Optimal Server Assignment in Multi-Server Queueing Systems with Random Connectivities

Hassan Halabian, Ioannis Lambadaris, Chung-Horng Lung
Department of Systems and Computer Engineering
Carleton University, 1125 Colonel By Drive, Ottawa, ON, K1S 5B6, Canada
Email: {hassanh, ioannis, chlung}@sce.carleton.ca

Abstract

In this paper, we investigate the problem of assignment of $K$ identical servers to a set of $N$ parallel queues in a time slotted queueing system. The connectivity of each queue to each server is randomly changing with time; each server can serve at most one queue and each queue can be served by at most one server per time slot. Such multi-server queueing systems were widely applied in modeling the scheduling (or resource allocation) problem in wireless networks. It has been previously proven that Maximum Weighted Matching (MWM) is a throughput optimal server assignment policy for such queueing systems [1], [2]. In this paper, first we prove that for a symmetric system with i.i.d. Bernoulli packet arrivals and connectivities, MWM minimizes, in stochastic ordering sense, a broad range of cost functions of the queue lengths including total queue occupancy (or equivalently minimizes the average queuing delay). Then, we extend the model by considering imperfect services where it is assumed that the service of a scheduled packet fails randomly with a certain probability. We prove that the same policy is still optimal for the extended model. We also show that the results are still valid for some extended connectivity and arrival processes (e.g., Poisson arrival process). Second, for the multi-server queueing system with general stationary connectivity and arrival processes, we derive a linear algebraic characterization of the network stability region polytope which is needed to solve general utility optimization problems in such queueing systems.

I. INTRODUCTION

Optimal stochastic control of emerging wireless networks is one of the primary objectives in the design of such networks. In general, the main goal in the stochastic control of wireless networks is to distribute the shared resources in physical (e.g., power) and MAC layers (e.g., radio
interfaces, relay stations and orthogonal sub-channels) among multiple users such that certain stochastic performance attributes are optimized. While various performance attributes including the stable throughput region, power consumption and utility functions of the admitted traffic rates have been studied in many papers [11]–[17], average queueing delay has been considered far less in literature. This is due to the inherent difficulty of delay optimal scheduling problems in queueing systems with time varying channel conditions. In this paper, we consider a discrete time queueing system which is suitable in modeling of orthogonal resource allocation (e.g., radio interfaces, relay stations and communication channels) in multi-user wireless access networks. In our system, we model the available shared resources by a set of identical servers. The model also consists of a set of queues whose connectivities to each server are changing by time randomly. Therefore, resource assignment problem is equivalent to finding a matching between the queues and the servers at each time slot such that some performance objectives are optimized. It has been previously shown that Maximum Weighed Matching (MWM) is throughput optimal for such a system, i.e., it maximizes the stable throughput region [1], [2]. The MWM server assignment policy has also been extensively used in literature for treating the scheduling problem in crossbar packet switches [5], [7]–[9]. In this paper, we first prove that for a symmetric system with i.i.d. Bernoulli packet arrivals and connectivities, MWM not only is throughput optimal, but also it is optimal in minimizing, in stochastic ordering sense, a broad range of cost functions of queue lengths including total queue occupancy (or equivalently average queueing delay). We further characterize the stability region by a set of linear inequalities and obtain an upper bound for the average queueing delay. Such a linear algebraic characterization of stability region is useful in solving utility and fairness optimization problems in these queueing systems.

The rest of this paper is organized as follows. In section II, we will review the related work in this area of research and summarize the contributions we made in this paper. Section III describes the queueing model and the notation required throughout the paper. In section IV, we introduce Maximum Weighted Matching (MWM) policy whose delay optimality for the

1Stability region of a communication network is the closure of the set of all arrival rate vectors for which there exists a resource allocation policy that can stabilize the network.

2We order two discrete time random processes \(A = \{A(t)\}_{t=1}^{\infty}\) and \(B = \{B(t)\}_{t=1}^{\infty}\) stochastically as follows: We say \(A\) is stochastically less than \(B\) and we write \(A \leq_{st} B\) if \(\Pr(A(t) > r) \leq \Pr(B(t) > r)\) for all \(t = 1, 2, \ldots\) and all \(r \in \mathbb{R}\). The notion and relevant properties will be discussed in more detail in Section V-A.
symmetric case of the introduced queueing model is proven in section V. In section V, we present one of the main contributions of this paper, that is proving delay optimality of MWM server assignment policy. In section VI, we will discuss about the network stability region of the introduced queueing model and show that the stability region is a polytope. Then, we derive a linear algebraic characterization of the stability region polytope. In section VII, we present the simulation results where we compare the performance of MWM policy with the performance of two other server assignment policies in terms of average total queue occupancy. Section VIII summarizes the conclusions of the paper.

II. RELATED WORK AND OUR CONTRIBUTIONS

The problem of optimal server allocation in queueing systems with random connectivities was mainly addressed in [2], [3], [10]–[13], [18]–[20]. In [2], the authors introduced the notion of stability region of a general queueing network with time varying connectivities and they proposed Back-pressure algorithm as a throughput optimal resource allocation policy for queueing networks. In [3], they considered a multi-queue single-server queueing system with random connectivities. They characterized the stability region by a set of linear inequalities and also proved that for a symmetric system (with the same arrival and connectivity parameters for all the queues), LCQ (Longest Connected Queue) provides the optimal performance in terms of average queue occupancy.

In [12], Maximum Weight (MW) policy was proposed as a throughput optimal server allocation policy for a multi-queue multi-server queueing system with stationary channel processes. In [13], the authors characterized the network stability region of multi-queue multi-server systems with time varying connectivities. They also obtained an upper bound for the average queueing delay of AS/LCQ policy which is a throughput optimal server allocation policy for such systems. The results were further extended in [14] for more general stationary channel distributions (and not just i.i.d. Bernoulli channels).

The authors in [18] considered a queueing model with a set of symmetrical parallel queues competing to attract service from K identical servers. The connectivity of each queue to all the servers is assumed to be the same at each time slot and during each time slot, each queue can attract at most one server. The authors proposed LCQ policy in which the servers are allocated to the K longest connected queues at each time slot. Using dynamic coupling and stochastic
ordering they proved the optimality of LCQ policy.

The work in [10], [11], [19], [20] focuses on the optimal server allocation problem in multi-queue multi-server queueing systems in terms of average queueing delay. In [10], [11], [19], the authors introduced MTLB (Maximum-Throughput Load-Balancing) policy and showed that this policy minimizes a class of cost functions including total average delay for the case of two symmetric queues. The work in [20] considers this problem for general number of symmetric queues and servers. In [20], a class of Most Balancing (MB) policies was characterized among all work conserving policies which are minimizing, in stochastic ordering sense, a class of cost functions including total queue occupancy. Note that in the model studied in [10], [11], [13], [19], [20], there is no restriction on the number of servers that are serving a queue at each time slot. In [1], it was shown that for a multi-queue multi-server system in which queues are restricted to attract at most one server at each time slot, maximum weighted matching is throughput optimal. The authors also considered the effect of infrequent channel state measurements on the network stability region.

Our contributions in this paper are summarized as follows:

- For a multi-server queueing system where each user is allowed to attract at most one server in each time slot (the multi-server queueing model we elaborate on throughout this paper), we prove that for a symmetric case with i.i.d. Bernoulli arrivals and connectivities (i.e., with the same arrival and connectivity parameters for all the queues), MWM not only is throughput optimal, but also it is optimal in minimizing, in stochastic ordering sense, a broad range of cost functions of queue lengths including total queue occupancy. We then extend the queueing model by considering imperfect services where it is assumed that the service of a scheduled packet fails randomly with a certain probability. We prove that MWM is still optimal for the extended model. We also show that the results are still valid for some extended connectivity and arrival processes (e.g., Poisson arrival process).

- We discuss about the stability region of a general multi-server queueing system with general arrival processes with finite first and second moments. We first argue that the stability region form is a polytope. Then, we will introduce a linear algebraic representation of the stability region polytope (a linear algebraic representation equivalent to the convex hull representation of the stability region) which can be applied to solve utility optimization problems in stochastic control of such queueing systems.
III. Model Description

Before we proceed to introduce the queueing model, we present basic notation often used throughout the paper. Other notations are introduced whenever it is necessary. All the vectors in this paper are row vectors. The operator “⊛” is used for entry-wise multiplication of two matrices. By $\mathbb{1}_K (0_K)$, we denote a row vector of size $K$ whose elements are all identically equal to “1” (“0”). Zero matrix of size $N \times K$ is denoted by $(0)_{N \times K}$. The expectation of random processes (or random variables) is denoted by $E[\cdot]$. The cardinality of a set is shown by $|\cdot|$. The closure of a set is denote by $\text{Cl}(\cdot)$. Finally, the boundary of a set is represented by $\text{bound}(\cdot)$.

We consider a time slotted parallel queueing system with a set of parallel symmetrical queues $N = \{1, 2, ..., N\}$ with infinite buffer space for each queue (See Figure 1). Packets in this system are assumed to have constant length and require one time slot to complete service. The service to this set of queues is provided through a set of identical servers namely $K = \{1, 2, ..., K\}$. The connectivity of each queue $n \in N$ to each server $k \in K$ at each time slot $t$ is random and varying across time slots. We denote the connectivity of queue $n$ to server $k$ at time slot $t$ by $C_{n,k}(t) \in \{0, 1\}$. At each time slot $t$, the connectivity state may be expressed by an $N \times K$ matrix $C(t) = (C_{n,k}(t))$, $n \in N, k \in K, C_{n,k}(t) \in \{0, 1\}$. Therefore, the connectivity process is defined as $\{C(t)\}_{t=1}^{\infty}$ with the state space $\mathcal{C}$ (i.e., $\mathcal{C}$ is the set of all $N \times K$ binary matrices). Note that $\mathcal{C}$ is a finite set with $|\mathcal{C}| = 2^{NK}$. The connectivity process is assumed to have stationary distribution with stationary probabilities $P(c) = \Pr(C(t) = c)$. In fact, $P(c)$ represents the fraction of time slots where the connectivity of the system is in state $c$.

At any time slot, each server can serve at most one packet from a connected non-empty queue. Note that in the system we do not allow server sharing, i.e., a server can serve at most one queue per time slot. We also assume that a queue which is being serviced by a server at a given time slot, cannot get service from other servers during the same time slot.

Let $A_n(t)$ be the packet arrival process (number of packet arrivals) to queue $n$ at time slot $t$. We assume that new arrivals at each time slot are added to the queues at the end of time slot. For these processes, assume that $E[A_n^2(t)] \leq A_{\max}^2 < \infty$ for all $t$. We denote the length of queue $n$ at the end of time slot $t$ (i.e., after adding the new arrivals) by $X_n(t)$. In other words, $X_n(t)$ represents the number of packets in the $n$th queue at the end of time slot $t$ (or beginning of time slot $t + 1$).
**Definition 1:** We define the multi-server queueing system of Figure 1 to be a symmetric system if

- $C_{n,k}(t)$’s are independent and $E[C_{n,k}(t)] = p$ for all $n \in \mathcal{N}$ and $k \in \mathcal{K}$ and $t = 1, 2, \ldots$.
- The arrival variables $A_n(t)$ at each time slot $t$ are i.i.d. Bernoulli random variables with the same parameter $\lambda$ for all $n$ and $t$.

A server assignment policy at each time slot determines an assignment of servers of set $\mathcal{K}$ to the queues of set $\mathcal{N}$. In other words, at each time slot the scheduler has to decide about a bipartite matching (matching in bipartite graphs) between sets $\mathcal{N}$ and $\mathcal{K}$. This should be accomplished based on the available information about the connectivities $C_{n,k}(t)$ and also the queue length vector process at the beginning of time slot $t$ (which is $X(t-1) = (X_1(t-1), X_2(t-1), \ldots, X_N(t-1))$). For a given policy $\pi$, suppose that indicator variable $M_{n,k}^{(\pi)}(t)$ is defined to be “1” if server $k$ is assigned to queue $n$ at time slot $t$ and “0” otherwise. We define the $N \times K$ matrix $M^{(\pi)}(t) = (M_{n,k}^{(\pi)}(t)), \forall n \in \mathcal{N}, k \in \mathcal{K}$ as the employed matching by policy $\pi$ at time slot $t$. Therefore, a server scheduling policy $\pi$ is defined as $\pi = \{M^{(\pi)}(t)\}_{t=1}^{\infty}$. We denote the matching space, i.e., the set of all the possible matchings in an $N \times K$ bipartite graph by $\mathcal{M}$. In other words,

$$\mathcal{M} = \left\{ M_{N \times K} = (M_{n,k}) \forall n \in \mathcal{N}, k \in \mathcal{K} \mid \sum_{k=1}^{K} M_{n,k} \leq 1 \quad \forall n \in \mathcal{N}, \sum_{n=1}^{N} M_{n,k} \leq 1 \quad \forall k \in \mathcal{K} \right\}. \quad (1)$$

According to the above discussion, we can see that the queue length random process $X_n(t)$,
evolves with time according to the following rule.

\[ X_n(t) = \left( X_n(t-1) - \sum_{k=1}^{K} C_{n,k}(t) M_{n,k}^{(\pi)}(t) \right)^+ + A_n(t) \quad \forall n \in \mathcal{N} \]  

(2)

In which, \((\cdot)^+\) returns the term inside the brackets if it is non-negative and zero otherwise. Note that a server can be assigned to an empty queue however it cannot serve it since there is no packet to be served. That is why we have used operator \((\cdot)^+\) in (2).

As we discussed earlier, the queueing model introduced in this section is useful in modeling the resource assignment problem in various systems with shared resources. In wireless communication systems, communication resources such as communication sub-channels, relay stations, etc. are shared among users and therefore resource assignment problem in these networks can be studied using our model (e.g., [1], [17]). For example, consider a relying access network with \(N\) users and \(K\) relays. By modeling the cooperative wireless channel between each user, each relay and the base station as a binary erasure channel, we can apply the introduced queueing model for performance evaluation.

In this paper, random variables are represented by CAPITAL letters and lower case letters are used to represent sample values of the random variables.

In the first part of our analysis, we will consider a symmetric multi-server queueing system (refer to Definition [1]) and we prove delay optimality of Maximum Weighted Matching (MWM) policy for this system. In the second part of the paper, we will consider a general system with stationary connectivity and packet arrival processes with finite first and second moments for which we will characterize the network stability region polytope by a finite set of linear inequalities.

IV. MAXIMUM WEIGHTED MATCHING

In [2], [4], [15], [16], it was shown that Back-pressure algorithm maximizes the stable throughput region of a general data network (i.e., Back-pressure algorithm is throughput optimal). For the model introduced in section III, Back-pressure algorithm is equivalent to solving the following optimization problem at each time slot \(t\) [1].

\[ \text{The reader is encouraged to refer to [2], [4], [15] for more information regarding Back-pressure algorithm.} \]
Maximize: \[
\sum_{n=1}^{N} x_n(t-1) \sum_{k=1}^{K} M_{n,k}(t) c_{n,k}(t)
\]
Subject to: \[
\sum_{k=1}^{K} M_{n,k}(t) \leq 1 \quad (n = 1, 2, \ldots, N) \]
\[
\sum_{n=1}^{N} M_{n,k}(t) \leq 1 \quad (k = 1, 2, \ldots, K),
\]
where \(x(t-1)\) and \(c(t)\) are the values of the random vector \(X(t-1)\) and random matrix \(C(t)\), respectively. Note that finding the solution of problem (3) is equivalent to finding a maximum weighted matching in the \(N \times K\) bipartite graph \(G_t = (\mathcal{N}, \mathcal{K}, \mathcal{E})\) shown in Figure 2. In \(G_t\), \(\mathcal{N}\) and \(\mathcal{K}\) are the two sets of vertices in each part of the graph and \(\mathcal{E} = \{e_{n,k}, \forall n \in \mathcal{N}, \forall k \in \mathcal{K}\}\) is the set of edges between these two parts. Note that in \(G_t\), the associated weight to each edge \(e_{n,k}\) is \(x_n(t-1)c_{n,k}(t)\). A matching in graph \(G_t\) is a sub-graph of \(G_t\) in which no two edges share a common vertex. Any matching \(M^{(\pi)}(t)\) at any time slot \(t\) is corresponding to a sub-graph of \(G_t\) namely \(G_t^{(\pi)} = (\mathcal{N}, \mathcal{K}, \mathcal{E}(\pi))\) in which \(e_{n,k} \in \mathcal{E}(\pi)\) if and only if \(M_{n,k}^{(\pi)}(t) = 1\). Suppose that \(M^{(\text{MWM})}(t) = \{M_{n,k}^{(\text{MWM})}(t)\} \forall n \in \mathcal{N}, k \in \mathcal{K}\) be the matching whose indicator variables are the solution of the optimization problem (3). \(M^{(\text{MWM})}(t)\) can also be presented by the following equation.
\[
M^{(\text{MWM})}(t) = \arg \max_{M \in \mathcal{M}} x(t-1)(c(t) \otimes M)_{1,K}^T
\]
Consequently, we define Maximum Weighted Matching (MWM) server assignment policy as \(\text{MWM} = \{M^{(\text{MWM})}(t)\}_{t=1}^{\infty}\). There are several algorithms to find the maximum weighted matching in bipartite graphs. The most well known algorithm is Hungarian algorithm whose complexity is of \(O((\min\{N, K\})(\max\{N, K\})^2)\) [21].

V. OPTIMALITY OF MWM POLICY

We first review the concepts of stochastic ordering and dynamic coupling arguments in the following subsection. Then, we will prove that optimality of MWM policy is beyond just maximizing the stable throughput region of the system. Indeed, we will prove that in a symmetric multi-server queueing system, MWM policy is optimal in minimizing, in stochastic ordering sense, a class of objective functions of queue length process. Such a class of objective functions...
Fig. 2: Bipartite graph corresponding to problem (3)

includes the total queue occupancy and therefore queueing delay. For brevity we will use the term ”delay optimality“ to refer to the optimality of MWM in this sense.

A. Stochastic Ordering and Dynamic Coupling

In this section, we briefly review the concepts of stochastic ordering (stochastic dominance) and dynamic coupling techniques. Consider two discrete time stochastic processes $A = \{A(t)\}_{t=1}^{\infty}$ and $B = \{B(t)\}_{t=1}^{\infty}$ in $\mathbb{R}$. We say $A$ is stochastically smaller than $B$ and we write $A \leq_{st} B$ if $\Pr(A(t) > r) \leq \Pr(B(t) > r)$ for all $t = 1, 2, \ldots$ and all $r \in \mathbb{R}$ \[22\], \[23\]. Some properties of stochastic ordering are the following. If $A \leq_{st} B$ then $E[A(t)] \leq E[B(t)]$. If $A \leq_{st} B$ then $f(A) \leq_{st} f(B)$ for all non-decreasing functions $f$. $A$ is stochastically smaller than $B$, if there exists process $\tilde{A} = \{\tilde{A}(t)\}_{t=1}^{\infty}$ defined on the same probability space as $B$ with the same probability distribution as $A$ and satisfy $\tilde{A}(t) \leq B(t)$ almost surely for every $t = 1, 2, \ldots$ \[18\]. The last statement is known as coupling of $A$ and $\tilde{A}$. In fact, when applying coupling technique, given process $A$ we try to construct a coupled process $\tilde{A}$ with the same distribution as $A$ and $\tilde{A}(t) \leq B(t)$ a.s. for all $t$. This gives us a tool for comparing the processes $A$ and $B$ stochastically when it is infeasible to derive the distributions of $A$ and $B$ (e.g., in our queueing model when comparing the total occupancy process for different server assignment policies).
B. Delay Optimality of MWM

In this section, we present one of the main results of this paper, that is proving the optimality of MWM with respect to minimization of a class of cost functions of queue lengths including the average queueing delay for the symmetric multi-server queueing system described in Definition 1.

Suppose that \( \mathbb{Z}_+ \) be the set of non-negative integers and \( \mathbb{Z}_+^N \) be the \( N \) dimensional Cartesian space of non-negative integers. We define relation \( \preceq \) on \( \mathbb{Z}_+^N \) as follows.

**Definition 2:** For two vectors \( \bar{x}, \tilde{x} \in \mathbb{Z}_+^N \), we write \( \tilde{x} \preceq x \) if one of the following relations holds:

- **D1:** \( \tilde{x}_n \leq x_n \) for all \( n = 1, 2, \ldots, N \).
- **D2:** \( \tilde{x} \) is obtained by permutation of two distinct elements of \( x \), i.e., \( \tilde{x} \) and \( x \) are different in only two elements \( n \) and \( m \) such that \( \tilde{x}_n = x_m \) and \( \tilde{x}_m = x_n \).
- **D3:** \( \tilde{x} \) and \( x \) are different in only two elements \( n \) and \( m \) such that \( x_n < \tilde{x}_n \leq \tilde{x}_m < x_m \) and the following constraints are satisfied: \( \tilde{x}_n = x_n + 1 \) and \( \tilde{x}_m = x_m - 1 \).

In **D3**, we say that \( \tilde{x} \) is more balanced than \( x \) and can be obtained by decreasing a larger element of \( x \) (i.e., \( m \)) by “1” and increasing a smaller element (i.e., \( n \)) by “1”. We call such an interchange as a balancing interchange on vector \( x \). Thus, the result of a balancing interchange on a vector \( x \) would be a vector \( \tilde{x} \) such that \( \tilde{x} \preceq x \). Suppose that vector \( x \in \mathbb{Z}_+^N \) represents the queue length vector at a given time slot. Then a balancing interchange is equivalent to taking a packet from a larger queue and adding it to a smaller queue.

We define the partial order \( \preceq_p \) on \( \mathbb{Z}_+^N \) as the transitive closure of relation \( \preceq \) [24]. In other words, \( \tilde{x} \preceq_p x \) if and only if \( \tilde{x} \) is obtained from \( x \) by performing a sequence of reductions, permutations of two elements and/or balancing interchanges. When \( x \) and \( \tilde{x} \) are two queue length vectors, we write \( \tilde{x} \preceq_p x \) if and only if queue length vector \( \tilde{x} \) is obtained from \( x \) by applying a series of packet removals, two queues permutations and balancing interchanges.

We define \( F \) as the class of real-valued functions on \( \mathbb{Z}_+^N \) that are monotone and non-decreasing with respect to the partial order \( \preceq_p \), i.e.,

\[
f \in F \iff \tilde{x} \preceq_p x \Rightarrow f(\tilde{x}) \leq f(x).
\]  

We can easily check that function \( f(x) = \sum_{n=1}^{N} x_n \) belongs to \( F \). This function captures the total queue occupancy of the system.
Let \( X'(t) = (X'_1(t), X'_2(t), \ldots, X'_N(t)) \) denote the queue length vector at time slot \( t \) exactly after serving the queues according to a server assignment policy \( \pi \) and before adding the new arrivals of time slot \( t \), i.e.,

\[
X'_n(t) = \left( X_n(t-1) - \sum_{k=1}^{K} C_{n,k}(t) M^{(\pi)}_{n,k}(t) \right)^+.
\]

Given \( x'(t) \) as a sample value of random vector \( X'(t) \), we define a balancing server reallocation at time slot \( t \) as follows.

**Definition 3:** A balancing server reallocation on vector \( x'(t) \) is a matching that results in vector \( \tilde{x}'(t) \) such that one of the following conditions is satisfied.

(C1) \( \tilde{x}'_n(t) \leq x'_n(t) \) for all \( n = 1, 2, \ldots, N \) and there exists \( m \in \{1, 2, \ldots, N\} \) such that \( \tilde{x}'_m(t) < x'_m(t) \).

(C2) \( \tilde{x}'(t) \) and \( x'(t) \) are different in only two elements \( n \) and \( m \) such that \( x'_n(t) < \tilde{x}'_n(t) \leq \tilde{x}'_m(t) < x'_m(t) \) and the following constraints are satisfied: \( \tilde{x}'_n(t) = x'_n(t) + 1 \) and \( \tilde{x}'_m(t) = x'_m(t) - 1 \).

Figures 3a and 3b show two examples of balancing server reallocations in two sample graphs. In these figures, the original allocations are specified by solid lines while the balancing reallocations are specified by dashed lines.

Consider an arbitrary server assignment policy \( \pi \) with the allocation variables \( \{M^{(\pi)}_{n,k}(t)\}_{t=1}^{\infty} \) for all \( k \in K \) and \( n \in N \). We introduce Matching Weight (MW) index associated to the server allocation policy \( \pi \) at time slot \( t \) by

\[
MW_{\pi}(t) = \sum_{n=1}^{N} x_n(t-1) \sum_{k=1}^{K} C_{n,k}(t) M^{(\pi)}_{n,k}(t) \tag{7}
\]

Note that MW index is exactly the objective of the optimization problem (3). According to Definition 3 and definition of MW index, we can prove the following Lemma.

**Lemma 1:** For a given policy \( \pi \) employing matching \( M^{(\pi)}(t) \) at time slot \( t \), by applying a balancing server reallocation at time slot \( t \) (if there exists any), we will have a new policy \( \tilde{\pi} \) differing from \( \pi \) only at time slot \( t \) such that \( MW_{\pi}(t) < MW_{\tilde{\pi}}(t) \).

The detailed proof of the lemma is given in Appendix I-A. Based on Lemma 1, we can state the following corollary.

\footnote{4A balancing server reallocation may consist of one or more server reassignment.}
Fig. 3: Examples of balancing server reallocations (the weight \( c_{n,k}(t)x_n(t-1) \) of each edge \((n, k)\) is also shown)

**Corollary 1:** For a given policy \( \pi \) at time slot \( t \), if \( MW_\pi(t) \) is maximized, i.e., policy \( \pi \) employs a maximum weighted matching at time slot \( t \), then there exists no balancing server reallocation at that time slot.

Note that Lemma 1 states that any balancing reallocation increases the matching weight index. However, it does not imply the existence of a balancing server reallocation when \( MW_\pi(t) \) is not maximized. In the following we will prove the existence result i.e., the reverse of Lemma 1.

**Lemma 2:** For a given policy \( \pi \) at time slot \( t \), if \( MW_\pi(t) \) is not maximized, i.e., \( MW_\pi(t) < MW_{MWM}(t) \), then there exists a balancing server reallocation at that time slot.

For the detailed proof, please refer to Appendix I-B.

By \( \Pi_{MWM} \), we denote the set of all policies who employ maximum weighted matching at all time slots. We also define \( \Pi_t \) as the set of all policies that employ maximum weighted matching exactly until time slot \( t \) (including \( t \)). We can easily observe that \( \Pi_{t-1} \supseteq \Pi_t \) and \( \Pi_{MWM} = \bigcap_{t=1}^{\infty} \Pi_t \). From Lemmas 1 and 2 we conclude that given a policy \( \pi \in \Pi_{t-1} \) which is using an arbitrary matching at time slot \( t \), we can reach to a policy \( \pi^* \in \Pi_t \) by applying a sequence of balancing server reallocations. Suppose that \( h_t^\pi \) represents the number of balancing server reallocations required to convert the employed matching in policy \( \pi \) at time slot \( t \) to a maximum weighted matching. In this case, we say that the distance of \( \pi \) from \( \Pi_t \) is \( h_t^\pi \) balancing server reallocations. Note that if the distance of \( \pi \) from \( \Pi_t \) is \( h_t^\pi \), after applying the first balancing server reallocation we get to a policy \( \tilde{\pi} \) whose distance from \( \Pi_t \) is \( h_t^\pi - 1 \) balancing
server reallocations. By repeating this procedure we finally get to a policy whose distance to \( \Pi_t \) is zero, i.e., it belongs to \( \Pi_t \). By \( \Pi_t^h \) \((0 \leq h \leq h_t^\pi)\) we denote the set of all server assignment policies in \( \Pi_{t-1} \) whose distance from \( \Pi_t \) is at most \( h \) balancing sever reallocations. Note that \( \Pi_0^0 = \Pi_t \).

Consider any two policies \( \pi \) and \( \tilde{\pi} \) such that \( f(\tilde{X}) \leq_{st} f(X) \) where \( X = \{X(t)\}_{t=1}^\infty \) and \( \tilde{X} = \{\tilde{X}(t)\}_{t=1}^\infty \) are the queue length processes when policies \( \pi \) and \( \tilde{\pi} \) are applied, respectively. For such a system, we say policy \( \tilde{\pi} \) dominates \( \pi \). If \( \tilde{\pi} \) dominates \( \pi \) we have \( E[f(\tilde{X})] \leq E[f(X)] \). Based on the above discussion about stochastic dominance, we can prove the following Lemma.

**Lemma 3:** For any policy \( \pi \in \Pi_t^h \) and \( 0 < h \leq h_t^\pi \) we can construct a policy \( \tilde{\pi} \in \Pi_t^{h-1} \) such that \( \tilde{\pi} \) dominates \( \pi \).

For the detailed proof, please refer to Appendix I-C. Using Lemma 3, we can prove the main result of this section in the following Theorem.

**Theorem 1:** The Maximum Weighted Matching policy dominates any server assignment policy.

**Proof:** Let \( \pi_0 \) be any arbitrary policy. Then \( \pi_0 \in \Pi_0 = \Pi_1^{H_1} \) where \( H_1 = h_1^{\pi_0} \). By applying Lemma 3 repeatedly, we can construct a sequence of policies such that each policy dominates the previous one. Thus, we obtain policies that belong to \( \Pi_0 = \Pi_1^{H_1}, \Pi_1^{H_1-1}, \Pi_1^{H_1-2}, ..., \Pi_1^0 = \Pi_1 \). The last policy is called \( \pi_1 \). Note that \( \pi_1 \in \Pi_2^{H_1} \) where \( H_2 = h_2^{\pi_1} \). By recursively continuing such an argument we obtain a sequence of policies \( \pi_t \in \Pi_t, t = 1, 2, ... \) such that \( \pi_j \) dominates \( \pi_i \) for \( j > i \). Note that this sequence of policies defines a limiting policy \( \pi^* \) that agrees with MWM at all time slots. Thus, \( \pi^* \) is an MWM policy who dominates all the previous policies, including the starting policy \( \pi_0 \).

**C. Extensions**

1) **Imperfect Services:** We can extend Theorem 1 for the case where the service of a scheduled packet by a connected server fails randomly with a certain probability. This is in analogous to realistic wireless networks operation where service failures usually occur due to unexpected and unpredictable effects of noise, interference, etc. In case of a packet service failure, the packet will be kept in the queue and will be rescheduled and retransmitted in future time slots.

Choosing \( f(x) = \sum_{n=1}^N x_n \), we conclude that the average queue occupancy (or equivalently average queueing delay) of policy \( \tilde{\pi} \) is smaller than that of policy \( \pi \).
We denote the binary random variable associated to successful/unsuccessful service of queue $n$ provided by server $k$ at time slot $t$ by $Q_{n,k}(t)$. Note that $Q_{n,k}(t) \in \{0, 1\}$ and is "$1$" ("$0$") when the service is successful (unsuccessful). We assume that $Q_{n,k}(t)$’s are i.i.d. Bernoulli random variables with the same success probability $q$. Note that parameter $q$ (the same as parameters $\lambda$ and $p$) does not involve in our analysis and in fact what is necessary in our analysis is the symmetry of the system (i.e., $E[Q_{n,k}(t)] = q, \forall n, k$). The queue lengths are then updated at the end of each time slot by the following rule.

$$X_n(t) = (X_n(t-1) - \sum_{k=1}^{K} C_{n,k}(t)M_{n,k}(t)Q_{n,k}(t))^{+} + A_n(t) \quad \forall n \in \mathcal{N}$$

(8)

Note that the network controller cannot observe the variables $Q_{n,k}(t)$ and from his perspective they are assumed to be random. The random vector $X'(t)$ is defined similar to equation (6). Hence, $X'(t)$ represents the queue lengths before adding the new arrivals of time slot $t$ as if all the services at that time slot are successful.

For such a system we can easily see that Lemmas 1 and 2 are still valid. We can also extend Lemma 3 for the system with service failures by considering random variables $Q_{n,k}(t)$ in our dynamic coupling argument. The detailed analysis is brought in Appendix I-D. By applying the same approach as what we did for Theorem 1, we can similarly prove the delay optimality of MWM policy for the system with imperfect services.

2) Extensions for Connectivity and Arrival Processes: The arguments in Lemma 3 and Theorem 1 remain valid if the i.i.d. assumption for connectivity and arrival processes is relaxed as follows; we will consider connectivity and arrival processes which follow conditional permutation invariant distributions. Given even $\mathcal{H}$ (which is usually used to denote the history of the system), we define a conditional multivariate probability distribution $f(y_1, y_2, ..., y_n \mid \mathcal{H})$ to be permutation invariant if for any permutation of the variables $y_1, y_2, ..., y_n$ namely $y'_1, y'_2, ..., y'_n$ we have $f(y_1, y_2, ..., y_n \mid \mathcal{H}) = f(y'_1, y'_2, ..., y'_n \mid \mathcal{H})$. We can readily see that for all the connectivity and arrival processes whose joint distributions at each time slot given the history of the system (i.e., $f_{A(t)}(a_1, a_2, ..., a_N \mid \mathcal{H})$ and $f_{C(t)}(c_{1,1}, c_{1,2}, ..., c_{N,K-1}, c_{N,K} \mid \mathcal{H})$) are permutation invariant, Lemma 3 and Theorem 1 are still valid and therefore MWM is delay optimal.

---

6By history of the system we mean all the channel states, arrivals and matchings of the previous time slots.
We also consider the generalization of Theorem 1 for non-Bernoulli arrival processes. Suppose that the number of arrivals to each queue can be represented by the summation of some i.i.d. Bernoulli random variables, i.e., has Binomial distribution. Also suppose that $A_n(t) \leq A_{max}$ for all $n \in N$ and all $t$. In this case, we can create a new (virtual) system in which after each time slot we append $A_{max} - 1$ virtual time slots and put the connectivities all equal to zero, i.e., for each virtual time slot $t$, $C_{n,k}(t) = 0, \forall n \in N, \forall k \in K$. We then distribute the arrivals of the actual time slot among these $A_{max}$ time slots (one actual time slot and $A_{max} - 1$ virtual time slots) randomly such that at each time slot at most one packet arrival occurs. Since the connectivities and the arrivals in both systems are permutation invariant, we can still prove Theorem 1 for the virtual system. Note that the operation of the two systems (the original system and the virtual system) are the same. Therefore, we can conclude that Theorem 1 is also valid for a multi-server system with Binomial arrival processes. As Poisson distribution asymptotically approximates the Binomial distribution when $A_{max}$ goes to infinity, Theorem 1 will be still valid for i.i.d. Poisson arrival processes.

VI. LINEAR ALGEBRAIC CHARACTERIZATION OF STABILITY REGION

In this section of the paper, we derive a linear algebraic characterization of the stability region for the queueing system of Figure 1. Linear algebraic representation of the stability region is specially useful in solving stochastic optimization problems and finding optimal flow control strategies in queueing networks.

A. Motivation

**Stability region** or **network stability region** of a communication network is the closure of the set of all arrival rate vectors for which there exists a resource allocation policy that can stabilize the system [2], [4], [15]. Having the stability region $\Lambda$ of a network characterized by a set of convex inequalities, we can obtain a precise solution for network utility optimization problems with the general form of (9).

Maximize: $\sum_{n=1}^{N} f_n(r_n)$

Subject to: $r = (r_1, r_2, ..., r_N) \in \Lambda$

$0 \leq r_n \leq \lambda_n \ \forall n = 1, 2, ..., N$
In (9), $N$ is the number of traffic sources, $r = (r_1, r_2, ..., r_N)$ denotes the admitted rate vector, $\lambda_n$ is the input rate of each source $n$ and $\Lambda$ denotes the network stability region. Functions $f_n(r_n)$ are non-decreasing and concave utility functions. The choice of $f_n(r_n)$ determines the desired fairness behavior of the network [25], [26].

To solve the optimization problem (9), we need to have a precise description of stability region $\Lambda$. Although stability region is already defined and characterized in [2]–[4], [6], [12], [15], [16] as the convex hull of a set of fixed points, such a description cannot be applied to solve problems similar to (9) as it does not provide explicit mathematical relations in the form of convex equalities or inequalities to put as the constraints of problem (9). In this section, we will derive a linear algebraic representation of the stability region of multi-server queueing system of Figure 1.

B. Preliminaries

1) Stability Region and Throughput Optimal Server Assignment: The concepts of stability region and throughput optimal resource allocation in queueing networks were first introduced in the seminal work [2], [3] by Tassiulas and Ephremides. The idea was further developed by Neely et al. for more general queueing networks in [4], [15], [16]. We first review the definition of strong stability in queueing networks [4]. Stability region is then described based on this definition. Consider a discrete time single queue with an arrival process $A(t)$ and service process $\mu(t)$. Let $X(t)$ denote the queue length process at the end of time slot $t$. Hence, $X(t)$ evolves with time according to the following equation.

$$X(t) = (X(t-1) - \mu(t))^+ + A(t)$$

(10)

**Definition 4:** A single queue evolving according to (10) is called strongly stable if it has bounded time averaged expected backlog [4], i.e.,

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} E[X(\tau)] < \infty.$$  

(11)

Definition 4 is extended to define stability in a queueing network as follows [4].

**Definition 5:** A queueing system is called to be strongly stable if all the queues in the system are strongly stable. Specifically, for a network consisting of $N$ queues, the system is strongly
stable if the time averaged expected aggregated backlog in the network is bounded, i.e.,

\[
\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum_{n=1}^{N} E[X_n(\tau)] < \infty.
\]  

(12)

Consider the queueing system introduced in section III (Figure 1). Assume that the arrival rate to each queue \( n \) is denoted by \( \lambda_n \) and therefore the arrival rate vector is denoted by \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N) \). A server assignment policy \( \pi \) is said to stabilize the system if by applying policy \( \pi \) the system is strongly stable. Let \( \Lambda_\pi \) represent the set of arrival rate vectors for which \( \pi \) can stabilize the system. \( \Lambda_\pi \) is called the stability region of policy \( \pi \). Now, consider the closure of the union of the stability regions of all the possible server assignment policies and let us denote it by \( \Lambda \), i.e., \( \Lambda = Cl(\bigcup_\pi \Lambda_\pi) \). \( \Lambda \) is called network stability region (also called network layer capacity region in the literature [4], [13]–[16]). Therefore, network stability region is the closure of the set of all arrival rate vectors for which there exists a policy that can stabilize the system. We observe that the network stability region is independent of resource allocation policy and is in fact a specific characteristic of any network. In section VI-C, we will describe the characterization of stability region as the convex hull of a set of fixed points in \( \mathbb{R}^N_+ \) (i.e., a convex polytope) introduced in [4], [7], [15].

2) Fundamental Concepts of Polytopes: In this part, we briefly review the notion of convex polytope and some of its fundamental properties. These concepts are needed when describing the stability region of the system in sections VI-C and VI-D.

**Definition 6:** A convex polytope is defined as the convex hull of a finite set of points [27], [28].

According to the Weyl’s Theorem [27], a polytope in \( \mathbb{R}^N \) always can be expressed by a set \( \mathcal{P} = \{ x \in \mathbb{R}^N \mid \alpha_\ell x^T \leq \beta_\ell \ , \ell = 1, 2, \ldots, L \} \) for some positive integer \( L \) and \( \alpha_\ell \in \mathbb{R}^N \) and \( \beta_\ell \in \mathbb{R} \).

**Definition 7:** An equality \( ax^T \leq b \) is called valid for a polytope \( \mathcal{P} \) if for every point \( x_0 \in \mathcal{P} \), \( ax_0^T \leq b \).

**Definition 8:** A face of polytope \( \mathcal{P} \) is defined as \( \mathcal{F} = \{ x \in \mathcal{P} \mid ax^T = b \} \) where inequality \( ax^T \leq b \) is a valid inequality for \( \mathcal{P} \).

Note that if the valid inequality \( ax^T \leq b \) does not touch polytope \( \mathcal{P} \), the associated face to this inequality is empty set \( \emptyset \). We call the hyperplane \( ax_0^T = b \) associated to the valid inequality \( ax_0^T \leq b \) a face defining hyperplane for \( \mathcal{P} \) if its associated face is not empty. In other words,
\( ax_0^T = b \) is a face defining hyperplane for \( P \) if it touches \( P \) at least at one point. Note that \( P \) has finitely many faces. However, the face defining hyperplanes of a polytope can be infinite.

**Definition 9:** A facet of polytope \( P \) is a maximal face distinct from \( P \) \(^{[23]}\). In other words, all the faces of \( P \) with dimension \( \dim(P) - 1 \) are called facets of \( P \) \(^{[23]}\).

For polytope \( P = \{ x \in \mathbb{R}^N | \alpha_\ell x^T \leq \beta_\ell, \ell = 1, 2, ..., L \} \) an inequality is redundant if polytope \( P \) remains unchanged by removing the inequality. **Redundancy theorem** in polytopes states that to describe a polytope, only facet defining hyperplanes are sufficient \(^{[27]}\).

### C. Convex Polytope Representation of Stability Region

Consider the class of deterministic server assignment policies \( G \) in which at each time slot \( t \), a policy \( g \in G \) assigns the servers to the queues according to a predetermined matching depending on the connectivity state of the system (i.e., matrix \( C(t) \)). More specifically, for each policy \( g \in G \), there exists a one to one mapping from the connectivity matrix space \( C \) to the matching space \( M \), namely \( M(g) : C \mapsto M \). At each time slot \( t \), the scheduler who is employing server assignment policy \( g \) observes the connectivity matrix of the system, i.e., \( c(t) \in C \) and assigns the servers to the queues according to matching \( M(g)(c(t)) \). Note that since \( M < \infty \) and \( |C| < \infty \), the set \( G \) of all the deterministic server assignment policies is finite.

Each deterministic policy \( g \) provides an average service rate for each queue \( n \). Let \( R_n^{(g)} \) denote the averaged service rate provided to queue \( n \) and \( R^{(g)} = (R_1^{(g)}, R_2^{(g)}, ..., R_N^{(g)}) \) is the vector of average service rates provided by policy \( g \). By conditioning on \( C(t) \), we can readily show that for each deterministic server assignment policy \( g \), \( R^{(g)} \) is obtained by the following equation.

\[
R^{(g)} = \left( \sum_{c \in C} P(c) (c \oplus M^{(g)}(c)) \frac{1^T}{K} \right)^T \tag{13}
\]

Note that each rate vector \( R^{(g)} \) determines a single point in \( \mathbb{R}_+^N \). Now, consider the convex hull of all the points \( R^{(g)} \), \( g \in G \) in \( \mathbb{R}_+^N \), i.e.,

\[
\mathcal{R} = \text{conv.hull}(R^{(g)}). \tag{14}
\]

\(^7\)For a polytope \( P \subset \mathbb{R}^N \), dimension of \( P \) is equal to \( N \) minus the maximum number of linearly independent equations satisfied by all the points in \( P \).
Set $\mathcal{R}$ contains all the achievable service rate vectors for the queueing system of Figure 1. Each point $R^\star$ in $\mathcal{R}$ can be represented by a convex combination of a finite set of points, $\{R^{(g)}, g \in \mathcal{G}\}$, i.e.,

$$R^\star = \sum_{i=1}^{\vert \mathcal{G} \vert} p_i R^{g_i}, \quad g_i \in \mathcal{G}, \quad \sum_{i=1}^{\vert \mathcal{G} \vert} p_i = 1, \quad p_i \geq 0.$$  

(15)

Hence, in order to achieve an arbitrary service rate vector $R^\star$, it is enough to select policy $g_i$ in $p_i$ fraction of time slots (a randomized policy). According to the definition of convex polytope [27], [28] and the fact that set $\mathcal{G}$ is finite, set $\mathcal{R}$ represents a convex polytope in $\mathbb{R}^N_+$. Based on the definition of stability region and the analysis in [4], [15], [16], we can conclude that polytope $\mathcal{R}$ specifies the stability region of the queueing system of Figure 1. Specifically, we can show that if the system is stable, we should have $\lambda \in \mathcal{R}$ and also if $\lambda \in \mathcal{R} - \text{bound}(\mathcal{R})$, then there exists a server assignment policy that can stabilize the system.

A server assignment policy $\pi^\star$ is called throughput optimal if it can stabilize the system for all the arrival rate vectors strictly inside the network stability region. Therefore, $\pi^\star$ is throughput optimal if its stability region coincides with the network stability region. The MWM policy is proven to be a throughput optimal server assignment policy, i.e., it will stabilize the system as long as the input rates are inside the stability region. Note that MWM does not need to know the stability region to act. However, for the cases where the input rates are outside the stability region, in order to guarantee a measure of fairness among the admitted rates of different users, we need to solve network utility optimization problems similar to (9). To solve such network optimization problems precisely, we need to have an explicit characterization of network stability region in terms of linear or non-linear convex equalities and inequalities to be used as the constraints of problem (9).

D. Linear Algebraic Representation of stability region

In the following, we will characterize the stability region of the queueing system of Figure 1 by a finite set of linear inequalities. We introduce the departure matrix $H_{N \times K}(t) = (H_{n,k}(t)), n \in \mathcal{N}, k \in \mathcal{K}$ in which $H_{n,k}(t)$ represents the number of packets served by server $k$ from queue $n$ at time slot $t$. Obviously $H_{n,k}(t) \in \{0,1\}$ and $H_{n,k}(t) \leq C_{n,k}(t) \leq 1$. Hence, the departure process from queue $n$ at time slot $t$ can be written as $\sum_{k=1}^{K} H_{n,k}(t)$. The following equation
describes the evolution of the queue length process in time.

\[ X_n(t) = X_n(t-1) - \sum_{k=1}^{K} H_{n,k}(t) + A_n(t) \quad \forall n \in \mathcal{N} \quad (16) \]

For a strongly stable queueing system the following is true.

**Lemma 4:** If the queueing system of Figure 1 is strongly stable under a server assignment policy \( \pi \), then for any vector \( \alpha = (\alpha_1, \alpha_2, ..., \alpha_N) \in \mathbb{R}^N \) we have

\[
\alpha \lambda^T = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \alpha E[H(\tau)]1_k^T.
\]

(17)

The proof is given in Appendix II-A.

In other words, for a strongly stable system, the weighted arrival rates to the queues is equal to the time averaged weighted expected departure from the queues. We further can prove the following Lemma.

**Lemma 5:** If the queueing system of Figure 1 is strongly stable under a server assignment policy \( \pi \), then for all \( \alpha \in \mathbb{R}_+^N \)

\[
\alpha \lambda^T \leq \sum_{c \in \mathcal{C}} P(c) \max_{M \in \mathcal{M}} (\alpha(c \oplus M)1_k^T).
\]

(18)

The proof is given in Appendix II-B.

As we can see, the set of inequalities in (18) is an infinite set. Note that each inequality in (18) determines a valid inequality for polytope \( \mathcal{R} \). It can be shown that the hyperplanes associated to the valid inequalities of (18) are all face defining hyperplanes of polytope \( \mathcal{R} \). Therefore, the hyperplane associated to each of the valid inequalities of (18) touches polytope \( \mathcal{R} \) at least at one point. Towards this, let \( \mathcal{M}^\alpha \) denote the set of matchings \( \{M^\alpha(c), c \in \mathcal{C}\} \) that maximizes the right hand side of (18), i.e.,

\[
M^\alpha(c) = \arg \max_{M \in \mathcal{M}} \alpha(c \oplus M)1_k^T.
\]

(19)

Note that \( \mathcal{M}^\alpha \) is not unique and there may be more than one set of matchings \( \mathcal{M}^\alpha \) whose elements maximize \( \alpha(c \oplus M)1_k^T \). According to (13) and (14), \( (\sum_{c \in \mathcal{C}} P(c) (c \oplus M^\alpha(c))1_k^T)^T \in \mathcal{R} \). Furthermore, since \( \sum_{c \in \mathcal{C}} P(c) (c \oplus M^\alpha(c))1_k^T = \alpha \sum_{c \in \mathcal{C}} P(c) (c \oplus M^\alpha(c))1_k^T \), according to (18) this point (i.e., \( (\sum_{c \in \mathcal{C}} P(c) (c \oplus M^\alpha(c))1_k^T)^T \)) is located on the hyperplane associated to (18). Hence, the set of inequalities of (18) determines all the non-empty faces of polytope \( \mathcal{R} \).
According to redundancy theorem in polytopes (refer to Section VI-B2 and also [27], [28]), not all the face defining hyperplanes of a polytope are necessary to describe a polytope; i.e., the inequalities corresponding to the facets of a polytope are sufficient to characterize a polytope. In the following theorem, we will show that not all $\alpha \in \mathbb{R}^N_+$ are required to describe polytope $\mathcal{R}$. In fact, Theorem 2 states that just $\alpha \in \{0, 1\}^N - \{0_N\}$ are sufficient to describe facet defining hyperplanes of polytope $\mathcal{R}$.

**Theorem 2:** If there exists a server assignment policy $\pi$ under which the system is stable, then

$$\alpha \lambda^T \leq \sum_{c \in \mathcal{C}} P(c) \max_{M \in \mathcal{M}} \left( \alpha(c \otimes M) \mathbb{1}_K^T \right), \quad \alpha \in \{0, 1\}^N - \{0_N\}. \quad (20)$$

The proof is given in Appendix II-C.

Note that Theorem 2 describes the necessary conditions for the stability of the system. For completeness of our study we also have to show that all the rate vectors inside polytope $\mathcal{R}$ are rate stable. In other word, there exists a server assignment policy who can stabilize the system for all the rate vectors inside polytope $\mathcal{R}$. In the following theorem, we will prove that for a system satisfying conditions (21) below (which is describing $\mathcal{R} - bound(\mathcal{R})$), the MWM policy stabilizes the system. We further obtain an upper bound for average expected aggregate backlog in the system.

**Theorem 3:** The multi-server queueing system of Figure 1 is stable under MWM if

$$\alpha \lambda^T < \sum_{c \in \mathcal{C}} P(c) \max_{M \in \mathcal{M}} \left( \alpha(c \otimes M) \mathbb{1}_K^T \right), \quad \alpha \in \{0, 1\}^N - \{0_N\}. \quad (21)$$

Furthermore, the following bound for the average expected “aggregate” occupancy holds.

$$\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{t=0}^{t-1} \sum_{n=1}^{N} E[X_n(\tau)] \leq \frac{NA_{max}^2 + (\min\{N, K\})^2}{2\delta} \quad (22)$$

where $\delta$ is a positive real number such that $\lambda + \delta \mathbb{1}_N \in bound(\mathcal{R})$.

The proof of Theorem 3 follows by applying Lyapunov stability analysis introduced in [4], [15], [16]. For completeness of our analysis the proof is provided in Appendix II-D.

According to Theorem 2 and 3 and the definition of network stability region, we can conclude that for the multi-server queueing system of Figure 1 with stationary connectivity distribution and stationary arrival processes, the stability region is characterized by (20).
VII. SIMULATION RESULTS

We have evaluated the performance of MWM policy with two other server assignment policies in terms of average queueing delay. The other two policies are described in the following.

- **Maximum Matching** (MM) policy that applies the maximum matching on matrix $C(t)$. In other words, the maximum matching policy at each time slot $t$ employs a server assignment (or matching) $M^{(MM)}(t)$ which is obtained by $M^{(MM)}(t) = \underset{M \in M}{\text{arg max}} (c(t) \circ M)_{1:k}^T$.

- A heuristic policy that assigns the servers to the queues at each time slot according to the following rule: It selects a server randomly and assigns it to its longest connected queue. Then, updates the set of servers by removing the selected server from $\mathcal{K}$ and the set of queues (i.e., $\mathcal{N}$) by removing the queue to which the selected server was assigned. This procedure is repeated $K$ times. Note that, for some servers the updated set $\mathcal{N}$ may be empty (e.g., when $K > N$) and therefore those servers are not assigned to any queue.

**Algorithm 1** Heuristic Policy Pseudocode

**Input**: $\mathcal{N}$, $\mathcal{K}$, $c(t)$ and $x(t-1)$

**Initialize**: $M^{(H)}(t) = (0)_{N \times K}$

**for** $i = 1$ **to** $K$ **do**

Choose a server $k^* \in \mathcal{K}$ randomly

if $\mathcal{N} \neq \emptyset$

$n^* \leftarrow \text{arg max}_{n \in \mathcal{N}} c_{n,k^*}(t) x_n(t-1)$

$M^{(H)}_{n^*,k^*}(t) \leftarrow 1$

$\mathcal{N} \leftarrow \mathcal{N} - \{n^*\}$

**endif**

$\mathcal{K} \leftarrow \mathcal{K} - \{k^*\}$

**end**

Return $M^{(H)}(t)$

The main idea of the heuristic policy is coming from Longest Connected Queue (LCQ) policy which was proven to be optimal for a single server system [3]. For multi-server system, we will use the same principle for each server. However, the order in which servers are selected for assignment is random. We have arranged a comprehensive set of simulations in which we
investigate the effects of number of servers $K$, probability of connectivity $p$ and probability of service success $q$ on the performance of the aforementioned policies. In all the simulations, we set $N = 8$ and the arrivals are i.i.d. Poisson distributed.

Figures 4-6 illustrate the simulation results. As we can see in all the cases, MWM has better performance with respect to other policies in terms of average queue occupancy or average queueing delay. Figure 4 shows the simulation results for $K = 4$, $q = 0.8$ and $p = 0.2, 0.5, 0.8$. In these cases, since the number of servers is relatively low, server assignment will be more competitive. MWM which was proven to be the optimal policy performs such that it tries to balance the queues. The heuristic policy follows the same principle. However, since the selection of servers for assignment is random there are some cases where two or more servers have the same longest connected queue. In such cases, the order of selecting the servers for assignment is effective in the performance of the system. Maximum Matching policy however, does not try to balance the queues. Indeed, it does not consider the queue lengths in its assignments and that is why it performs worse than the other two policies. Note that as the connectivity of the system gets larger, the performances of MWM and the heuristic policy get closer. It is worth mentioning that the heuristic policy introduced here performs the same as MWM for $K = 1$ (which is equivalent to LCQ whose optimality was previously shown in [3]).

Figure 5 shows the results for 8 servers. In this case, since the number of servers is relatively high and comparable to the number of queues, in MWM and MM policies each queue gets service with high probability when the probability of server connectivities tends to 1. As the connectivity probability gets smaller, the difference in performance of MWM and MM becomes more apparent. Note that in this case, the heuristic policy performs worse than the other two policies since it is more probable to lead to cases where two or more servers are connected to the same longest connected queue. As the number of servers becomes larger, we expect MM to perform the same as MWM as in this case the probability of serving all the queues increases. Therefore, in the limiting case where $K$ becomes very large, MM and MWM result in similar performance.

In Figure 6 we have investigated the effect of service success probability. As we can see in the figures, the only effect of this parameter is to change the stability point (the arrival rate at which queue occupancy tends to infinity). In this case, again we can see that for all $q = 0.2, 0.5, 0.8$ MWM policy performs better than the other policies.
Fig. 4: Average total queue occupancy, $N = 8$, $K = 4$, $q = 0.8$
Fig. 5: Average total queue occupancy, $N = 8, K = 8, q = 0.8$
Fig. 6: Average total queue occupancy, $N = 8$, $K = 6$, $p = 0.5$
VIII. Conclusions

In this paper, we considered the problem of assignment of $K$ identical servers to a set of $N$ parallel queues in a time slotted multi-server queueing system with random connectivities. For such systems, it has been previously shown that MWM is throughput optimal, i.e., has the maximum stability region. In this paper, we first showed that for a symmetric system (refer to Definition I) MWM is also optimal in minimization of a class of cost functions of queue lengths including the average queueing delay. Our approach in the proof was based on stochastic ordering and dynamic coupling techniques. Second, we derived a linear algebraic representation of the stability region for the aforementioned multi-server queueing system. We showed that the stability region form is a convex polytope for which we explicitly determined the coefficients of the facet defining hyperplanes associated to the stability region polytope.

Appendix I

Proof of Lemma 1, Lemma 2 and Lemma 3

A. Proof of Lemma 1

Proof: Let $M^{(\tilde{\pi})}(t)$ denote the employed matching after applying the balancing reallocation. According to the definition of balancing server reallocation, a server reallocation at time slot $t$ results in a queue length vector $\tilde{x}'(t)$ that satisfies either condition $C_1$ or $C_2$. Therefore, we consider the following two cases:

Case 1: Condition $C_1$ is satisfied at time slot $t$. Thus, $\tilde{x}'_n(t) \leq x'_n(t)$ for all $n = 1, 2, ..., N$ and there exists $m \in \{1, 2, ..., N\}$ such that $0 \leq \tilde{x}'_m(t) < x'_m(t)$. Therefore, there exists no queue that was served by $\pi$ but not by $\tilde{\pi}$. Also queue $m$ which was not receiving service by $\pi$ at time slot $t$, is now receiving service after applying the balancing server reallocation. Therefore, $x_m(t - 1) \sum_{k=1}^{K} c_{m,k}(t)M^{(\pi)}_{m,k}(t) = 0$, $x_m(t - 1) \sum_{k=1}^{K} c_{m,k}(t)M^{(\tilde{\pi})}_{m,k}(t) = x_m(t - 1) > 0$ and

$$\sum_{n=1, n \neq m}^{N} x_n(t - 1) \sum_{k=1}^{K} c_{n,k}(t)M^{(\pi)}_{n,k}(t) \leq \sum_{n=1, n \neq m}^{N} x_n(t - 1) \sum_{k=1}^{K} c_{n,k}(t)M^{(\tilde{\pi})}_{n,k}(t).$$

(23)

Therefore, $\text{MW}_{\tilde{\pi}}(t) < \text{MW}_{\pi}(t)$.

Case 2: Condition $C_2$ is satisfied at time $t$. In this case, by using policy $\pi$ at time slot $t$ queue $n$ is receiving service but queue $m$ is not. In contrast, by using policy $\tilde{\pi}$, at time slot $t$ queue $m$ is receiving service but queue $n$ is not. Therefore, we have

$$\text{MW}_{\tilde{\pi}}(t) - \text{MW}_{\pi}(t) = x_m(t - 1) - x_n(t - 1) > 0$$

(24)
Therefore, \( \text{MW}_\pi(t) < \text{MW}_{\bar{\pi}}(t) \). By considering cases 1 and 2 the lemma follows. \( \blacksquare \)

\[ B. \text{ Proof of Lemma 2} \]

\textbf{Proof:} Without loss of generality, we may convert the bipartite graph \( G_t \) to a complete weighted bipartite graph \( G'_t \). This is done by adding some vertices and edges of zero weight as necessary. In particular, if \( N > K \), we will add \( N - K \) servers on the right hand side with edges of weight “0” to each queue (each vertex on the left hand side). It is like the case that we add some servers with service rate 0 to the system. If \( N < K \), we will add \( K - N \) queues on the left hand side with edges of weight “0” to each server (each vertex on the right hand side). It is like the case that we add some queues which are disconnected from all the servers. Thus, we deal with a complete bipartite graph with \( \max\{N,K\} \) vertices on each side. We denote the sets of vertices on each part of \( G'_t \) by \( \mathcal{N}' \) and \( \mathcal{K}' \), respectively and the set of edges by \( \mathcal{E}' \). Consequently, a policy \( \pi \) is defined as \( \pi = \{ M^{(\pi)}(t) \}_{t=1}^\infty \) where \( M^{(\pi)}(t) \) is a perfect matching in the complete bipartite graph \( G'_t \). Note that a maximum weight perfect matching \( M^{(\text{MWM})}(t) \) in the complete bipartite graph \( G'_t \) is the same as the maximum weighted matching in graph \( G_t \) if we remove the added edges of weight “0” from matching \( M^{(\text{MWM})}(t) \).

Consider now a policy \( \pi \) which is employing perfect matching \( M^{(\pi)}(t) \) at time slot \( t \). Suppose that \( M^{(\pi)}(t) \) is not a maximum weight perfect matching on graph \( G'_t \), i.e., \( \text{MW}_\pi(t) < \text{MW}_{\text{MWM}}(t) \). Also, consider a maximum weight perfect matching \( M^{(\text{MWM})}(t) \) at time slot \( t \). Now, consider these two matchings on \( G'_t = (\mathcal{N}', \mathcal{K}', \mathcal{E}') \). Each of \( M^{(\pi)}(t) \) and \( M^{(\text{MWM})}(t) \) corresponds to a distinct sub-graph of \( G'_t \) namely \( G'^{\pi}_t = (\mathcal{N}', \mathcal{K}', \mathcal{E}'^{(\pi)}) \) and \( G'^{(\text{MWM})}_t = (\mathcal{N}', \mathcal{K}', \mathcal{E}'^{(\text{MWM})}) \), respectively.

We now build two directed weighted sub-graphs \( D^{(\pi)}_t \) and \( D^{(\text{MWM})}_t \) as follows: \( D^{(\pi)}_t \) is the same as \( G'^{(\pi)}_t \) with all the edges directed from \( \mathcal{N}' \) to \( \mathcal{K}' \) with the same edge weights as \( G'^{(\pi)}_t \). \( D^{(\text{MWM})}_t \) is the same as \( G'^{(\text{MWM})}_t \) with all the edges directed from \( \mathcal{K}' \) to \( \mathcal{N}' \) with edge weights equal to the negative of edge weights of \( G'^{(\text{MWM})}_t \). Now, consider graph \( U = D^{(\pi)}_t \cup D^{(\text{MWM})}_t \), i.e., the union of the sub-graphs \( D^{(\pi)}_t \) and \( D^{(\text{MWM})}_t \). Graph \( U \) is also the union of some even cycles denoted by \( L \). This is directly concluded from the fact that \( D^{(\pi)}_t \) and \( D^{(\text{MWM})}_t \) are each perfect matchings of \( G'_t \) and thus each vertex is incident to an incoming edge and an outgoing edge. Furthermore, for the weight of \( U \) we have

\[
w(U) = \sum_{\ell \in L} w(\ell) = \text{MW}_\pi(t) - \text{MW}_{\text{MWM}}(t) < 0. \quad (25)
\]
Therefore, there must exist a negative cycle in $U$ namely $\ell^*$. Cycle $\ell^*$ is an even cycle that contains an even number of nodes let say $2W$ nodes ($W$ nodes from $\mathcal{N}'$ and $W$ nodes from $\mathcal{K}'$) and also $2W$ edges. Let us denote the nodes of set $\mathcal{N}'$ and $\mathcal{K}'$ that contribute in $\ell^*$ by $n_1, n_2, ..., n_W$ and $k_1, k_2, ..., k_W$, respectively. Thus, cycle $\ell^*$ can be represented by the sequence of its edges as $\ell^* = e_{n_1,k_1}, e_{k_1,n_2}, e_{n_2,k_2}, ..., e_{n_{W-1},n_W}, e_{n_W,k_W}, e_{k_W,n_1}$ (see Figure 7).

Note that edges $e_{k_1,n_2}, e_{k_2,n_3}, ..., e_{k_W,n_1}$ belong to $D^{(\text{MWM})}_t$ and have negative weights while edges $e_{n_1,k_1}, e_{n_2,k_2}, ..., e_{n_W,k_W}$ belong to $D^{(\pi)}_t$ and have positive weights. We can also represent the edges of $\ell^*$ by pairing the edges incident to each node $n_1, n_2, ..., n_W$ as follows; $\ell^* = (e_{k_W,n_1}, e_{n_1,k_1}), (e_{k_1,n_2}, e_{n_2,k_2}), ..., (e_{k_W-1,n_W}, e_{n_W,k_W})$. We label each pair of edges as follows: $r_{n_1} = (e_{k_W,n_1}, e_{n_1,k_1}), r_{n_2} = (e_{k_1,n_2}, e_{n_2,k_2}), ..., r_{n_W} = (e_{k_W-1,n_W}, e_{n_W,k_W})$. All $r_{n_1}, r_{n_2}, ..., r_{n_W}$ pairs in $\ell^*$ have the following property: The weights associated to the edges of each pair $r_{n_i}$, $1 \leq i \leq W$ is either $(0,0)$, $(0, x_{n_i}(t-1)),(-x_{n_i}(t-1),0)$ or $(-x_{n_i}(t-1), x_{n_i}(t-1))$. Accordingly, we will specify three types of edge pair as follows:

- **Type 0 (T0):** pairs with edge weights $(0,0)$ or $(-x_{n_i}(t-1), x_{n_i}(t-1))$, $1 \leq i \leq W$.
- **Type 1 (T1):** pairs with edge weights $(-x_{n_i}(t-1),0)$, $1 \leq i \leq W$ and $x_{n_i}(t-1) > 0$.
- **Type 2 (T2):** pairs with edge weights $(0, x_{n_i}(t-1))$, $1 \leq i \leq W$ and $x_{n_i}(t-1) > 0$.

Since $\ell^*$ is a negative cycle, there must exist a node $n_j$ with edge pairs $(e_{k_{j-1},n_j}, e_{n_j,k_j}) = (-x_{n_j}(t-1),0)$ (if $j = 1$, $(e_{k_W,n_1}, e_{n_1,k_1}) = (-x_{n_1}(t-1),0)$) and $x_{n_j}(t-1) > 0$. Obviously, the edge pair associated to node $n_j$ is of type T1. We now consider the following three cases:

**Case 1:** All the edge pairs other than $r_{n_j}$ are of type T0. In this case, by replacing the edges of $D^{(\pi)}_t$ by the edges of $D^{(\text{MWM})}_t$ in the negative cycle $\ell^*$ we obtain a server reallocation at time
Fig. 8: Cases 2 and 3 in the proof of Lemma 2

This server reallocation is balancing as in $\pi$ queue $n_j$ was not receiving service while after the server reallocation it is. Therefore, if $\tilde{x}'(t)$ be the queue length vector after server reallocation we have $\tilde{x}'(t) \leq x'(t)$ (satisfying condition C1).

Case 2: If we go backward on the edge pairs of cycle $\ell^*$ from $r_{n_j}$ and the first non-$T0$ pair is a $T1$ pair. Let $r_{n_j'}$ be such a $T1$ edge pair. In other words, in cycle $\ell^*$ the pairs $r_{n_j'}$ and $r_{n_j}$ are of type $T1$ while other pairs between them are of type $T0$ (see Figure 8a). In this case, by replacing the edges of $D_l^{(\pi)}$ by the edges of $D_l^{(MWM)}$ just for nodes $n_{j'+1}$ to $n_j$ of cycle $\ell^*$ and allocating $k_j$ to $n_j'$ we obtain a server reallocation at time slot $t$. This server reallocation is balancing as in $\pi$ queue $n_j$ was not receiving service while after the server reallocation it is and also queue $n_j'$ was not being serviced in $\pi$ but now may get service (depending on the weight of edge $e_{k_j,n_j'}$) and the service of other queues is not disturbed. Therefore, if $\tilde{x}'(t)$ be the queue length vector after server reallocation, we have $\tilde{x}'(t) \leq x'(t)$ and $\tilde{x}'_{n_j'}(t) < x'_{n_j}(t)$ (satisfying condition C1).

Case 3: If we go backward on the edge pairs of cycle $\ell^*$ from $r_{n_j}$ and the first non-$T0$ pair is a $T2$ pair. Let $r_{n_j'}$ be such a $T2$ edge pair. In other words, in cycle $\ell^*$, pair $r_{n_j'}$ is of type $T2$ and $r_{n_j}$ is of type $T1$ while other pairs between them are of type $T0$ (see Figure 8b). First of all, we claim that $x_{n_j'}(t-1) \leq x_{n_j}(t-1)$. If $x_{n_j'}(t-1) > x_{n_j}(t-1)$, by replacing the edges of $D_l^{(MWM)}$ by the edges of $D_l^{(\pi)}$ just for nodes $n_{j'}$ to $n_{j-1}$ of the cycle $\ell^*$ and not serving queue
n_j, without disturbing the service of other queues we obtain a server reallocation at time slot t with larger MW index than MW_{MWM}(t) and this contradicts with the fact that MWM has the maximum MW index at time slot t. Accordingly, we consider the following two sub-cases:

Sub-case 3.1: x_{n_j}(t − 1) > x_{n_j}(t − 1). In this case, we replace the edges of D_t(π) by the edges of D_t^{MWM} just for queues n_j+1 to n_j of the cycle ℓ* and allocate server k_j to queue n_j'. Note that server k_j may or may not serve queue n_j'. We consider the worst case where it does not serve this queue (w_{n_j,k}(t) = 0). Thus, without disturbing the service of other queues we obtain a server reallocation with the following property: all the queues other than n_j and n_j' have the same service as before; queue n_j which was receiving zero service in π, now receives one packet service and queue n_j' which was receiving one packet service, now in the worst case looses its service. Therefore, this server reallocation is balancing as the queue length after server reallocation (denoted by \tilde{x}'(t)) and the queue length after applying policy π (denoted by x'(t)) satisfy condition C2. So, \tilde{x}'(t) ≤ x'(t) and therefore the applied server reallocation is a balancing one.

Sub-case 3.2: x_{n_j}(t − 1) = x_{n_j}(t − 1). In this case, the sequence of edge pairs r_{n_j}, ..., r_{n_j} contribute “0” in the calculation of w(U). Therefore, we may treat them as a sequence of edge pairs of type T0. By doing so, we have a new negative cycle ℓ* which is the same as ℓ* but the sequence of edge pairs r_{n_j}, ..., r_{n_j} are replaces by the same number of edge pairs of type T0. Note that cycle ℓ* is a negative cycle with the same weight as ℓ*. Therefore, there must exist an edge pair of type T1 in ℓ* (In this case, it is not possible for ℓ* to have just edge pairs of type T0 as it results in w(ℓ*) = 0) and all the cases of 1, 2 and 3 applies for ℓ* as well. Note that as long as case 3.2 is true, we always obtain a new negative cycle and the whole process repeats.

Cases 1, 2 and 3 cover all the possible cases of negative cycle ℓ* for all of which we proved that there exists a balancing server reallocation. Putting everything in a nutshell, we proved that if the policy π is not maximum weighted matching at time slot t there exists a negative cycle in the graph obtained by the taking the union of the matchings of π and MWM (i.e., U). We proved that by reallocation of the servers involved in the negative cycle, we always can find a balancing server reallocation for policy π.
C. Proof of Lemma 3

**Proof:** Fix any arbitrary policy \( \pi \in \Pi_t^h \) and any arbitrary sample path

\[
\omega = (x(0), c(1), a(1), x(1), c(2), a(2), x(2), ...)
\]

of the underlying random variables \((X(0), C(1), A(1), X(1), C(2), A(2), X(2), ...)\). We apply the coupling method to construct from \( \omega \) a new sample path

\[
\tilde{\omega} = (\tilde{x}(0), \tilde{c}(1), \tilde{a}(1), \tilde{x}(1), \tilde{c}(2), \tilde{a}(2), \tilde{x}(2), ...)
\]

resulting in a new sequence of random variables \((\tilde{X}(0), \tilde{C}(1), \tilde{A}(1), \tilde{X}(1), \tilde{C}(2), \tilde{A}(2), \tilde{X}(2), ...)\) with \(X(0) = \tilde{X}(0)\). Note that \(X(0)\) is the queue length vector by which the system starts working. We denote the policy driven by the new sample path \(\tilde{\omega}\) by \(\tilde{\pi}\). In fact, we construct \(\tilde{\omega}\) and \(\tilde{\pi} \in \Pi_t^{h-1}\) in such a fashion that for all the sample paths and all time slots we have \(\tilde{x}(t) \preceq_x x(t)\).

Therefore, for any \(f \in \mathcal{F}\) we have \(f(\tilde{x}(t)) \leq f(x(t))\) for all \(t\). Since processes \(\{(C(t), A(t))\}_t^{\infty}\) and \(\{\tilde{C}(t), \tilde{A}(t)\}_t^{\infty}\) are the same in distribution (these processes are permutation invariant), the process \(\{f(\tilde{X}(t))\}\) obtained by applying policy \(\tilde{\pi}\) to the system is stochastically smaller than \(\{f(X(t))\}\), i.e., \(\{f(\tilde{X}(t))\} \leq_{st} \{f(X(t))\}\) and \(\tilde{\pi}\) dominates \(\pi\).

Therefore, in the following, our goal would be to construct \(\tilde{\pi}\) and \(\tilde{\omega}\) such that \(\tilde{x}(t) \preceq_x x(t)\) is satisfied for all time slots. In the proof, we always use the tilde notation for all random variables that belong to the new system. The construction of \(\tilde{\pi}\) is done in two steps:

**Step 1: Construction of \(\tilde{\pi}\) for \(\tau \leq t\):** To construct the new sample path \(\tilde{\omega}\) we let the arrival, connectivity and the policy be the same as the first system until time slot \(t - 1\), i.e., \(\tilde{c}(\tau) = c(\tau)\), \(\tilde{a}(\tau) = a(\tau)\) and \(M^{(\pi)}(\tau) = M^{(\tilde{\pi})}(\tau)\) for \(\tau \leq t - 1\). Thus, the resulting queue lengths at the beginning of time slot \(t\) (or at the end of time slot \(t - 1\)) are equal, i.e., \(\tilde{x}(t - 1) = x(t - 1)\). We now consider the construction of \(\tilde{\omega}\) and \(\tilde{\pi}\) for time slot \(t\). For this time slot, one of the following two cases may apply.

1) The distance of policy \(\pi\) to \(\Pi_t\) is less that \(h\) balancing server reallocations. Therefore \(\pi \in \Pi_t^{h-1}\) and we let \(\tilde{c}(t) = c(t)\) and \(\tilde{a}(t) = a(t)\) and \(M^{(\tilde{\pi})}(t) = M^{(\pi)}(t)\). Therefore, \(\tilde{x}(t)\) and \(x(t)\) are equal in both sample path \(\omega\) and \(\tilde{\omega}\). Thus, \(\tilde{x}(t) \leq x(t)\).

2) The distance of policy \(\pi\) to \(\Pi_t\) is exactly \(h\) balancing server reallocations. Since \(\pi \in \Pi_t^h\) and \(h > 0\), according to Lemma 2 there exists a balancing server reallocation such that either \(C1\) or \(C2\) is satisfied. Thus, we consider two cases:
Case 2.1: After applying the balancing reallocation, condition C1 is satisfied. In other words, there exists a matching such that if applied on the queue length $\bar{x}(t-1) = x(t-1)$ at time slot $t$, we get $\bar{x}'(t)$ such that $\bar{x}'(t) \leq x'(t)$. We call such a matching by $M^{(\bar{\pi})}(t)$. In this case, we let $\tilde{c}(t) = c(t)$ and $\tilde{a}(t) = a(t)$ and we apply $M^{(\bar{\pi})}(t)$ at time slot $t$, i.e., arrivals and connectivities are the same in both systems and policy $\bar{\pi}$ acts at time slot $t$. So, we can easily check that $\bar{x}(t) \leq x(t)$ and therefore $\bar{x}(t) \leq x(t)$.

Case 2.2: After applying the balancing reallocation, condition C2 is satisfied. In other words, there exists a matching such that if applied on the queue length $\bar{x}'(t)$ which is different from $x'(t)$ in two elements $m$ and $n$ such that $x'_m(t) < x'_n(t) \leq \bar{x}_m(t) < x'_m(t)$ and the following constraints are satisfied: $\bar{x}_n(t) = x'_n(t) + 1$ and $\bar{x}_m(t) = x'_m(t) - 1$. We call such a matching by $M^{(\bar{\pi})}(t)$. In this case, we let $\tilde{c}(t) = c(t)$ and $\tilde{a}(t) = a(t)$ and we apply $M^{(\bar{\pi})}(t)$ at time slot $t$, i.e., arrivals and connectivities are the same in both systems and policy $\bar{\pi}$ acts at time slot $t$. We consider all the following conditions for arrivals to queues $m$ and $n$ as follows:

- If there is no arrival or there is an arrival to both queues $m$ and $n$ (i.e., $a_m(t) = a_n(t) = 0$ or $a_m(t) = a_n(t) = 1$), we conclude that $\bar{x}(t)$ and $x(t)$ satisfy condition D3. Thus, $\bar{x}(t) \leq x(t)$.
- If there is an arrival to queue $m$ but not $n$ (i.e., $a_m(t) = 1$, $a_n(t) = 0$), we conclude that $\bar{x}(t)$ and $x(t)$ satisfy condition D3. Thus, $\bar{x}(t) \leq x(t)$.
- If there is an arrival to queue $n$ but not $m$ (i.e., $a_m(t) = 0$, $a_n(t) = 1$) and $\bar{x}_m(t) = \bar{x}_n(t)$, we conclude that $\bar{x}(t)$ and $x(t)$ satisfy condition D2. Thus, $\bar{x}(t) \leq x(t)$.
- If there is an arrival to queue $n$ but not $m$ (i.e., $a_m(t) = 0$, $a_n(t) = 1$) and $\bar{x}_n(t) < \bar{x}_m(t)$, we conclude that $\bar{x}(t)$ and $x(t)$ satisfy condition D3. Thus, $\bar{x}(t) \leq x(t)$.

In all the cases we can see that $\bar{x}(t) \leq x(t)$.

Note that the obtained policy $\bar{\pi}$ belongs to $\Pi^{h-1}_{t}$ as we applied a balancing server reallocation to the matching employed in $\pi$ at time slot $t$.

Step 2: Construction of $\bar{\pi}$ for $\tau > t$: In this step, we focus on construction of $\tilde{\omega}$ and $\bar{\pi}$ for $\tau > t$. We will employ mathematical induction to achieve this goal. In particular, we assume that $\tilde{\omega}$ and $\bar{\pi}$ are constructed up to time slot $\tau$ ($\tau \geq t$) such that $\bar{x}(\tau) \leq x(\tau)$ i.e., one of the conditions D1, D2 and D3 is satisfied for $x(\tau)$ and $\bar{x}(\tau)$. We will prove that policy $\bar{\pi}$ and sample
path \( \tilde{\omega} \) can be constructed such that \( \tilde{x}(\tau + 1) \leq x(\tau + 1) \) (i.e., one of the conditions \( D1 \), \( D2 \) and \( D3 \) is satisfied for \( x(\tau + 1) \) and \( \tilde{x}(\tau + 1) \)). Accordingly, we consider three cases corresponding to each condition \( D1 \), \( D2 \) or \( D3 \) at time slot \( \tau \):

1. Case 1: \( \tilde{x}(\tau) \leq x(\tau) \). In this case, the construction of \( \tilde{\omega} \) and \( \tilde{\pi} \) at time slot \( \tau + 1 \) is straightforward. We let \( \tilde{c}(\tau + 1) = c(\tau + 1) \) and \( \tilde{a}(\tau + 1) = a(\tau + 1) \) and \( M(\tilde{\pi})(\tau + 1) = M(\pi)(\tau + 1) \). Thus, \( \tilde{x}(\tau + 1) \leq x(\tau + 1) \) and therefore \( \tilde{x}(\tau + 1) \leq x(\tau + 1) \).

2. Case 2: \( \tilde{x}(\tau) \) is obtained from \( x(\tau) \) by permutation of two distinct elements \( m \) and \( n \). In this case, we let \( \tilde{c}_{m,k}(\tau + 1) = c_{m,k}(\tau + 1) \) and \( \tilde{c}_{m,k}(\tau + 1) = c_{n,k}(\tau + 1) \) for \( k = 1, 2, \ldots, K \); \( \tilde{a}_{m}(\tau + 1) = a_{n}(\tau + 1) \) and \( \tilde{a}_{m}(\tau + 1) = a_{n}(\tau + 1) \) for \( k = 1, 2, \ldots, K \). Suppose that \( M(\pi)(\tau + 1) = (M_{n,k}(\tau + 1) \rangle \forall n \in \mathcal{N}, k \in \mathcal{K} \) be the applied matching by policy \( \pi \) at time slot \( \tau + 1 \). We construct \( M(\tilde{\pi})(\tau + 1) \) as follows: Let \( M_{i,k}(\tau + 1) = M_{i,k}(\tau + 1) \rangle \) for \( i \in \mathcal{N}, i \neq n, m \) and also let \( M_{m,k}(\tau + 1) = M_{m,k}(\tau + 1) \rangle \) and \( M_{m,k}(\tau + 1) = M_{m,k}(\tau + 1) \rangle \) for all \( i \in \mathcal{N}, i \neq n, m \). As a result, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition \( D2 \) at time slot \( \tau + 1 \) and therefore \( \tilde{x}(\tau + 1) \leq x(\tau + 1) \).

3. Case 3: \( \tilde{x}(\tau) \) is obtained from \( x(\tau) \) by performing a balancing interchange of two distinct elements \( m \) and \( n \) as defined in condition \( D3 \). In particular, \( \tilde{x}(\tau) \) and \( x(\tau) \) are different in only two elements \( n \) and \( m \) such that \( x_n(\tau) < \tilde{x}_n(\tau) \leq \tilde{x}_m(\tau) < x_m(\tau) \) and the following constraints are satisfied: \( \tilde{x}_n(\tau) = x_n(\tau) + 1 \) and \( \tilde{x}_m(\tau) = x_m(\tau) - 1 \). In this case, we consider the following sub-cases:

   **Sub-case 3.1:** \( \tilde{x}_n(\tau) < \tilde{x}_m(\tau) - 1 \): In this case, we let \( \tilde{c}(\tau + 1) = c(\tau + 1) \) and \( \tilde{a}(\tau + 1) = a(\tau + 1) \) and we let \( M(\tilde{\pi})(\tau + 1) = M(\pi)(\tau + 1) \rangle \). Thus, if \( x_n(\tau) = 0 \) and queue \( n \) is serviced, condition \( D1 \) is satisfied at \( \tau + 1 \). Otherwise, \( \tilde{x}(\tau + 1) \) is obtained from \( x(\tau + 1) \) by performing a balancing interchange of elements \( m \) and \( n \). Therefore \( \tilde{x}(\tau + 1) \leq x(\tau + 1) \).

   **Sub-case 3.2:** \( \tilde{x}_n(\tau) = \tilde{x}_m(\tau) - 1 \): In this case again we let \( \tilde{c}(\tau + 1) = c(\tau + 1) \) and \( \tilde{a}(\tau + 1) = a(\tau + 1) \) and we let \( M(\tilde{\pi})(\tau + 1) = M(\pi)(\tau + 1) \rangle \). Thus, if \( x_n(\tau) = 0 \) and queue \( n \) is serviced, condition \( D1 \) is satisfied at \( \tau + 1 \). If queue \( m \) gets service, queue \( n \) does not get service, there is an arrival to queue \( n \) and no arrival to queue \( m \), then \( \tilde{x}_n(\tau + 1) = x_m(\tau + 1) \) and \( \tilde{x}_m(\tau + 1) = x_n(\tau + 1) \). Therefore, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition \( D2 \) and \( \tilde{x}(\tau + 1) \leq x(\tau + 1) \). Otherwise, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition \( D3 \) and \( \tilde{x}(\tau + 1) \leq x(\tau + 1) \).

   **Sub-case 3.3:** \( \tilde{x}_n(\tau) = \tilde{x}_m(\tau) \): In this case, we let \( \tilde{c}(\tau + 1) = c(\tau + 1) \) and \( M(\tilde{\pi})(\tau + 1) = M(\pi)(\tau + 1) \rangle \). Now, we consider the following cases to determine the arrivals at time slot \( \tau + 1 \).
If \( x_n(\tau) > 0 \) and both queues \( m \) and \( n \) or none of them get service at time slot \( \tau + 1 \), we let \( \tilde{a}(\tau + 1) = a(\tau + 1) \). Therefore, if \( a_m(\tau + 1) = 0 \) and \( a_n(\tau + 1) = 1 \), \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D2 and thus \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \). Otherwise, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D3 and thus \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

If \( x_n(\tau) > 0 \) and queue \( n \) gets service at time slot \( \tau + 1 \) and queue \( m \) does not get service at time slot \( \tau + 1 \), we let \( \tilde{a}(\tau + 1) = a(\tau + 1) \). Therefore, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D3 and thus \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

If \( x_n(\tau) > 0 \) and queue \( m \) gets service at time slot \( \tau + 1 \) and queue \( n \) does not get service at time slot \( \tau + 1 \), we let \( \tilde{a}_m(\tau + 1) = a_m(\tau + 1) \) and \( \tilde{a}_n(\tau + 1) = a_m(\tau + 1) \) and \( \tilde{a}_i(\tau + 1) = a_i(\tau + 1) \) for \( i \in \mathcal{N} \) and \( i \neq m,n \). Therefore, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D2 and thus \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

If \( x_n(\tau) = 0 \) and queue \( n \) gets service at time slot \( \tau + 1 \) (although it does not have any packet to be served), we let \( \tilde{a}(\tau + 1) = a(\tau + 1) \). Therefore, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D1 and thus \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

If \( x_n(\tau) = 0 \) and queue \( m \) gets service at time slot \( \tau + 1 \) and queue \( n \) does not get service at time slot \( \tau + 1 \), we let \( \tilde{a}_m(\tau + 1) = a_m(\tau + 1) \) and \( \tilde{a}_n(\tau + 1) = a_m(\tau + 1) \) and \( \tilde{a}_i(\tau + 1) = a_i(\tau + 1) \) for \( i \in \mathcal{N} \) and \( i \neq m,n \). Therefore, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D2 and thus \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

If \( x_n(\tau) = 0 \) and neither queue \( m \) nor \( n \) gets service at time slot \( \tau + 1 \), we let \( \tilde{a}(\tau + 1) = a(\tau + 1) \). If \( a_m(\tau + 1) = 0 \) and \( a_n(\tau + 1) = 1 \), \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D2 and thus \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \). Otherwise, \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D3 and thus \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

Note that the above cases cover all the possible cases for all of which we constructed \( \tilde{\omega} \) and \( \tilde{\pi} \) such that \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

According to steps 1 and 2, from any sample path \( \omega \) and any arbitrary policy \( \pi \in \Pi^h_t \) we can construct a sample path \( \tilde{\omega} \) and a policy \( \tilde{\pi} \in \Pi^{h-1}_t \) such that at all time slots we have \( \tilde{x}(t) \preceq_p x(t) \). Therefore, \( f(\tilde{x}(t)) \preceq f(x(t)) \). Consequently, the process \( \{ f(\tilde{X}(t)) \} \) obtained by applying policy \( \tilde{\pi} \) to the system is stochastically smaller than \( \{ f(X(t)) \} \), i.e., \( \{ f(\tilde{X}(t)) \} \preceq_{st} \{ f(X(t)) \} \) and therefore \( \tilde{\pi} \in \Pi^{h-1}_t \) dominates \( \pi \in \Pi^h_t \).
D. Proof of Lemma \ref{lemma:system_with_random_service_failures} for the System with Random Service Failures

Proof: The only difference of the proof in this case from the proof of Lemma \ref{lemma:system_with_random_service_failures} is that we have to consider random variables $Q_{n,k}(t) \forall n \in \mathcal{N}, \forall k \in \mathcal{K}$ in our dynamic coupling argument. Therefore, the arbitrary sample path has the form

$$\omega = (x(0), c(1), q(1), a(1), x(1), c(2), q(2), a(2), x(2), ... )$$

and we apply the coupling method to construct from $\omega$ a new sample path

$$\tilde{\omega} = (\tilde{x}(0), \tilde{c}(1), \tilde{q}(1), \tilde{a}(1), \tilde{x}(1), \tilde{c}(2), \tilde{q}(2), \tilde{a}(2), \tilde{x}(2), ... )$$

resulting in a new sequence of random variables

$$(\tilde{X}(0), \tilde{C}(1), \tilde{Q}(1), \tilde{A}(1), \tilde{X}(1), \tilde{C}(2), \tilde{Q}(2), \tilde{A}(2), \tilde{X}(2), ... )$$

with $X(0) = \tilde{X}(0)$. We will follow the same approach and consider the same cases. Before we proceed to the details we introduce the following complementary notation. Suppose that $s_n^{(\pi)}(t)$ denotes the index of the server assigned to queue $n$ at time slot $t$ by policy $\pi$. Note that $s_n^{(\pi)}(t) \in \mathcal{K}$ or it is empty.

Step 1: Construction of $\tilde{\pi}$ for $\tau \leq t$: In this case, as in the proof of Lemma \ref{lemma:system_with_random_service_failures} we first consider $\tau \leq t - 1$. For those slots, we let all $\tilde{c}(\tau), \tilde{q}(\tau), \tilde{a}(\tau)$ and $M^{(\tilde{\pi})}(\tau)$ to be the same in both systems i.e., $\tilde{c}(\tau) = c(\tau), \tilde{q}(\tau) = q(\tau), \tilde{a}(\tau) = a(\tau)$ and $M^{(\tilde{\pi})}(\tau) = M^{(\pi)}(\tau)$ and therefore we have $\tilde{x}(t - 1) = x(t - 1)$.

For time slot $t$, we again consider the following cases.

1) The distance of policy $\pi$ to $\Pi_t$ is less that $h$ balancing server reallocations. In this case, we act the same as what we did in the proof of Lemma \ref{lemma:system_with_random_service_failures} and also let $\tilde{q}(t) = q(t)$. Thus, $\tilde{x}(t) \leq x(t)$.

2) The distance of policy $\pi$ to $\Pi_t$ is exactly $h$ balancing server reallocations. Since $\pi \in \Pi_t^h$ and $h > 0$, according to Lemma \ref{lemma:system_with_random_service_failures} there exists a balancing server reallocation such that either C1 or C2 is satisfied. Thus, we consider two cases.

Case 2.1: After applying the balancing reallocation, condition C1 is satisfied. In other words, there exists a matching $M^{(\tilde{\pi})}(t)$ such that if applied on the queue lengths $\tilde{x}(t - 1) = x(t - 1)$, we get $\tilde{x}'(t)$ such that $\tilde{x}'(t) \leq x'(t)$. Note that if a non-empty queue is serviced under policy $M^{(\pi)}(t)$, it should also get service under policy $\tilde{\pi}$. Otherwise, the condition $\tilde{x}'(t) \leq x'(t)$
will be violated. In this case, we let \( \tilde{c}(t) = c(t) \) and \( \tilde{a}(t) = a(t) \) and we apply \( M(\tilde{\pi})(t) \) at time slot \( t \). For any non-empty queue \( n \) that is serviced under both policies \( \pi \) and \( \tilde{\pi} \) we let \( \tilde{q}_{n,s_n^{(\pi)}}(t) = q_{n,s_n^{(\pi)}}(t) \) and \( \tilde{q}_{n,s_n^{(\pi)}}(t) = q_{n,s_n^{(\pi)}}(t) \). In other words, we let each non-empty queue \( n \) which was being serviced in both systems experience the same service failure. For other variables \( \tilde{q}_{n,k}(t) \) we let \( \tilde{q}_{n,k}(t) = q_{n,k}(t) \). Then, we can easily check that \( \tilde{x}(t) \leq x(t) \) and therefore \( \tilde{x}(t) \leq x(t) \).

**Case 2.2:** After applying the balancing reallocation, condition \( C2 \) is satisfied. In other words, there exists a matching such that if applied on the system at time slot \( t \), we get \( \tilde{x}'(t) \) which is different from \( x'(t) \) in two elements \( m \) and \( n \) such that \( x'_n(t) < \tilde{x}'_n(t) \leq \tilde{x}'_m(t) < x'_m(t) \) and the following constraints are satisfied; \( \tilde{x}'_n(t) = x'_n(t) + 1 \) and \( \tilde{x}'_m(t) = x'_m(t) - 1 \). We call such a matching by \( M(\tilde{\pi})(t) \). Note that in this case, each queue (other than \( m \) and \( n \)) which is (is not) receiving service under \( \pi \) is (is not) also receiving service under \( \tilde{\pi} \). Queue \( n \) is receiving service under \( \pi \) and queue \( m \) is not. Queue \( n \) is not receiving service under \( \tilde{\pi} \) and queue \( m \) is. In this case, we let \( \tilde{c}(t) = c(t) \) and \( \tilde{a}(t) = a(t) \) and we apply \( M(\tilde{\pi})(t) \) at time slot \( t \). We let \( \tilde{q}_{i,s_i^{(\pi)}}(t) = q_{i,s_i^{(\pi)}}(t) \) and \( \tilde{q}_{i,s_i^{(\pi)}}(t) = q_{i,s_i^{(\pi)}}(t) \) for any queue \( i \) which is receiving service under both matchings \( M(\pi)(t) \) and \( M(\tilde{\pi})(t) \). We also let \( \tilde{q}_{m,s_m^{(\pi)}}(t) = q_{m,s_m^{(\pi)}}(t) \), \( \tilde{q}_{n,s_n^{(\pi)}}(t) = q_{n,s_n^{(\pi)}}(t) \) and for other failure variables we let \( \tilde{q}_{n,k}(t) = q_{n,k}(t) \). By such coupling of the service success/failure random variables we will consider the following cases for arrivals and service failures:

- If \( q_{n,s_n^{(\pi)}}(t) = 0 \), we let \( \tilde{a}(t) = a(t) \) and therefore, \( \tilde{x}(t) = x(t) \) which implies \( \tilde{x}(t) \leq x(t) \).
- If \( q_{n,s_n^{(\pi)}}(t) = 1 \), then we can use the same coupling argument on the arrival processes as what we did in the proof of Lemma [3] and conclude that \( \tilde{x}(t) \leq x(t) \). We omit the argument to avoid redundant discussion.

Note that the obtained policy \( \tilde{\pi} \) belongs to \( \Pi^{h-1}_t \) as we applied a balancing server reallocation to the matching employed in \( \pi \) at time slot \( t \).

**Step 2: Construction of \( \tilde{\pi} \) for \( \tau > t \):** In this step, the same as the proof of Lemma [5] we use mathematical induction to construct policy \( \tilde{\pi} \) for \( \tau > t \). Therefore, suppose that \( \tilde{\omega} \) and \( \tilde{\pi} \) are constructed up to time slot \( \tau \) (\( \tau \geq t \)) such that \( \tilde{x}(\tau) \leq x(\tau) \). Therefore, one of the conditions \( D1, D2 \) and \( D3 \) is satisfied for \( x(\tau) \) and \( \tilde{x}(\tau) \) as follows.

**Case 1:** \( \tilde{x}(\tau) \leq x(\tau) \). In this case, the construction of \( \tilde{\omega} \) and \( \tilde{\pi} \) at time slot \( \tau + 1 \) is
straightforward. We let \( \tilde{c}(\tau + 1) = c(\tau + 1), \tilde{q}(\tau + 1) = q(\tau + 1), \tilde{a}(\tau + 1) = a(\tau + 1) \) and policy \( M^{(\tilde{x})}(\tau + 1) = M^{(\pi)}(\tau + 1) \). Thus, \( \tilde{x}(\tau + 1) \leq x(\tau + 1) \) and therefore \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

**Case 2:** \( \tilde{x}(\tau) \) is obtained from \( x(\tau) \) by permutation of two distinct elements \( m \) and \( n \). In this case, we couple the random variables \( c(\tau + 1), a(\tau + 1) \) and construct matching \( M^{(\tilde{x})}(\tau + 1) \) the same as what we did in case 2 of step 2 in the proof of Lemma 3. In addition to these settings, we let \( \tilde{q}_{m,k}(\tau + 1) = q_{n,k}(\tau + 1) \) and \( \tilde{q}_{n,k}(\tau + 1) = q_{m,k}(\tau + 1) \) for \( k = 1, 2, \ldots, K \) and also \( \tilde{q}_{i,k}(\tau + 1) = q_{i,k}(\tau + 1) \) for all \( i \in \mathcal{N}, i \neq n, m \) and \( k = 1, 2, \ldots, K \). By doing such a coupling, we conclude that \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy condition D2 at time slot \( \tau + 1 \) and therefore \( \tilde{x}(\tau + 1) \preceq x(\tau + 1) \).

**Case 3:** \( \tilde{x}(\tau) \) is obtained from \( x(\tau) \) by performing a balancing interchange of two distinct elements \( m \) and \( n \) as defined in condition D3. In particular, \( \tilde{x}(\tau) \) and \( x(\tau) \) are different in only two elements \( n \) and \( m \) such that \( x_n(\tau) < \tilde{x}_n(\tau) \leq \tilde{x}_m(\tau) < x_m(\tau) \) and the following constraints are satisfied; \( \tilde{x}_n(\tau) = x_n(\tau) + 1 \) and \( \tilde{x}_m(\tau) = x_m(\tau) - 1 \). In this case, with the same argument as we did in the proof of Lemma 3, we can check that \( \tilde{x}(\tau + 1) \) and \( x(\tau + 1) \) satisfy one of the conditions D1-D3. We omit the details to avoid redundant discussion.

### APPENDIX II

**Proof of Lemma 4, Lemma 5, Theorem 2 and Theorem 3**

**A. Proof of Lemma 4**

**Proof:** If we write equation (16) for each queue \( n \) and for \( \tau = 1, 2, \ldots, t \) and then adding them up, we will have

\[
X_n(t) = X_n(0) - \sum_{\tau=1}^{t} \sum_{k=1}^{K} H_{n,k}(\tau) + \sum_{\tau=1}^{t} A_n(\tau). \tag{26}
\]

By multiplying the vector \( \alpha \) to the queue length vector \( X(t) \) we have

\[
\alpha X^T(t) = \alpha X^T(0) - \sum_{\tau=1}^{t} \alpha H(\tau) \mathbb{1}_K^T + \sum_{\tau=1}^{t} \alpha A^T(\tau). \tag{27}
\]

Taking the expectation from both sides, dividing by \( t \) and then taking the limit as \( t \) goes to infinity, we will have the following.

\[
\lim_{t \to \infty} \frac{1}{t} \alpha E[X(t)]^T = \lim_{t \to \infty} \frac{1}{t} \alpha E[X(0)]^T - \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \alpha E[H(\tau)] \mathbb{1}_K^T + \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \alpha E[A(\tau)]^T \tag{28}
\]
According to Lemma 3.3 in [4] and the assumption that \( E[X(0)] < \infty \), the left hand side term and the first term in the right hand side term are equal to zero. Using the fact that 
\[
\lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \alpha E[A(\tau)]^\top = \alpha \lambda^T
\]
the lemma follows.

\[ \square \]

**B. Proof of Lemma 5**

**Proof:** Recall that \( M^{(\pi)}(t) \) denotes the employed matching by policy \( \pi \) at time slot \( t \). In (17), by conditioning the process \( H(t) \) on the connectivity process, we will have
\[
\alpha \lambda^T = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \alpha E[H(\tau)] \mathbf{1}_K^T = \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{c \in \mathcal{C}} P(c) \alpha E[H(\tau) \mid C(\tau) = c] \mathbf{1}_K^T
\]

For an arbitrary vector \( v \) of size \( |v| \), \((v)^+\) is defined as a vector of the same size whose \( i \)'th element, \((v)_i^+\), is
\[
(v)_i^+ = \begin{cases} 
0 & \text{if } v_i < 0 \\
v_i & \text{if } v_i \geq 0 
\end{cases}
\]

Therefore, (29) can be bounded by
\[
\alpha \lambda^T \leq \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{c \in \mathcal{C}} P(c) (\alpha)^+ E \left[ c \otimes M^{(\pi)}(\tau) \right] \mathbf{1}_K^T
\]
\[
\leq \lim_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{c \in \mathcal{C}} P(c) \max_{M \in \mathcal{M}} \left( (\alpha)^+ (c \otimes M) \mathbf{1}_K^T \right)
\]
\[
= \sum_{c \in \mathcal{C}} P(c) \max_{M \in \mathcal{M}} \left( (\alpha)^+ (c \otimes M) \mathbf{1}_K^T \right). \tag{31}
\]

Now, consider two different cases.

- \( \alpha \in \mathbb{R}^N_+ \): The result follows directly from (31) since \((\alpha)^+ = \alpha\).

- \( \alpha \not\in \mathbb{R}^N_+ \): In this case, from (29) we have the following inequality.
\[
\alpha \lambda^T \leq \sum_{c \in \mathcal{C}} P(c) \max_{M \in \mathcal{M}} \left( (\alpha)^+ (c \otimes M) \mathbf{1}_K^T \right) \tag{32}
\]

However, \((\alpha)^+ \in \mathbb{R}_+^N\) and according to the previous case we also have
\[
(\alpha)^+ \lambda^T \leq \sum_{c \in \mathcal{C}} P(c) \max_{M \in \mathcal{M}} \left( (\alpha)^+ (c \otimes M) \mathbf{1}_K^T \right). \tag{33}
\]

Noting (33) and the fact that \( \lambda \in \mathbb{R}^N_+ \), we conclude that (32) is a redundant inequality.

\[ \square \]
C. Proof of Theorem 2

Proof: Let \( \mathcal{M}^\alpha = \{ M^\alpha(c), c \in C \} \) denote a set of matchings that solve (19). \( \mathcal{M}^\alpha \) is not a unique solution of (19) and there may be more than one set of matchings \( \mathcal{M}^\alpha \) whose elements maximize (19). Let \( \tilde{\mathcal{M}}^\alpha = \{ M^\alpha_i, 1 \leq i \leq |\tilde{\mathcal{M}}^\alpha| \} \) denote the set of all distinguished solutions of (19). Obviously \( |\tilde{\mathcal{M}}^\alpha| < \infty \), since set \( \mathcal{M} \) is finite. Note that each solution \( \mathcal{M}^\alpha_i = \{ M^\alpha_i(c), c \in C \}, 1 \leq i \leq |\tilde{\mathcal{M}}^\alpha| \) is corresponding to a deterministic policy with the average service rate vector

\[
R^\alpha_i = \left( \sum_{c \in C} P(c) (c \otimes M^\alpha_i(c)) \mathbf{1}_K^T \right)^T
\]

and therefore, according to (14) each solution of (19) is associated with a vertex of the polytope \( \mathcal{R} \). Since there are \( |\tilde{\mathcal{M}}^\alpha| \) distinguished solutions of (19), \( |\tilde{\mathcal{M}}^\alpha| \) vertices of polytope \( \mathcal{R} \) are on the hyperplane associated to (18). In other words, each face of \( \mathcal{R} \) associated to (18) is represented by \( F_\alpha = \text{conv.hull}_{1 \leq i \leq |\tilde{\mathcal{M}}^\alpha|} R^\alpha_i \).

We now introduce the following mapping.

\[
\Omega : \mathbb{R}_+^N \mapsto \{0, 1\}^N
\]

\[
\alpha = (\alpha_1, \alpha_2, ..., \alpha_N) \mapsto \beta = (\beta_1, \beta_2, ..., \beta_N) : \beta_n = \begin{cases} 
0 & \text{if } \alpha_n = 0 \\
1 & \text{if } \alpha_n > 0 
\end{cases} \forall n \in \mathcal{N} \quad (35)
\]

Mapping \( \Omega \) takes any vector \( \alpha \in \mathbb{R}_+^N \) and returns another \( N \) dimensional vector \( \beta \) whose elements are in \( \{0, 1\} \). For each element of \( \beta \) namely \( \beta_n \), \( \Omega \) returns “0” if \( \alpha_n \) is zero and returns “1” if \( \alpha_n \) is non-zero. In the following, our objective is to prove that \( \tilde{\mathcal{M}}^\alpha \subseteq \mathcal{M}^{\Omega(\alpha)} \) (or equivalently \( F_\alpha \subseteq F^{\Omega(\alpha)} \)). To this end, we have to show that \( \mathcal{M}^\alpha_i \subseteq \mathcal{M}^{\Omega(\alpha)} \) for all \( 1 \leq i \leq |\tilde{\mathcal{M}}^\alpha| \). Remember that \( \mathcal{M}^\alpha_i = \{ M^\alpha_i(c), c \in C \} \) and \( M^\alpha(c) = \arg \max_{M \in \mathcal{M}} \alpha(c \otimes M) \mathbf{1}_K^T \).

We claim that any matching \( M^\alpha(c) \) that is maximizing \( \alpha(c \otimes M) \mathbf{1}_K^T \) also maximizes \( \Omega(\alpha)(c \otimes M) \mathbf{1}_K^T \). Note that if a matching \( M^\alpha(c) \) is maximizing \( \alpha(c \otimes M) \mathbf{1}_K^T \), this means that \( M^\alpha(c) \) is a maximum matching in a bipartite graph \( G^\alpha = (\mathcal{N}, \mathcal{K}, \mathcal{E}^\alpha) \) (see Figure 9). In \( G^\alpha \), \( \mathcal{N} \) and \( \mathcal{K} \) are the two sets of vertices in each part of the graph and \( \mathcal{E}^\alpha = \{ e^\alpha_{n,k}, \forall n \in \mathcal{N}, \forall k \in \mathcal{K} \} \) is the set of edges between these two parts where \( e^\alpha_{n,k} = \alpha_n c_{n,k} \). If there exists \( M^\alpha(c) \) who maximizes \( \alpha(c \otimes M) \mathbf{1}_K^T \) but not \( \Omega(\alpha)(c \otimes M) \mathbf{1}_K^T \), this means that matching \( M^\alpha(c) \) is a maximum matching for graph \( G^\alpha \) but not for \( G^{\Omega(\alpha)} \). If we treat vector \( \Omega(\alpha) \) as \( x'(t) \) in definition 3 (i.e., suppose that \( x'(t) = \Omega(\alpha) \) in definition 3) and graph \( G^{\Omega(\alpha)} \) as graph \( G_t \), then according to Lemma 2 there must exist a server reallocation for matching \( M^\alpha(c) \) in graph \( G^{\Omega(\alpha)} \) that results into another
matching $M'$ whose matching weight is greater than that of $M^\alpha(c)$. Since $\Omega(\alpha) \in \{0, 1\}^N$ it is not possible for this server reallocation to satisfy condition C2. Thus, it must satisfy condition C1. This means that by applying the server reallocation (i.e., matching $M'$) we will get into $\tilde{x}'(t)$ such that $\tilde{x}'_n(t) \leq x'_n(t)$ for all $n = 1, 2, ..., N$ and there exists at least one $m \in \{1, 2, ..., N\}$ such that $\tilde{x}'_m(t) < x'_m(t)$. This means that there is at least one queue $m$ which is not receiving service under $M^\alpha(c)$ but it is serviced under $M'$ while the service of other queues is not disturbed. If we apply matching $M'$ to graph $G^\alpha$ we have the same results, i.e., there is at least one queue $m$ which is not receiving service under $M^\alpha(c)$ but it is serviced under $M'$ while the service of other queues is not disturbed. This means that matching weight of $M^\alpha(c)$ is less than that of matching $M'$, i.e., matching $M^\alpha(c)$ is not maximum matching in graph $G^\alpha$. This contradicts with our assumption that $M^\alpha(c)$ is maximum matching in graph $G^\alpha$.

Therefore, for all $c \in C$ all the maximum matchings $M^\alpha(c)$ in graph $G^\alpha$ are also maximum matchings in graph $G^{\Omega(\alpha)}$. In other words, all the solutions of “$\arg\max \alpha(c \oplus M^{\downarrow K}_T)$” are solutions for “$\arg\max \Omega(\alpha)(c \oplus M^{\downarrow K}_T)$”. Hence, $M^\alpha \subseteq M^{\Omega(\alpha)}$. Consequently $F_\alpha \subseteq F^{\Omega(\alpha)}$.

Thus, according to Lemma 5 and the fact that $F_\alpha \subseteq F^{\Omega(\alpha)}$, we conclude that all the hyperplanes associated to half spaces (18) with $\alpha \in \mathbb{R}_+^N - \{0, 1\}^N$ are redundant in describing polytope $R$. 

![Graph $G^\alpha$](image-url)
Note that vector $\alpha = 0_N$ will result in the obvious equality $0 = 0$ and therefore is redundant too. Consequently, from all the vectors $\alpha \in \mathbb{R}_+^N$, just $\alpha \in \{0, 1\}^N - \{0_N\}$ are sufficient to describe polytope $\mathcal{R}$ and the theorem follows.

D. Proof of Theorem 3

Proof: We start with the Lyapunov function evaluation. we use the quadratic function $L(X) = \sum_{n=1}^N X_n^2(t)$ as our Lyapunov function. The Lyapunov drift $D$ for two successive time slots has the following form.

$$D(t+1) = E[L(X(t+1)) - L(X(t)) \mid X(t)]$$

$$= E \left[ \sum_{n=1}^N X_n^2(t+1) - X_n^2(t) \mid X(t) \right]$$

$$= E \left[ \sum_{n=1}^N (X_n(t+1) - X_n(t))^2 \mid X(t) \right]$$

$$+ 2E \left[ \sum_{n=1}^N X_n(t)(X_n(t+1) - X_n(t)) \mid X(t) \right]$$

(36)

For the first term, we have

$$E \left[ \sum_{n=1}^N (X_n(t+1) - X_n(t))^2 \mid X(t) \right]$$

$$= E \left[ \sum_{n=1}^N (A_n(t+1) - \sum_{k=1}^K H_{n,k}(t+1))^2 \mid X(t) \right]$$

$$= E \left[ \sum_{n=1}^N A_n^2(t+1) \mid X(t) \right]$$

$$- 2E \left[ \sum_{n=1}^N \sum_{k=1}^K A_n(t+1)H_{n,k}(t+1) \mid X(t) \right]$$

$$+ E \left[ \sum_{n=1}^N \left( \sum_{k=1}^K H_{n,k}(t+1) \right)^2 \mid X(t) \right]$$

$$\leq N A_{max}^2 + \sum_{n=1}^N E \left[ \left( \sum_{k=1}^K H_{n,k}(t+1) \right)^2 \mid X(t) \right].$$

(37)
For the second term in (36), we have
\[
E \left[ \sum_{n=1}^{N} X_n(t)(X_n(t + 1) - X_n(t)) \mid X(t) \right]
\]
\[
= E \left[ \sum_{n=1}^{N} X_n(t) \left( A_n(t + 1) - \sum_{k=1}^{K} H_{n,k}(t + 1) \right) \mid X(t) \right]
\]
\[
= \sum_{n=1}^{N} X_n(t)E[A_n(t + 1)] - \sum_{n=1}^{N} E \left[ X_n(t) \sum_{k=1}^{K} H_{n,k}(t + 1) \mid X(t) \right].
\] (38)

Therefore, the Lyapunov drift \( D(t + 1) \) can be written as
\[
D(t + 1) \leq NA_{max}^2 + \sum_{n=1}^{N} E \left[ \left( \sum_{k=1}^{K} H_{n,k}(t + 1) \right)^2 \mid X(t) \right]
\]
\[
+ 2 \sum_{n=1}^{N} X_n(t)E[A_n(t + 1)] - 2 \sum_{n=1}^{N} E \left[ X_n(t) \sum_{k=1}^{K} H_{n,k}(t + 1) \mid X(t) \right].
\] (39)

In the following, we show that the Lyapunov drift in (39) is bounded as follows.
\[
D(t + 1) \leq NA_{max}^2 + \sum_{n=1}^{N} E \left[ \left( \sum_{k=1}^{K} c_{n,k}(t + 1)M_{n,k}(t + 1) \right)^2 \mid X(t) \right]
\]
\[
+ 2 \sum_{n=1}^{N} X_n(t)E[A_n(t + 1)] - 2 \sum_{n=1}^{N} E \left[ X_n(t) \sum_{k=1}^{K} c_{n,k}(t + 1)M_{n,k}(t + 1) \mid X(t) \right].
\] (40)

where \( c(t + 1) = (c_{n,k}(t + 1)) \), \( n \in \mathcal{N}, k \in \mathcal{K} \) is the observed connectivity matrix at the beginning of time slot \( t + 1 \). To prove (40), note that for each queue \( n \in \mathcal{N} \) one of the following conditions is satisfied.

- \( \sum_{k=1}^{K} H_{n,k}(t + 1) = \sum_{k=1}^{K} c_{n,k}(t + 1)M_{n,k}(t + 1) \): In this case, we can easily check that
  \[
  E[X_n^2(t + 1) - X_n^2(t) \mid X(t)] \leq NA_{max}^2
  + E \left[ \left( \sum_{k=1}^{K} c_{n,k}(t + 1)M_{n,k}(t + 1) \right)^2 \mid X(t) \right]
  + 2X_n(t)E[A_n(t + 1)] - 2E \left[ X_n(t) \sum_{k=1}^{K} c_{n,k}(t + 1)M_{n,k}(t + 1) \mid X(t) \right].
  \] (41)

- \( \sum_{k=1}^{K} H_{n,k}(t + 1) < \sum_{k=1}^{K} c_{n,k}(t + 1)M_{n,k}(t + 1) \): In this case, \( \sum_{k=1}^{K} H_{n,k}(t + 1) = X_n(t) \) and \( X_n(t + 1) = A_n(t + 1) \) as there are not enough packets in queue \( n \) to be served at time slot \( t + 1 \). It is not hard to check that inequality (41) is also satisfied in this case.
According to the above discussion, we can observe that inequality (41) is satisfied for all the queues and therefore (40) follows.

Using the fact that \( \sum_{k=1}^{K} c_{n,k}(t+1)M_{n,k}(t+1) \geq 0 \), we have the following inequality.

\[
\sum_{n=1}^{N} \left( \sum_{k=1}^{K} c_{n,k}(t+1)M_{n,k}(t+1) \right)^2 \leq \left( \sum_{n=1}^{N} \sum_{k=1}^{K} c_{n,k}(t+1)M_{n,k}(t+1) \right)^2 \leq (\min \{ N, K \})^2 \quad (42)
\]

Hence, the Lyapunov drift (40) is bounded by

\[
D(t+1) \leq NA_{\max}^2 + (\min \{ N, K \})^2 + 2 \sum_{n=1}^{N} X_n(t)E[A_n(t+1)]
\]

\[
- 2 \sum_{n=1}^{N} E \left[ X_n(t) \sum_{k=1}^{K} c_{n,k}(t+1)M_{n,k}(t+1) \mid X(t) \right]. \quad (43)
\]

By conditioning the last term of (43) on the connectivity process at time slot \( t+1 \), we will have

\[
D(t+1) \leq NA_{\max}^2 + (\min \{ N, K \})^2 + 2 \sum_{n=1}^{N} X_n(t)E[A_n(t+1)]
\]

\[
- 2 \sum_{n=1}^{N} E \left[ X_n(t) \sum_{k=1}^{K} c_{n,k}(t+1)M_{n,k}(t+1) \mid X(t) \right] \quad (44)
\]

Note that matching \( M \) in (44) depends on the selected policy. According to (3), we can see that by selecting MWM policy, the second term of (44) will be minimized and therefore the right hand side term in (44) will be minimized over all the existing server assignment policies.

In other words, for MWM policy and any arbitrary server assignment policy \( \pi \) we have

\[
D_{MWM}(t+1) \leq NA_{\max}^2 + (\min \{ N, K \})^2
\]

\[
+ 2 \sum_{n=1}^{N} E \left[ X_n(t) \left( A^T(t+1) - \sum_{c \in \mathcal{C}} P(c) \left( c \otimes M^{(\pi)}(t+1) \right) \right) \right] \mid X(t) \quad (45)
\]

where \( D_{MWM}(t+1) \) is the Lyapunov drift of MWM policy at time slot \( t+1 \).

It is important to note that the set of inequalities in (21) determines an open polytope \( \mathcal{R}' \) for which we have \( \mathcal{R}' = \mathcal{R} - bound(\mathcal{R}) \). In fact, Theorem 3 states that if the vector \( \lambda \) is strictly inside polytope \( \mathcal{R} \) then the system is stable. Now, consider the system of Figure 1 with arrival processes for which we have \( \lambda \in \mathcal{R}' \) at each time slot \( t \). Suppose that \( \delta \) is a positive real number

December 7, 2011 DRAFT
such that $\lambda + \delta 1_N \in \text{bound}(R)$. According to our discussion about deterministic and randomized policies, there exists a randomized policy $M^{\text{rnd}} = \{M^{\text{rnd}}(t)\}_{t=1}^{\infty}$ for which we will have

$$E \left[ \left( \sum_{c \in C} P(c) \left( c \otimes M^{\text{rnd}}(t) \right) \frac{1_T}{K} \right)^T \right] = \lambda + \delta 1_N. \quad (46)$$

Therefore,

$$E \left[ X(t) \left( A_T(t+1) - \sum_{c \in C} P(c) \left( c \otimes M^{\text{rnd}}(t+1) \right) \frac{1_T}{K} \right) \mid X(t) \right] = -\delta \sum_{n=1}^{N} X_n(t). \quad (47)$$

Putting together from (44) to (47), we conclude that

$$D^{\text{MWM}}(t+1) \leq N \alpha^{\text{max}}_2 + (\min\{N, K\})^2 - 2\delta \sum_{n=1}^{N} X_n(t), \quad (48)$$

and according to Lemma 4.1 in [4], this proves the stability of MQMS system. Furthermore, we can conclude the upper bound given in (22) for the average total queue occupancy of MWM policy.

**REFERENCES**

[1] K. Kar, X. Luo, and S. Sarkar, “Throughput-optimal scheduling in multichannel access point networks under infrequent channel measurements,” IEEE Trans. Wireless Comm., vol. 7, no. 7, pp. 2619–2629, July 2008.

[2] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Trans. Auto. Control, vol. 37, no. 12, pp. 1936–1949, Dec. 1992.

[3] ——, “Dynamic server allocation to parallel queues with randomly varying connectivity,” IEEE Trans. Inform. Theory, vol. 39, no. 2, pp. 466–478, Mar. 1993.

[4] L. Georgiadis, M. J. Neely, and L. Tassiulas, Resource Allocation and Cross Layer Control in Wireless Networks. Now Publisher, 2006.

[5] N. McKeown, A. Mekki, V. Anantharam, and J. Walrand, “On achieving 100% throughput in an input-queued switch,” IEEE Trans. Commun., vol. 47, no. 8, pp. 1260–1272, Aug. 1999.

[6] L. Tassiulas, “Scheduling and performance limits of networks with constantly changing topology,” IEEE Trans. Inform. Theory, vol. 43, no. 3, pp. 1067–1073, May 1997.

[7] ——, “Linear complexity algorithms for maximum throughput in radio networks and input queued switches,” in Proc. of IEEE INFOCOM, San Francisco, CA, USA, Apr. 1998.

[8] M. J. Neely and E. Modiano, “Logarithmic delay for nxn packet switches,” in Proc. of IEEE Workshop on High Performance Switching and Routing, Phoenix, AZ, USA, Apr. 2004.

[9] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, “Bounds on average delays and queue size averages and variances in input-queued cell-based switches,” in Proc. of IEEE INFOCOM, Anchorage, AK, USA, Apr. 2001.

[10] S. Kittipiyakul and T. Javidi, “A fresh look at optimal subcarrier allocation in ofdma systems,” in Proc. IEEE Conference on Decision and Control, Dec. 2004.
[11] ——, “Resource allocation in ofdma with time-varying channel and bursty arrivals,” IEEE Commun. Lett., vol. 11, no. 9, pp. 708–710, Sep. 2007.

[12] T. Javidi, “Rate stable resource allocation in ofdm systems: from waterfilling to queue-balancing,” in Proc. Allerton Conference on Communication, Control, and Computing, Oct. 2004.

[13] H. Halabian, I. Lambadaris, and C.-H. Lung, “Network capacity region of multi-queue multi-server queueing system with time varying connectivities,” in Proc. of IEEE Int. Symp. on Inform. Theory (ISIT’10), Austin, TX, USA, June 2010.

[14] ——, “On the stability region of multi-queue multi-server queueing systems with stationary channel distribution,” in Proc. of IEEE Int. Symp. on Inform. Theory (ISIT’11), Saint Petersburg, Russia, Aug. 2011.

[15] M. J. Neely, “Dynamic power allocation and routing for satellite and wireless networks with time varying channels,” Ph.D. dissertation, Massachusetts Institute of Technology, LIDS, 2003.

[16] M. J. Neely, E. Modiano, and C. E. Rohrs, “Dynamic power allocation and routing for time varying wireless networks,” IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad-hoc Networks, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[17] H. Halabian, I. Lambadaris, C.-H. Lung, and A. Srinivasan, “Throughput-optimal relay selection in multiuser cooperative relaying networks,” in IEEE MILCOM 2010, San Jose, CA, USA, Nov. 2010.

[18] A. Ganti, E. Modiano, and J. N. Tsitsiklis, “Optimal transmission scheduling in symmetric communication models with intermittent connectivity,” IEEE Trans. Inform. Theory, vol. 53, no. 3, pp. 998–1008, Mar. 2007.

[19] S. Kittipiyakul and T. Javidi, “Delay-optimal server allocation in multi-queue multi-server systems with time-varying connectivities,” IEEE Trans. Inform. Theory, vol. 55, no. 5, pp. 2319–2333, May 2009.

[20] H. Al-Zubaidy, I. Lambadaris, and I. Viniotis, “Optimal resource scheduling in wireless multi-service systems with random channel connectivity,” in Proc. of IEEE Global Communications Conference (GLOBECOM 2009), Honolulu, HI, USA, Nov. 2009.

[21] H. W. Kuhn, “The hungarian method for the assignment problem,” Naval Research Logistic Quarterly, pp. 2:83–97, 1955.

[22] D. Stoyan, Comparison Methods for Queues and other Stochastic Models. Chichester: J. Wiley and Sons, 1983.

[23] S. M. Ross, Stochastic Processes, 2nd ed. New York: J. Wiley and Sons, 1996.

[24] R. Lidl and G. Pilz, Applied abstract algebra, 2nd edition. New York: Springer, 1998.

[25] F. Kelly, A. Maulloo, and D. Tan, “Rate control for communication networks: Shadow prices, proportional fairness, and stability,” Journal of the Operational Research Society, vol. 49, no. 3, p. 237252, Mar. 1998.

[26] F. Kelly, “Charging and rate control for elastic traffic,” European Transactions on Telecommunications, vol. 8, no. 1, p. 3337, Jan. 1997.

[27] J. Lee, A First Course in Combinatorial Optimization. Cambridge, UK: Cambridge Texts in Applied Mathematics, Cambridge University Press, 2004.

[28] A. Schrijver, Theory of Linear and Integer Programming. Chichester: John Wiley and Sons, 1986.