t^{l,\pi}}{t}\right)\left[\frac{1}{\beta_i}\sum_{k=1}^{K} u_{ik}^l \left(\frac{\beta_i D_{ik}^{\pi}(t)}{t^{l,\pi}}\right)\hat{p}^{t^{l,\pi}}(i,k,j)\right]. \tag{20}$$

Observe that $\sum_{l=1}^{L}(t^{l,\pi}/t) = 1$ (note: an idle period corresponds to following a schedule $u^l \in \mathcal{U}$ such that $u_{ik}^l = 0$ for all i, k). Use (17) and the fact that $\hat{p}^t(i,k,j) \to p(i,k,j)$ as $t \to \infty$ in (20); a look at the definition of $\mathcal{S}$ in (6) then shows that $T_{ij}^{\pi}(t)/t$ becomes close to a member of $\mathcal{S}$ in the following sense: for all $\epsilon > 0$, there exists $\phi^{(t)} \in \mathcal{S}$ and t_0 such that for all i, j,

$$|T_{ij}^{\pi}/t - \phi_{ij}^{(t)}| \leq \epsilon, \quad t \geq t_0, \quad a.s. \tag{21}$$

We will now show that a specific limit of $\phi^{(t)}$ can be chosen to belong to $\mathcal{F}$ as well. Equating the customers arriving at and departing from a node, we write

$$Q_i^\pi(t) \;=\; Q_i^\pi(0) \;+\; A_i(t) \;-\; \sum_{j=1}^{N+1} T_{ij}^\pi(t) \;+\; \sum_{j=1}^{N} T_{ji}^\pi(t), \quad i = 1, \cdots, N. \tag{22}$$

We want to show that $\left|\lambda_i + \sum_{j=1}^{N} \phi_{ji}^{(t)} - \sum_{j=1}^{N+1} \phi_{ij}^{(t)}\right|$ becomes small as $t \to \infty$. Adding and subtracting $Q_i^\pi(t)/t$ to this quantity and using first (22) and then the triangle inequality, we have

$$\begin{aligned}\left|\lambda_i + \sum_{j=1}^{N} \phi_{ji}^{(t)} - \sum_{j=1}^{N+1} \phi_{ij}^{(t)}\right| \;&\le\; Q_i^\pi(t)/t + Q_i^\pi(0)/t + |\lambda_i - A_i(t)/t| \\ &\quad + \sum_{j=1}^{N} \left|T_{ji}^\pi(t)/t - \phi_{ji}^{(t)}\right| + \sum_{j=1}^{N} \left|T_{ij}^\pi(t)/t - \phi_{ij}^{(t)}\right| \end{aligned} \tag{23}$$

Let t become large and consider the terms on the right hand side in (23). Condition (16) implies that the first two terms converge to 0 a.s.. The third term converges a.s. to 0 because of the assumption on arrival time distributions in the previous section. The final two terms are small because of (21). Therefore, we have shown that for all $\epsilon > 0$, there exists $\phi^{(t)} \in \mathcal{S}$ and t_0 such that for all i:

$$\left|\lambda_i + \sum_{j=1}^{N} \phi_{ji}^{(t)} - \sum_{j=1}^{N+1} \phi_{ij}^{(t)}\right| \;\le\; \epsilon, \quad t \ge t_0. \tag{24}$$

It is now easy to construct $\phi \in \mathcal{S}$ that is in $\mathcal{F}$ as well. Choosing $\epsilon = 1/n$, construct a sequence $\{\phi^{(n)}\} \in \mathcal{S}$ such that for all i:

$$\left|\lambda_i + \sum_{j=1}^{N} \phi_{ji}^{(n)} - \sum_{j=1}^{N+1} \phi_{ij}^{(n)}\right| \;\le\; 1/n, \quad n = 1, 2, \cdots. \tag{25}$$

Since $\{\phi^{(n)}\} \in \mathcal{S}$ is compact, there exists a converging subsequence along which the limit is say $\phi^* \in \mathcal{S}$. Taking limit along this subsequence in (25), we obtain that $\phi^* \in \mathcal{F} \cap \mathcal{S}$.

4 Proof of Theorem 2

In this section, we show that if $\mathcal{F} \cap \mathcal{S}^o$ is non-empty, then the policy π^* stabilizes the system. The key is to study the Markov Chain $\{Q(\tau_n)\}_{n=0}^{\infty}$. We first establish a strong version of

Foster's criterion (see Lemma 1) which states that for some suitably chosen Lyapunov function, the drift of the Markov chain $\{Q(\tau_n)\}_{n=0}^{\infty}$ can be made arbitrarily negative uniformly outside a finite subset of $\mathbb{R}^N$. This enables us to show (see Lemma 4) that the time between successive visits of the queue size vector to that finite subset and that the total response times of customers served during this period has finite expectation. This is easily seen to imply (4). In this section we will make the dependence of the various quantities (e.g. the queue size vector) on the policy π^* implicit for notational convenience.

Let B denote a diagonal matrix with elements $B_{ii} := \beta_i$, $i = 1, \cdots, N$ and define

$$\sigma(q) \ := \ [\delta(u^*, q)]^r \tag{26}$$

where $u^* \in \mathbb{R}^{N\times K}$ with elements u^*_{ik} is computed as in (8) with $Q(T_n) = q$. The following result establishes the version of Foster's criterion needed for our purposes. Let $V : \mathbb{R}^N \mapsto [0, \infty)$ be the Lyapunov function defined by

$$V(q) \ = \ q^T B q \ = \ \sum_{i=1}^{N} \beta_i q_i^2. \tag{27}$$

Lemma 1 *There exists $\epsilon > 0$ and $C < \infty$ such that for every q satisfying $V(q) > C$, the following relations hold for the choices $f(q) := \sigma^2(q)$ or $\sigma(q)\sum_{i=1}^N q_i$:*

$$E\left[V(Q(\tau_1)) \ - \ V(q) \mid Q(\tau_0) = q\right] \ \leq \ -\epsilon \ f(q). \tag{28}$$

A look at the statement of Lemma 1 shows that the result holds uniformly for queue lengths belonging to the set $\{q : V(q) > C\}$ for large C. Before proceeding to the proof of Lemma (1), we will state and prove the following preliminary result that relates the growth rates of $\sigma(q)$, $\sum_{i=1}^N q_i$ and $\left(\sum_{i=1}^N q_i\right)^r$: on the set $\{q : V(q) > C\}$ the growth rate of $\sigma(q)$ as $C \to \infty$ is uniformly slower than that of $\sum_{i=1}^N q_i$ but faster than that of $\left(\sum_{i=1}^N q_i\right)^r$ (recall $r < 1$). The following two statements state this fact precisely.

$$\lim_{C\to\infty} \sup_{q:V(q)>C} \frac{\sigma(q)}{\sum_{i=1}^N q_i} \ = \ 0 \tag{29}$$

$$\liminf_{C\to\infty} \inf_{q:V(q)>C} \frac{[\sigma(q)]^{1/r}}{\sum_{i=1}^N q_i} \ > \ 0 \tag{30}$$

The proof of (29) is straightforward. Observe from (7), (8) and (26) that $\sigma(q) \leq [K\sum_{i=1}^N q_i]^r$; since $r < 1$, (29) follows easily. It is considerably much harder to show (30). It follows from the result stated below (Lemma 2) and the following easily verified inequality:

$$V(q) \geq \left(\min_{1\leq i\leq N}\beta_i\right)\left(\sum_{i=1}^N q_i^2\right) \geq N^{-1}\left(\min_{1\leq i\leq N}\beta_i\right)\left(\sum_{i=1}^N q_i\right)^2.$$

Lemma 2 *There exists C sufficiently large such that for every q satisfying $V(q) > C$, $[\sigma(q)]^{1/r} = \delta(u^*, q) \geq C_1[V(q)]^{1/2}$ for some constant C_1 that is independent of q.*

Proof. We first introduce some notation.

$$\delta_{ik}(q) := q_i - \sum_{j=1}^N p(i,k,j)q_j\beta_j/\beta_i, \quad i \leq N,\ k \leq K, \tag{31}$$

$$\delta_{\max}(q) := \max\left\{\delta_{ik}(q) : u_{ik} > 0 \text{ for some } u \in \mathcal{U},\ i \leq N,\ k \leq K\right\}. \tag{32}$$

If i^*, k^* achieves the maximum in the definition of $\delta_{\max}(q)$, then by choosing $u \in \mathcal{U}$ such that $u_{i^*k^*} = 1$ and $u_{ik} = 0$ for $i \neq i^*$, $k \neq k^*$ (this is certainly possible because of assumption (A)), we obtain

$$\delta(u^*, q) = \max_{u\in\mathcal{U}} \sum_{i=1}^N\sum_{k=1}^K u_{ik}\delta_{ik}(q) \geq \delta_{\max}(q), \tag{33}$$

so that it suffices to show the existence of a constant C so that

$$\inf_{q:V(q)>C} \frac{\delta_{\max}(q)}{[V(q)]^{1/2}} > 0.$$

Assume otherwise. Then for every $M = 1, 2, \cdots$, define the sets

$$G^M := \left\{q \in \mathbb{R}^N : [V(q)]^{1/2} \geq M\delta_{\max}(q)\right\} \neq \emptyset.$$

Note that $G^1 \supset G^2 \supset G^3 \cdots$. In the rest of the proof we will assume that $q \in G^1$ and therefore $q \in G^M$ for all M. Note from the definition of G^1 that

$$[V(q)]^{1/2} \geq \delta_{\max}(q), \quad q \in G^1. \tag{34}$$

Let $i_1, i_2, \cdots, i_{N+1}$ denote a permutation of $\{1, 2, \cdots, N+1\}$ so that $\beta_{i_1}q_{i_1} \geq \beta_{i_2}q_{i_2} \geq \cdots \geq \beta_{i_{N+1}}q_{i_{N+1}} = 0$. We can assume that node i_{N+1} in the permutation is always the destination node $N+1$.

Define $S(q) \subset \{1, 2, \cdots, N+1\}$ as follows. Let $c := (2N)^{-1}(\sum_{i=1}^{N} 1/\beta_i)^{-1/2}$ denote a specific constant and

$$j^* := \min\left\{ j \geq 1 : \beta_{i_j} q_{i_j} - \beta_{i_{j+1}} q_{i_{j+1}} \; > \; c\, \delta_{\max}(q) \left[[V(q)]^{1/2} / \delta_{\max}(q) \right]^{\frac{j+N-2}{2(N-1)}} \right\}., \tag{35}$$

Define

$$S(q) \; := \; \left\{1, \cdots, j^*\right\}.$$

Note that j^* depends on q but this dependence is suppressed in the notation. The proof of the Lemma is by contradiction consists of two steps.

Step 1. We show that for $j^* \leq N$ or equivalently $i_{N+1} \notin S(q)$. Indeed, otherwise, from (34) and $\frac{j+N-2}{2(N-1)} \leq 1$ for every $j \leq N$, we have from (35) that

$$\beta_{i_j} q_{i_j} - \beta_{i_{j+1}} q_{i_{j+1}} \; \leq \; c\, \delta_{\max}(q) \left[[V(q)]^{1/2} / \delta_{\max}(q) \right]^{\frac{j+N-2}{2(N-1)}} \; \leq \; c[V(q)]^{1/2}, \;\; j \leq N;$$

this yields (using the definition of c)

$$\beta_{i_1} q_{i_1} \; = \; \sum_{j=1}^{N} (\beta_{i_j} q_{i_j} - \beta_{i_{j+1}} q_{i_{j+1}}) \; \leq \; 2^{-1} (\sum_{i=1}^{N} 1/\beta_i)^{-1/2} [V(q)]^{1/2}. \tag{36}$$

However, this is a contradiction to the following fact:

$$V(q) \; = \; \sum_{i=1}^{N} \beta_i q_i^2 \; \leq \; \beta_{i_1}^2 q_{i_1}^2 (\sum_{i=1}^{N} 1/\beta_i). \tag{37}$$

Step 2. The contradiction. From assumption (B), it follows that there is a path starting from node $i_{j^*} \in S(q)$ to the destination node $i_{N+1} \notin S(q)$ (by step 1); that is, there is a sequence of nodes $(i_{j^*}, \cdots, i_m, i_n, \cdots, i_{N+1})$ such that nodes between i_{j^*} and i_m (including i_j and i_m) in the sequence belong to $S(q)$, node $i_n \notin S(q)$ and $p(i_m, k_m, i_n) > 0$ for some server k_m. From the definition of j^*, it follows that $m \leq j^* < n \leq N+1$. We obtain two inequalities needed for the proof. First, note that nodes $i_k \in S(q)$, $k \leq m-1$ and hence

$$\beta_{i_k} q_{i_k} - \beta_{i_{k+1}} q_{i_{k+1}} \; \leq \; \delta_{\max}(q) \left[[V(q)]^{1/2} / \delta_{\max}(q) \right]^{\frac{k+N-2}{2(N-1)}}. \tag{38}$$

Using first (38) and then (34), we have the first inequality : for $j \leq m-1$,

$$\beta_{i_j} q_{i_j} - \beta_{i_m} q_{i_m} \; = \; \sum_{k=j}^{m-1} (\beta_{i_k} q_{i_k} - \beta_{i_{k+1}} q_{i_{k+1}})$$

$$
\begin{aligned}
&= \sum_{k=j}^{m-1} \delta_{\max}(q) \left[[V(q)]^{1/2}/\delta_{\max}(q)\right]^{\frac{k+N-2}{2(N-1)}} \\
&\le cN\delta_{\max}(q) \left[[V(q)]^{1/2}/\delta_{\max}(q)\right]^{\frac{m+N-3}{2(N-1)}}. \qquad (39)
\end{aligned}
$$

Observe that $m \le j^* < j^* + 1 \le n$ implies that $\beta_{i_m} q_{i_m} \ge \beta_{i_{j^*}} q_{i_{j^*}} \ge \beta_{i_{j^*+1}} q_{i_{j^*+1}} \ge \beta_{i_n} q_{i_n}$. Using this fact together with the definition of j^* and (34), we obtain our second inequality:

$$
\begin{aligned}
\beta_{i_m} q_{i_m} - \beta_{i_n} q_{i_n} &\ge \beta_{i_{j^*}} q_{i_{j^*}} - \beta_{i_{j^*+1}} q_{i_{j^*+1}} \\
&> c\delta_{\max}(q) \left[[V(q)]^{1/2}/\delta_{\max}(q)\right]^{\frac{j^*+N-2}{2(N-1)}} \\
&\ge c\delta_{\max}(q) \left[[V(q)]^{1/2}/\delta_{\max}(q)\right]^{\frac{m+N-2}{2(N-1)}}. \qquad (40)
\end{aligned}
$$

The definition of $\delta_{\max}$ now yields

$$
\begin{aligned}
\delta_{\max}(q) &\ge q_{i_m} - \sum_{j=1}^{N} p(i_m, k_m, j) q_j \beta_j / \beta_{i_m} \\
&= (\beta_{i_m})^{-1} \sum_{j=1}^{N+1} p(i_m, k_m, i_j)(\beta_{i_m} q_{i_m} - \beta_{i_j} q_{i_j}) \\
&\ge (\beta_{i_m})^{-1} \Big[p(i_m, k_m, i_n)(\beta_{i_m} q_{i_m} - \beta_{i_n} q_{i_n}) \\
&\qquad - \sum_{j=1}^{m-1} p(i_m, k_m, i_j)(\beta_{i_j} q_{i_j} - \beta_{i_m} q_{i_m}) \Big], \qquad (41)
\end{aligned}
$$

where the final inequality follows from the fact that $\beta_{i_m} q_{i_m} \ge \beta_{i_j} q_{i_j}$ for $j \ge m$. Relation (41) holds for $q \in G^1$ and hence for all $q \in G^M$, $M = 1, 2, \cdots$. Apply (39) and (40) in (41) and consider the inequality for $q \in G^M$, M large. Since $G^M \ne \emptyset$ for all M, this is possible. The term $[V(q)]^{1/2}/\delta_{\max}(q) \to \infty$ as $M \to \infty$ and therefore, the term on rhs of (40) dominates that on rhs of (39). Also, since $\frac{m+N-2}{2(N-1)} \ge 1/2$ we have shown the following result: a positive constant C_2 can be chosen independently of q such that for M sufficiently large and for every $q \in G^M$:

$$
\delta_{\max}(q) \ge C_2 \delta_{\max}(q) \left([V(q)]^{1/2}/\delta_{\max}(q)\right)^{1/2} \qquad (42)
$$

or equivalently,

$$
[V(q)]^{1/2} \le (1/C_2)^2 \, \delta_{\max}(q); \qquad (43)
$$

this is clearly a contradiction in view of the definition of G^M. □

Before proceeding to the proof of Lemma 1, we note a fact from renewal theory. Consider a renewal process with inter-renewal r.v. X and let $R(t)$ be the remaining life (also known as excess) at time t.

Lemma 3 *For $k \geq 1$, if $EX^{k+1} < \infty$, then $\sup_{t\geq 0} ER^k(t) < \infty$.*

Proof. From the renewal equation describing $ER^k(t)$ (see Ross [10]), it is easy to see that $ER^k(t) \leq [1 + M(t)]EX^k$ where $M(t)$ is the expected number of renewals by time t. Also, $ER^k(t) \to EX^{k+1}/EX$ as $t \to \infty$. The claim follows. □

We will use Lemma 3 to claim that the residual service times have finite second moment (recall that it is assumed in this section that the service times have finite third moments).

Proof of Lemma 1. First note that because of (29), it suffices to prove (28) for $f(q) = \sigma(q)\sum_{i=1}^{N} q_i$. Write $\tau(q)$ for τ_1 and let $\bar{\tau}(q)$ denote the conditional expectation of τ_1 given $Q(\tau_0) = Q(0) = q$. Since the matrix B is symmetric, we can write, after some algebra:

$$E\Big[V(Q(\tau(q))) - V(q)|Q(0) = q\Big] = E\left[(Q(\tau(q)) - q)^T B\,(Q(\tau(q)) - q) \mid Q(0) = q\right] + 2E\left[(Q(\tau(q)) - q)^T Bq \mid Q(0) = q\right]. \quad (44)$$

We will bound each of the two terms on the right in (44). Consider the first term. Since the sum of the number of (exogenous) arrivals to node i and the total number of service completions at nodes $j \neq i$ during $\tau(q)$ is an upper bound for $Q_i(\tau) - q_i$, $i \leq N$, it follows that

$$E\left[(Q(\tau(q)) - q)^T B\,(Q(\tau(q)) - q) \mid Q(0) = q\right] \leq C_2\, E[\tau^2(q) + \tau(q) + 1]$$

for some positive constant C_2. The random variable $\tau(q)$ equals $\sigma(q)$ plus a residual service time that has finite second moment because of Lemma 3. Therefore (by choosing a larger C_2 if necessary), we have

$$E\left[(Q(\tau(q)) - q)^T B\,(Q(\tau(q)) - q) \mid Q(0) = q\right] \leq C_2\,[\sigma^2(q) + \sigma(q) + 1]. \quad (45)$$

We proceed to estimate the second term on the right in (44). This will provide the negative term that we need to show (28). Let $D_{ik}(q, T)$ denote the number of completions by server k at node i during time $[0,T]$ and let $\bar{D}_{ik}(q,T)$ denote the expectation. Recall that the schedule u^* is followed by policy π^* during $[\tau_0, \tau_1]$. Equating the number of arrivals to and departures from a node and taking expectations, we obtain

$$\begin{aligned} E\ [(Q_i(\tau) - q_i) \mid Q(0) = q] &= \lambda_i \bar{\tau}(q) - \sum_{k=1}^{K} u^*_{ik} \bar{D}_{ik}(q, \tau(q)) \\ &\quad + \sum_{j=1}^{N} \sum_{k=1}^{K} p(j,k,i) u^*_{jk} \bar{D}_{jk}(q, \tau(q)), \quad i \le N. \end{aligned} \tag{46}$$

Using first (46) and then the notation

$$d_{ik}(q, \tau(q)) := \frac{\beta_i \bar{D}_{ik}(q, \tau(q))}{\bar{\tau}(q)}, \quad i \le N, \quad k \le K, \tag{47}$$

we obtain after some algebra

$$\begin{aligned} E\ \left[(Q(\tau(q)) - q)^T Bq \mid Q(0) = q\right] &= \sum_{i=1}^{N} \beta_i q_i E\ [(Q_i(\tau(q)) - q_i) \mid Q(0) = q] \\ &= \bar{\tau}(q) \left[\sum_{i=1}^{N} \lambda_i \beta_i q_i - \sum_{i=1}^{N} \sum_{k=1}^{K} u^*_{ik} q_i d_{ik}(q, \tau(q)) \right. \\ &\quad \left. + \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} p(j,k,i) u^*_{jk} q_i d_{jk}(q, \tau(q)) \right] \end{aligned} \tag{48}$$

We need to represent the arrival rates $\{\lambda_i\}$ in terms of the elements of $\mathcal{S}$. From the definition of $\mathcal{S}$ in section 2, it follows that every $\phi \in \mathcal{S}^o$ can be written as

$$\phi_{ij} = \epsilon_1 \sum_{l=1}^{L} \frac{c_l}{\beta_i} \sum_{k=1}^{K} u^l_{ik} p(i,k,j) \tag{49}$$

for some $0 < \epsilon_1 < 1$ and non-negative $\{c_l\}$ satisfying $\sum_{l=1}^{L} c_l = 1$. Since $\mathcal{F} \cap \mathcal{S} \neq \emptyset$, we have that $\lambda_i = \sum_{j=1}^{N} \phi_{ij} - \sum_{j=1}^{N} \phi_{ji}$ for some $\phi \in \mathcal{F} \cap \mathcal{S}^o$ and using the representation of ϕ as given by (49) we derive

$$\lambda_i = \epsilon_1 \sum_{l=1}^{L} c_l \Big[\frac{1}{\beta_i} \sum_{k=1}^{K} u^l_{ik} - \sum_{j=1}^{N} \frac{1}{\beta_j} \sum_{k=1}^{K} u^l_{jk} p(j,k,i) \Big], \quad i \le N. \tag{50}$$

Multiplying both sides of (50) by $\beta_i q_i$ and summing, it follows from the definition (7) that

$$\begin{aligned} \sum_{i=1}^{N} \lambda_i \beta_i q_i &= \epsilon_1 \sum_{l=1}^{L} c_l \left[\sum_{i=1}^{N} \sum_{k=1}^{K} u^l_{ik} q_i - \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{K} p(j,k,i) u^l_{jk} q_i \beta_i / \beta_j \right] \\ &= \epsilon_1 \sum_{l=1}^{L} c_l \delta(u^l, q). \end{aligned} \tag{51}$$

Using first (51), (31) and then (8) and the fact that $\sum_{l=1}^{L} c_l = 1$ in the rhs of (48), we obtain

$$\begin{aligned} E\left[(Q(\tau(q)) - q)^T Bq \mid Q(0) = q\right] &= \bar{\tau}(q)\left[\epsilon_1 \sum_{l=1}^{L} c_l \delta(u^l, q) - \sum_{i=1}^{N}\sum_{k=1}^{K} d_{ik}(q, \tau(q)) u_{ik}^* \delta_{ik}(q)\right] \\ &\leq \bar{\tau}(q)\delta(u^*, q)\left[\epsilon_1 - \sum_{i=1}^{N}\sum_{k=1}^{K} d_{ik}(q, \tau(q)) \frac{u_{ik}^* \delta_{ik}(q)}{\delta(u^*, q)}\right]. \end{aligned} \tag{52}$$

If we could show that there exists a positive constant $C_3(\epsilon_1)$ such that

$$\inf_{q:V(q)>C_3(\epsilon_1)} \sum_{i=1}^{N}\sum_{k=1}^{K} d_{ik}(q, \tau(q)) \frac{u_{ik}^* \delta_{ik}(q)}{\delta(u^*, q)} \geq 1 - (1 - \epsilon_1)/2; \tag{53}$$

then we are done. To see this observe that we can deduce the following claims for q such that $V(q) > C$, C sufficiently large. First substituting (53) in (52), noting that $\tau(q) \geq \sigma(q)$ and using (26), we have

$$E\left[(Q(\tau(q)) - q)^T Bq \mid Q(0) = q\right] \leq -\bar{\tau}(q)\delta(u^*, q)(1 - \epsilon_1)/2 \leq -[\sigma(q)]^{1+1/r}(1 - \epsilon_1)/2. \tag{54}$$

Now use the estimates (45) and (54) in (44) to obtain

$$\begin{aligned} E\Big[V(Q(\tau(q))) - V(q) | Q(0) = q\Big] \leq &\left(\sigma(q)\sum_{i=1}^{N} q_i\right)\left\{C_2 \frac{\sigma(q)}{\sum_{i=1}^{N} q_i} + \frac{C_2}{\sum_{i=1}^{N} q_i}\right. \\ &\left. + \frac{C_2}{\sigma(q)\sum_{i=1}^{N} q_i} - (1 - \epsilon_1)\frac{[\sigma(q)]^{1/r}}{\sum_{i=1}^{N} q_i}\right\}; \end{aligned} \tag{55}$$

Consider each term inside $\{\cdots\}$ on the rhs in (55) for q such that $V(q) > C$, $C \to \infty$. The first term goes to 0 in view of (29). The second and third terms become small trivially. The third term remains negative uniformly in view of (30). The desired conclusion (28) now follows.

It remains to show that (53) holds. We first write using (47) that

$$\sum_{i=1}^{N}\sum_{k=1}^{K} d_{ik}(q, \tau(q)) \frac{u_{ik}^* \delta_{ik}(q)}{\delta(u^*, q)} = \frac{\sigma(q)}{\bar{\tau}(q)} \sum_{i=1}^{N}\sum_{k=1}^{K} \frac{\beta_i \bar{D}_{ik}(q, \tau(q))}{\sigma(q)} \frac{u_{ik}^* \delta_{ik}(q)}{\delta(u^*, q)}. \tag{56}$$

The idea will be to identify $S^* \subset \{1, \cdots, N\} \times \{1, \cdots, K\}$ such that (i) $q_i \to \infty$, $i \in S^*$ at a rate faster than $\sigma(q)$ and (ii) $\sum_{(i,k)\in S^*} u_{ik}^* \delta_{ik}(q) \approx \delta(u^*, q)$, as $C \to \infty$ for every q such that

$V(q) > C$. Since $\sigma(q)/\bar{\tau}(q) \to 1$ as $C \to \infty$ for q satisfying $V(q) > C$, (53) would follow then from the (elementary) Renewal Theorem. Towards this end, set $\epsilon_2 := (1-\epsilon_1)/6$ and

$$S^* \;:=\; \Big\{(i,k) : u^*_{ik}\delta_{ik}(q) \geq (1+\epsilon_2)^{-1}\delta(u^*,q),\; i \leq N,\; k \leq K\Big\} \neq \emptyset.$$

It is immediate from the definition of S^* that

$$\frac{\sum_{(i,k)\in S^*} u^*_{ik}\delta_{ik}(q)}{\delta(u^*,q)} \;\geq\; (1+\epsilon_2)^{-1}. \tag{57}$$

We now show that there exists $C_3(\epsilon_1)$ such that for every q satisfying $V(q) > C_3(\epsilon_1)$,

$$\frac{\bar{D}_{ik}(q,\tau(q))\beta_i}{\sigma(q)} \;\geq\; 1-\epsilon_2, \quad (i,k) \in S^*. \tag{58}$$

Set $\tilde{q}^i_j$ to be equal to q_j if $j \neq i$ and ∞ otherwise. Since (here $x \wedge y := \min(x,y)$)

$$\bar{D}_{ik}(q,\tau(q)) \;\geq\; \bar{D}_{ik}(q,\sigma(q)) \;\geq\; E\Big[D_{ik}(\tilde{q}^i,\sigma(q)) \wedge q_i\Big], \quad (i,k) : u^*_{ik} > 0,$$

we have, for every $2 \leq \theta < \infty$:

$$\frac{\bar{D}_{ik}(q,\tau(q))\beta_i}{\sigma(q)} \;\geq\; E\Big[\frac{D_{ik}(\tilde{q}^i,\sigma(q))\beta_i}{\sigma(q)} \wedge \frac{q_i\beta_i}{\sigma(q)} \wedge \theta\Big], \quad (i,k) : u^*_{ik} > 0. \tag{59}$$

Consider each term inside the expectation on the rhs in (59) for q such that $V(q) > C$, as $C \to \infty$. From (30), we have that $\sigma(q) \to \infty$ so that from the elementary renewal theorem it follows that the first term inside the expectation converges a.s. to 1 as $C \to \infty$. For the second term: use (31) (which implies that $\delta_{ik}(q) \leq q_i$), (26) and the definition of S^* to obtain that for $(i,k) \in S^*$,

$$q_i/\sigma(q) \;\geq\; \delta_{ik}(q)/\sigma(q) \;\geq\; u^*_{ik}\delta_{ik}(q)[\delta(u^*,q)]^{-r} \;\geq\; (1+\epsilon_2)^{-1}[\delta(u^*,q)]^{1-r}$$

and therefore the second term becomes unbounded as $C \to \infty$. Now (58) follows from (59) by letting $C \to \infty$ and using the bounded convergence theorem. Finally, as argued before, by choosing a larger $C_3(\epsilon_1)$ if necessary, we can ensure that for every q satisfying $V(q) > C_3(\epsilon_1)$,

$$[\sigma(q)/\bar{\tau}(q)] \;\geq\; 1 \;-\; \frac{\epsilon_2}{1-\epsilon_2}. \tag{60}$$

Using (57), (58) and (60) in (56) and restricting the sum to S^*, (53) follows easily. □

Lemma 1 has the following implication. With V and C as in Lemma 1, let F denote the finite set $\{q \in \mathbb{R}^N : V(q) \le C\}$ and let T_m^F denote the time at which the queue length process $\{Q(\tau_n)\}_{n=0}^{\infty}$ enters F for the m^{th} time; formally,

$$T_m^F \; := \; \inf\left\{t > T_{m-1}^F : Q(t) \in F,\ t \in \{\tau_n\}_{n=0}^{\infty}\right\}, \quad m = 1, 2 \cdots; \quad T_0^F := \tau_0.$$

Let $\mathcal{G}_n$ denote the sigma field generated by $(Q(\tau_k),\ k \le n)$, $n = 1, 2, \cdots$. Let P_q denote the probability measure governing the process $\{Q(t),\ t \ge 0\}$ starting from $Q(\tau_0) = q$ and E_q denote the expectation with respect to this measure. The following result shows that the total delay of customers between successive visits of the queue length vector $Q(t)$ to the finite set F has finite expectation. We will use the notation $1(A)$ to denote the indicator function of an event A.

Lemma 4 $E_q \int_{T_0^F}^{T_1^F} \sum_{i=1}^N Q_i(s)\, ds < \infty$ *for* $q \in F$.

Proof. Let

$$\nu^F \; := \; \inf\{n \ge 1 : Q(\tau_n) \in F\}$$

which is a $\mathcal{G}_n$-stopping time. Observe that letting $A[\tau_k, \tau_{k+1}]$ denote the total number of arrivals to the system during $[\tau_k, \tau_{k+1}]$, we can write

$$\int_{T_0^F}^{T_1^F} \sum_{i=1}^N Q_i(s)\, ds \; \le \; \sum_{k=0}^{\nu^F - 1} (\tau_{k+1} - \tau_k) \left[\sum_{i=1}^N Q_i(\tau_k) + A[\tau_k, \tau_{k+1}]\right]; \tag{61}$$

so that it suffices to show

$$E_q \sum_{k=0}^{\nu^F - 1} (\tau_{k+1} - \tau_k) \sum_{i=1}^N Q_i(\tau_k) \; < \; \infty, \quad q \in F; \tag{62}$$

and

$$E_q \sum_{k=0}^{\nu^F - 1} (\tau_{k+1} - \tau_k) A[\tau_k, \tau_{k+1}] \; < \; \infty, \quad q \in F. \tag{63}$$

We first show that with $f(q) = \sigma^2(q)$ or $\sigma(q) \sum_{i=1}^N q_i$:

$$E_q \sum_{k=0}^{\nu^F - 1} f(Q(\tau_k)) \; < \; \infty, \quad q \in F. \tag{64}$$

By using successively the facts that $\{\nu^F \geq n\} \in \mathcal{G}_{n-1}$, the process $\{Q(\tau_n)\}_{n=0}^{\infty}$ is Markov, Lemma 1 (note that on $\{\nu^F \geq n\}$, $Q(\tau_{n-1}) \notin F$) and $\{\nu^F \geq n\} \subset \{\nu^F \geq n-1\}$, we obtain

$$\begin{aligned} E_q\left[V(Q(\tau_n))1\{\nu^F \geq n\}\right] &= E_q\left[E\left[V(Q(\tau_n)) \mid \mathcal{G}_{n-1}\right]1\{\nu^F \geq n\}\right] \\ &= E_q\left[E\left[V(Q(\tau_n)) \mid Q(\tau_{n-1})\right]1\{\nu^F \geq n\}\right] \\ &\leq E_q\left[V(Q(\tau_{n-1}))1\{\nu^F \geq n\}\right] \\ &\qquad - \epsilon\, E_q\left[f(Q(\tau_{n-1}))1\{\nu^F \geq n\}\right] \\ &\leq E_q\left[V(Q(\tau_{n-1}))1\{\nu^F \geq n-1\}\right] \\ &\qquad - \epsilon\, E_q\left[f(Q(\tau_{n-1}))1\{\nu^F \geq n\}\right]. \end{aligned} \tag{65}$$

Upon iteration, (65) yields

$$0 \leq E_q\left[V(Q(\tau_n))1\{\nu^F \geq n\}\right] \leq E_q V(Q(\tau_1)) - \epsilon\, E_q\left[\sum_{k=1}^{n-1} f(Q(\tau_k))1\{k \leq \nu^F - 1\}\right]$$

and this implies that

$$E_q\left[\sum_{k=1}^{\infty} f(Q(\tau_k))\; 1\{k \leq \nu^F - 1\}\right] \leq \epsilon^{-1}E_q V(Q(\tau_1)) < \infty, \quad q \in F.$$

This proves (64). Observe from Lemma 2 that C can be chosen in Lemma 1 so that $\sigma(q) \geq 1$ whenever $q \notin F$; therefore (64) yields

$$E_q \nu^F < \infty \;\; \text{and} \;\; E_q \sum_{k=0}^{\nu^F-1} \sum_{i=1}^{N} Q_i(\tau_k) < \infty, \quad q \in F. \tag{66}$$

Let

$$S_k := \tau_{k+1} - \tau_k - \sigma(Q(\tau_k)) . \tag{67}$$

denote the maximum of the residual service times at $\tau_k + \sigma(Q(\tau_k))$. From Lemma 3, it follows that S_k has finite conditional second moment given $\mathcal{G}_k$; which coupled with (66) implies that

$$E_q \sum_{k=0}^{\nu^F-1} S_k^2 = \sum_{k=0}^{\infty} E_q E[S_k^2|\mathcal{G}_k]\; 1\{k \leq \nu^F - 1\} < \infty, \quad q \in F, \tag{68}$$

and

$$E_q \sum_{k=0}^{\nu^F-1} S_k \sum_{i=1}^{N} Q_i(\tau_k) = \sum_{k=0}^{\infty} E_q E[S_k|\mathcal{G}_k] \sum_{i=1}^{N} Q_i(\tau_k)\; 1\{k \leq \nu^F - 1\} < \infty, \quad q \in F. \tag{69}$$

We are now ready to conclude that (62) and (63) holds. First use (67) to write

$$E_q \sum_{k=0}^{\nu^F-1} (\tau_{k+1}-\tau_k)\sum_{i=1}^{N} Q_i(\tau_k) \;=\; E_q \sum_{k=0}^{\nu^F-1} \sigma(Q(\tau_k))\sum_{i=1}^{N} Q_i(\tau_k) + E_q \sum_{k=0}^{\nu^F-1} S_k \sum_{i=1}^{N} Q_i(\tau_k). \quad (70)$$

The finiteness of the two terms on rhs of (70) follows from (64) with $f(q) = \sigma(q)\sum_{i=1}^{N} q_i$ and (69) and we have shown (62). To prove (63), we use the measurability of $\{\nu^F \geq k+1\}$ with respect to $\mathcal{G}_k$, the Markov property and the memoryless property of arrivals to obtain (for some positive constant C_4)

$$\begin{aligned} E_q\left[\sum_{k=0}^{\nu^F-1} (\tau_{k+1}-\tau_k) A[\tau_k, \tau_{k+1}]\right] &= \sum_{k=0}^{\infty} E_q E\left[(\tau_{k+1}-\tau_k)A[\tau_k,\tau_{k+1}] \mid \mathcal{G}_k\right] 1\{k \leq \nu^F - 1\} \\ &= \sum_{k=0}^{\infty} E_q E\left[(\tau_{k+1}-\tau_k)A[\tau_k,\tau_{k+1}] \mid Q(\tau_k)\right] 1\{k \leq \nu^F - 1\} \\ &\leq C_4 E_q \sum_{k=0}^{\nu^F-1} [\tau_{k+1}-\tau_k]^2 . \end{aligned}$$

It remains to show that $E_q \sum_{k=0}^{\nu^F-1} [\tau_{k+1}-\tau_k]^2 < \infty$. Use (67) and the inequality $(a+b)^2 \leq 2(a^2+b^2)$ to write

$$\begin{aligned} E_q \sum_{k=0}^{\nu^F-1} [\tau_{k+1}-\tau_k]^2 &= E_q \sum_{k=0}^{\nu^F-1} [\sigma(Q(\tau_k)) \;+\; S_k]^2 \\ &\leq 2E_q \sum_{k=0}^{\nu^F-1} \sigma^2(Q(\tau_k)) \;+\; 2E_q \sum_{k=0}^{\nu^F-1} S_k^2. \end{aligned} \quad (71)$$

The finiteness of the two terms on the rhs of (71) follows from (64) with $f(q) = \sigma^2(q)$ and (68). □

The proof of Theorem 2 can now be concluded. From Lemma 4 and the finiteness of the set F, we obtain

$$C_5 \;:=\; \sup_{q\in F} E_q \int_{T_0^F}^{T_1^F} \sum_{i=1}^{N} Q_i(s)\, ds \;<\; \infty. \quad (72)$$

Set $\gamma(t) := \inf\{n \geq 1 : T_n^F > t\}$. Using first the fact $\{\gamma(t) \geq k+1\} = \{T_k^F \leq t\} \in \mathcal{G}_{T_k^F}$ and then (72), we have

$$E_q \int_0^t \sum_{i=1}^{N} Q_i(s)\, ds \;\leq\; E_q \int_0^{T_{\gamma(t)}^F} \sum_{i=1}^{N} Q_i(s)\, ds$$

$$
\begin{aligned}
&= E_q \sum_{k=0}^{\infty} \int_{T_k^F}^{T_{k+1}^F} \sum_{i=1}^{N} Q_i(s)\ ds\ 1\{k \le \gamma(t) - 1\} \\
&= \sum_{k=0}^{\infty} E_q E \left[\int_{T_k^F}^{T_{k+1}^F} \sum_{i=1}^{N} Q_i(s)\ ds\ |\mathcal{G}_{T_k^F} \right]\ 1\{k \le \gamma(t) - 1\} \\
&\le C_5 E_q \gamma(t),
\end{aligned}
$$

To complete the proof, it suffices to show that for some constant C_6, $E_q\gamma(t) \le C_6 t$, $t \ge 0$. Towards this end, let $\{S_i^n\}_{n=1}^{\infty}$ denote independent copies of the service times at node i and $\tilde{S}^n := \min_{i \le N} S_i^n$, $n = 1, 2, \cdots$.. Also let $\tilde{\gamma}(t) := \inf\{n \ge 1 : \sum_{k=1}^{n} \tilde{S}^k > t\}$. Since at least one service is completed in $[T_n^F, T_{n+1}^F]$ we have that $T_{n+1}^F - T_n^F \ge \tilde{S}^n$, $n = 1, 2, \cdots$, and therefore $\gamma(t) \le \tilde{\gamma}(t)$, $t \ge 0$. But from renewal theory (Prop. 3.2.2 in Ross [10]), we know that $E_q\tilde{\gamma}(t) \le C_6 t$ and the proof is complete.

5 Other Applications

In sections 1 and 2, we have shown in detail how GCQS can be used to model the application of a multi-processor computer system processing parallelizable jobs. We discuss two more applications in this section. The first application involving a radio network appeared in [12] but as mentioned in the introduction, the model considered here is more general since it allows random service times and asynchronous service completions. This application is interesting as it involves the combined choice of appropriate scheduling and routing policies for stabilization.

Consider a radio network that is, a communication network in which nodes communicate by means of radio. It consisting of N nodes the connectivities of which are specified by a directed graph $G = (V, E)$ in the following manner. Each node of V corresponds to a node of the radio network and a directed edge $[i, j] \in E$ denotes that node j is within the transmission range of node i. We assume here that the nodes use spread spectrum signaling for transmission and has only one transceiver. Hence two nodes i and j can simultaneously transmit to nodes k and l if $[i, k], [j, l] \in E$ and these edges do not share a common node; that is, if the set of simultaneously activated edges is a matching of the graph G (see [9] for the definition of a matching of a graph). A packet entering the system at node i is destined for some node of the network and has to be routed to its destination via appropriately chosen intermediate nodes; it leaves the system as soon as it reaches its destination. A message with destination node j is said to belong to class j. A scheduling policy amounts to to choosing a set of links for transmission corresponding to a matching of graph G and a message class for transmission along every activated link. How to schedule the messages so as to sustain the maximum possible arrival rates is the issue here.

The GCQS model of this network consists of $N = V(V-1)$ nodes, each of which is of the form (i, m) corresponding to a message class m, $m \neq i$, at radio network node i. The servers are the links of network, that is, the edges E of graph G and are of the form $[i, j]$ for $i, j \in V$. The routing probabilities are $p((i, m), [i', j'], (j, n)) = 1$ if $i = i'$, $j = j'$, $m = n$, $[i, j] \in E$ and is equal to 0 otherwise. A schedule u has its element $u_{(i,m),[i',j]} = 1$ if $i' = i$, link $[i, j]$ is activated and a message of class m is forwarded along that link; it is 0 otherwise. A schedule u is feasible, that is belongs to $u \in \mathcal{U}$ if the set of activated links forms a matching of G. In sections 2 and 3, we have defined flows ϕ from a node to another node; for this example, the notation simplifies if we define $\tilde{\phi} \in I\!R^{|E|}$ along a link, that is, along edges of E, and scale by the average service time at the originating node; specifically set

$$\tilde{\phi}_{[i,j]} := \beta_i \sum_{m=1}^{V} \phi_{(i,m),(j,m)}, \quad [i, j] \in E.$$

Let $\lambda_{(i,m)}$ denote the exogenous arrival rate of class m customers at node i. We obtain from the definition of $\mathcal{F}$ that if $\phi \in \mathcal{F}$, then $\tilde{\phi}$ satisfies the following equations.

$$\beta_i \sum_{m=1}^{M} \lambda_{(i,m)} + \sum_{j:(j,i) \in E} \tilde{\phi}_{[j,i]} = \sum_{j:(i,j) \in E} \tilde{\phi}_{[i,j]}, \quad i = 1, \cdots, V. \tag{73}$$

The stabilizability condition now follows easily. A matching M of the graph G is a subset of its edges, naturally it can be represented by a binary vector $1_M \in I\!R^{|E|}$ with elements $1_M(e) = 1$ if the edge e belongs to M and 0 otherwise. Let $\mathcal{M} \subset I\!R^{|E|}$ denote the set of all matchings. The necessary stabilizability condition $\mathcal{F} \cap \mathcal{S} \neq \emptyset$ is equivalent to the existence of a vector $\tilde{\phi} \in I\!R^{|E|}$ that satisfies the flow conservation equations (73) and belongs to

$$\text{convex hull } \{1_M : M \in \mathcal{M}\}; \tag{74}$$

for sufficiency, we need $\tilde{\phi}$ to belong to the interior of the set defined in (74). The stabilizing policy π^* works as follows. Let $Q_{(i,m)}(\tau_n)$ denote the queue size at node (i, m) at time τ_n.

During $[\tau_n, \tau_{n+1}]$, π^* chooses to activate the links corresponding to the matching M^* that solves

$$\max_{M \in \mathcal{M}} \left[\sum_{e=[i,j] \in M} \max_{1 \leq m \leq V} \Big(Q_{(i,m)}(\tau_n) \; - \; Q_{(j,m)}(\tau_n) \beta_j / \beta_i \Big) \right] \tag{75}$$

and for every activated link $[i, j]$ of M^*, selects the message class m^* that solves

$$\max_{1 \leq m \leq V} \Big(Q_{(i,m)}(\tau_n) \; - \; Q_{(j,m)}(\tau_n) \beta_j / \beta_i \Big)$$

The optimization problem in (75) is the well known weighted matching problem for which an $O(V^4)$ solution is available (see Papadimitriou [9]). An important issue here is that the true application demands a distributed scheduling policy since the current queue sizes at a nodes is not generally available at other nodes. A distributed implementation of the policy π^* could be attempted with the presence of a control station in the network that collects all the queue size information and coordinates the activation of the links according to matchings. In situations where a distributed implementation of π^* is not feasible, the stability condition provides an upper bound to achievable throughput and the policy π^* provides guidelines on the design of good scheduling policies. A truly distributed stabilizing policy remains an open problem.

As our final example of GCQS, consider cellular networks that have received considerable attention as the prevalent architecture for providing wireless personal communication services. Consider the following cellular network model. The geographical area served by the network is divided into N contiguous regions called cells. Each cell is provided with a base station and the mobile users in a particular cell communicate to the corresponding base station through a radio channel using a particular frequency. The base stations are wired to a communication network. A total of K frequency bands are allocated for communication purposes. At every cell, only one user can communicate using with frequency band k at a time. In addition, depending on the physical proximity of the cells and the proximity of the frequency bands in the electromagnetic spectrum, there are additional restrictions on simultaneous use of the frequency bands for communication. These restrictions can be expressed by the *interference graph* $G_I = (V_I, E_I)$ that consists of $N * K$ nodes, one node $[i, k]$ for each frequency band k in each cell i, and an edge connecting node $[i, k]$ to $[i', k']$ if the frequency bands k in cell i and k' in cell i' can not be used simultaneously. A user located in cell i

located in cell i generates messages that it needs to transmit to the base station of cell i. At the time of message arrival, a frequency band k can be allocated to a user of cell i if there are no edges between node $[i, k]$ and any member of V' in G_I, where V' consists of all the nodes $[i', k']$ of G_I corresponding to a user at cell i' currently transmitting using frequency k'. If such a frequency is not available, then the message is queued. The above cellular network is easily seen to fall within the scope of the constrained queueing model. A cell corresponds to a node of the constrained queueing system and a frequency band corresponds to a server. An activation schedule represents a conflict free allocation of the frequency bands to the cells; that is, $u_{ik} = u_{i',k'} = 1$ if and only if the nodes $[i, k]$ and $[i', k']$ in the interference graph G_I are not connected by an edge. In other words, using a terminology from graph theory, a feasible frequency allocation policy corresponds to an independent set (see Papadimitriou [9]) of the interference graph G_I. There is no routing; that is, $p(i, k, N+1) = 1$ for every i, k. The stabilizability condition is very similar to the one for the parallel processing application in section 2. Specifically, if λ_i is the arrival rate at node (cell) i, then the necessary condition for stability is

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix} \in \text{convex hull} \left\{ \begin{pmatrix} \frac{1}{\beta_1} \sum_{k=1}^K u_{1k} \\ \frac{1}{\beta_2} \sum_{k=1}^K u_{2k} \\ \vdots \\ \frac{1}{\beta_N} \sum_{k=1}^K u_{Nk} \end{pmatrix} : u \in \mathcal{U} \right\}.$$

and for sufficiency, the arrival rate vector needs to belong to the interior of the convex hull. We now describe the stabilizing policy π^* during $[\tau_n, \tau_{n+1}]$. Let $Q_i(\tau_n)$ denote the queue size at node (cell) i at time τ_n. Associate with each node v of the interference graph G_I a weight w_v as follows: $w_v = Q_i(\tau_n)$ if $v = [i, k]$. The policy π^* chooses at time τ_n the frequency allocation policy corresponds to the solution of the following weighted independent set problem:

$$\max \left\{ \sum_{v \in V} w_v \ : \ V \subset V_I, \ V \text{ independent set of } G_I \right\} \tag{76}$$

Unfortunately, (76) is an strongly NP-complete problem (see Papadimitriou [9]) and can only be solved in practice by exhaustive search for problems of small size.

For a special case, the situation simplifies. Consider a linear cellular network that can be used to model the coverage of a highway. The restriction on the use of frequencies is that the same frequency cannot be used in adjacent cells. Therefore the set of edges E_I of the interference graph consists only of edges connecting node $[i, k]$ to node $[i+1, k]$ for every $i \leq N-1$ and $k \leq K$. Each frequency can be allocated to all even numbered cells or to all odd numbered cells. The stabilizing policy at time τ_n acts as follows. If $\sum_{i \leq N, i \text{ even}} Q_i(\tau_n) > \sum_{i \leq N, i \text{ odd}} Q_i(\tau_n)$ then all the frequencies are allocated to even numbered nodes, that is, for all k, $u_{ik} = 1$ if and only if i is even; otherwise the frequencies are allocated to odd numbered nodes.

6 Conclusions

In this paper we considered the problem of scheduling the allocation of resources in a generalized constrained queueing network (GCQS). Because of forced idle periods, a work conserving policy has no meaning in these systems and the problem of stabilizing the system is interesting. A non-preemptive policy is proposed that selects an appropriate server activation schedule at a decision time instant and keeps this schedule activated for a period of time depending on the network state at that instant. The policy is shown to stabilize the system for the largest possible arrival rates and achieve maximum possible throughput. The policy does not need the knowledge of the arrival rates and in some special cases, the knowledge of the service rates is also not needed. The necessary and sufficient conditions for stability are also obtained. Several practical systems that fall within the scope of the GCQS were described and the application of our results was discussed. For some special situations arising in practice, the stabilizing policy is of polynomial complexity and hence implementable with low overhead.

One drawback of the stabilizing policy proposed here is that it does not preserve the FCFS order in service. In applications where it is essential to provide service in the order of job arrivals, the policy is not directly applicable since it does not take into consideration any precedence constraints among the customers. Also, for some applications like the multi-hop radio networks and cellular networks, the scheduling policy proposed here may not be directly applicable if the applications require truly distributed policies. We believe that the results obtained in this paper are still interesting in these cases since the stability condition yields an upperbound for the achievable throughput in the system and the scheduling policy provides guidelines on a good distributed policy may be designed by incorporating the server allocations constraints in a meaningful way. Incorporating precedence constraints in the GCQS and investigating truly distributed scheduling policies are interesting problems to pursue for future research.

Another topic of interest that has not been addressed here is the behavior of the stabilizing policy with respect to delay. We have only shown that delays are finite. As argued in section 2, the policy tries to avoid the build up of large queues and chooses the activation schedules systematically to reduce the forced idle periods. We believe that its delay properties are favorable. It would be interesting to consider the implementation of the policy in the different practical systems to which it applies and to study the delay for different values of the statistical parameters of the system. Finally the problem of allocating the resources such that the delay is minimized is left unaddressed. This is much deeper problem and has been addressed only for a tandem constrained queueing system in Tassiulas et. al. [13]. For more general topologies, the problem essentially remains open.

References

[1] F. Baccelli and Z. Liu, "On the stability of a precedence based queueing discipline,' Res. Report No. 880, INRIA, Sophia Antipolis, France, July 1988.

[2] N. Bambos and J. Walrand, "On stability and performance of parallel processing systems," *Jour. Assoc. Comput. Mach.*, 38, p.429-452, 1989.

[3] N. Bambos and J. Walrand, "Scheduling and stability aspects of a general class of parallel processing system," *Adv. Appl. Prob.*, 25, p.176-202, 1993.

[4] C.A. Courcoubetis, M.I. Reiman and B. Simon, "Stability of a queueing system concurrent service and locking," *SIAM Jour. Computing*, pp. 169-178, vol. 16, 1987.

[5] F. P. Kelly, "Stochastic Models of Computer Communication Systems", *Jour. Royal Stat. Soc.*, 1985.

[6] R.M. Loynes, "The stability of a queue with non-independent interarrival and service times." *Proc. Cambridge Phil. Soc.*, 60, p.569-574, 1964.

[7] D. Mitra and P.J. Weinberger, "Probabilistic models of database locking: solutions, computational algorithms and asymptotics," *Jour. Assoc. Comp. Mach.*, vol. 31, pp. 855-878 1984.

[8] D. Mitra, "Probabilistic models and asymptotic results for concurrent processing with exclusive and nonexclusive locks," *SIAM Jour. Comput.*, vol. 14, pp. 1030-1051, 1985.

[9] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization:Algorithms and Complexity,* Prentice Hall, 1982

[10] S.M. Ross, *Stochastic Processes* , John Wiley and Sons, 1983.

[11] R.G. Parker and R.L. Rardin, *Discrete Optimization*, Academic Press, 1988.

[12] L. Tassiulas, A and Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling for Maximum Throughput in Multihop Radio Networks," *Transactions on Automatic Control*, 37, p.1936-1948, 1992.

[13] L. Tassiulas and A. Ephremides, "Dynamic Scheduling for Minimum Delay in Tandem and Parallel Constrained Queueing Systems," *Annals of Operations Research*, vol. 48, p. 333-355, 1993.

[14] J. Tsitsiklis, C. Papadimitriou and C. Humblet, "The Performance of a Precedence-based queueing discipline", *Jour. Assoc. Comput. Mach.* , vol. 33, pp. 593-602, 1986.