Optimal Resource Scheduling in Wireless Multiservice Systems With Random Channel Connectivity[1]

Hussein Al-Zubaidy*, Ioannis Lambadaris*, Ioannis Viniotis**
*SCE-Carleton University, 1125 Colonel By Drive, Ottawa, ON, K1S 5B6 Canada
**ECE-North Carolina State University, Raleigh, NC, USA
Email: {hussein, ioannis.lambadaris}@sce.carleton.ca, candice@ncsu.edu

***Abstract*—We investigate an optimal scheduling problem in a discrete-time system of L parallel queues that are served by K identical servers. This model has been widely used in studies of emerging 3G/4G wireless systems. We introduce the class of Most Balancing (MB) policies and provide their mathematical characterization. We prove that MB policies are optimal among all work conserving policies; we define optimality as minimization, in stochastic ordering sense, of a range of cost functions of the queue lengths, including the process of total number of packets in the system. We use dynamic coupling arguments for our proof. We also introduce the Least Connected Server First/Longest Connected Queue (LCSF/LCQ) policy as an approximate implementation of MB policies. We conduct a simulation study to compare the performance of several work conserving policies to that of the optimal one. In the simulations we relax some of the mathematical assumptions we required for the analytical proofs. The simulation results show that: (a) in all cases, MB policies outperform the other policies, (b) randomized policies perform fairly close to the optimal one, and, (c) the performance advantage of the optimal policy over the other work conserving policies increases as the channel connectivity decreases.**

I. Introduction, Model Description and Prior Research

Emerging 3G/4G wireless networks can be categorized as high speed IP-based packet access networks [3]. They utilize the channel variability, using data rate adaptation, and user diversity to increase their channel capacity. These systems usually use a mixture of Time and Code Division Multiple Access (TDMA/CDMA). Time is divided into equal size slots, each of which can be allocated to one or more users. To optimize the use of the enhanced data rate, these systems allow several users to share the wireless channel simultaneously using CDMA. This will minimize the wasted capacity due to the allocation of the whole channel capacity to one user at a time even when that user is unable to utilize all of that capacity. Another reason for sharing system capacity between several users, at the same time slot, is that some of the user equipment at the receiving side might have design limitations on the amount of data it can receive and process at a given time. The connectivity of users to the base station in any wireless system is varying with time and can be best modelled as a random process. In the following subsection, we provide a more formal model description and motivation for the problem at hand.

A. Model Description

In this work, we assume that time is slotted into equal length deterministic intervals. We model the wireless system under investigation as a set of L parallel, symmetrical queues with infinite capacity (see figure 1); the queues correspond to the different users in the system. The queues share a set of K identical servers, each server representing transmission channels (or any other network resource, e.g., power, CDMA codes, etc.). We make no assumption regarding the number of servers relative to the number of queues, i.e., K can be less, equal or greater than L. The packets in this system are assumed to have constant length, and require one time slot to complete service. A server can serve one packet only at any given time slot. A server can only serve connected non-empty queues. Therefore, the system can serve up to K packets during each time slot. Those packets may belong to one or several queues.

The connectivity between a user and a channel is random. The state of the channel connecting the i^{th} queue to the j^{th} server during the n^{th} time slot is denoted by $G_{i,j}(n)$ and can be either connected ($G_{i,j}(n) = 1$) or not connected ($G_{i,j}(n) = 0$). Hence, $G_{i,j}(n)$ will determine (in a real system) if a transmission channel j can be used by user i or not.

The number of arrivals to the i^{th} queue during time slot n is denoted by $Z_i(n)$. We assume that $Z_i(n)$ and $G_{i,j}(n)$ for all $i = 1, 2, \ldots, L$ and $j = 1, 2, \ldots, K$ are independent Bernoulli random variables. Furthermore, we assume that $Z_i(n)$, for all i, n are i.i.d with parameter α. Similarly, $G_{i,j}(n)$, for all i, j, n are i.i.d with parameter p. We define $X_i(n)$ to represent the number of packets in the i^{th} queue at the beginning of time slot n.

A scheduler (or server allocation or scheduling policy) decides, at the beginning of each time slot, what servers will be assigned to which queue during that time slot. In figure 1, we used $q_j(t)$ to represent the index of the queue selected by the scheduler to be served by server j during time slot t.

[1] This work was supported by Mathematics of Information Technology and Complex Systems (MITACS) and Natural Sciences and Engineering Research Council of Canada (NSERC).

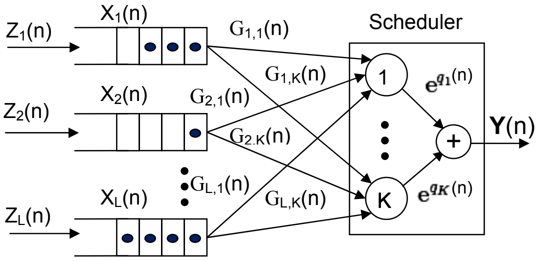


Fig. 1. Abstraction of downlink scheduler in a 3G wireless network.

The vector $\mathbf{e}^i \in \{0,1\}^L$ is an L-dimensional vector such that $e_i = 1, e_k = 0, \forall k \neq i$.

The objective of this work is to identify and study the optimal scheduling policy that minimizes a range of cost functions of the system queue occupancies, including the total number of queued packets, in the aforementioned system. The choice of the class of cost functions and the minimization process will be discussed later.

B. Previous Work and Our Contributions

In the literature, there is substantial research effort focusing on the optimal server allocation in wireless networks. Tassiulas and Ephremides [4] for example, tackled a similar problem where a single server (i.e., $L = 1$) can only be allocated to one user and can only serve one packet at each time slot. They proved, using coupling arguments, that LCQ (Longest Connected Queue) is optimal. In our work we show that LCQ is not always optimal in a multi-server system since servers can be assigned to one or more queues simultaneously. Bambos and Michailidis [5] worked on a similar model (a continuous time version of [4] with finite buffer capacity) and found that under stationary ergodic input job flow and modulation processes, both MCW (Maximum Connected Workload) and LCQ dynamic allocation policies maximize the stability region for this system. Furthermore, they proved that C-FES, a policy that allocates the server to the connected queue with the fewest empty spaces, stochastically minimizes the loss flow and maximizes the throughput [6].

Another relevant result is that reported by Ganti et al [7]. They presented a model for a satellite node that has K transmitters. The system was modelled by a set of parallel queues with symmetrical statistics competing for K identical servers. At each time slot, no more than one server is allocated to each scheduled queue. They proved, using stochastic coupling arguments, that LCQ, a policy that allocates the K servers to the K longest connected queues at each time slot, is optimal. This model is similar to the one we consider in this work, except that in our model one or more servers can be allocated to each queue in the system. A further, stronger difference between the two models is that we consider the case where each queue has *independent connectivities* to different servers. We make these assumptions for a more suitable representation of the 3G wireless systems described earlier. These differences make it substantially harder to identify (and even describe) the optimal policy (see section III-A). Lott and Teneketzis [8] tackled a multi class system of N weighted cost parallel queues and M servers. They also used the same restriction of one server per queue used in [7]. They showed that an index rule is optimal and provided conditions sufficient, but not necessary, to guarantee its optimality.

Koole et al [9] studied a model similar to that of [4] and [6]. They found that the Best User (BU) policy maximizes the expected discounted number of successful transmissions. Liu et al [10][11] studied the optimality of opportunistic schedulers (e.g., Proportional Fair (PF) scheduler). They presented the characteristics and optimality conditions for such schedulers. However, Andrews [13] showed that there are six different implementation algorithms of PF scheduler, none of which is stable. For more information on resource allocation and optimization in wireless networks the reader may consult [12], [14], [15], [16], [17], and [18].

In summary the main contributions of our work are the following: We introduce the class of Most Balancing (MB) policies and prove theoretically that they are optimal among all work conserving policies. The proof is based on coupling arguments of considerable complexity. We also provide an easily programmable algorithm for constructing the Least Connected Server First/Longest Connected Queue (LCSF/LCQ) policy as a viable approximation of MB policies. We compare the performance of several work conserving policies to that of the optimal one via simulations. In the simulations we relax some of the mathematical assumptions (on the arrival distribution) we required for the analytical proofs. The simulation results show that: (a) in all cases, MB policies outperform the other policies, (b) randomized policies perform fairly close to the optimal one, and, (c) the performance advantage of the optimal policy over the other work conserving policies increases as the channel connectivity decreases and the number of servers in the system increases (a result that is fully justified by intuition).

The rest of the paper is organized as follows. In section II, we introduce notation and formulate the optimization problem. In section III, we introduce the MB policies for server allocation in the described system. In section IV, we present the main result, i.e., the optimality of MB server allocation policies. In section V, we present the Least Balancing policies, and show that these policies perform the worst among all work conserving policies. In section VI, we present simulation results for the MB and four other policies.

II. Problem Formulation

Throughout this work, we will use the following notation:

- $\mathbf{X}(n) = (X_1(n), X_2(n), \ldots, X_L(n))$ is the vector of queue lengths (measured in number of packets) at the beginning of time slot n.
- $\mathbf{Z}(n) = (Z_1(n), Z_2(n), \ldots, Z_L(n))$ is the vector of the number of exogenous arrivals during time slot n.
- $\mathbf{Y}(n) = (Y_1(n), Y_2(n), \ldots, Y_L(n))$ is the vector of the number of packets withdrawn from the system during time slot n. $Y_i(n) \in \{0, 1, \ldots, K\}$ denotes the number of packets withdrawn from queue i during time slot n.

- $\mathbf{G}(n) \in \{0,1\}^{L \times K}$, where $G_{i,j}(n)$ is the channel connectivity random variable as defined previously.

We will use the notation $\mathbf{q}(n) = (q_1(n), \ldots, q_K(n))$ to denote the server allocation control vector, where $q_j(n) \in \{0, 1, \ldots, L\}$ denotes the index of the queue that is served by server j during time slot n. We also define a dummy queue, queue 0, such that $q_j(n) = 0$ means server j is idling during time slot n. Hence,

$$Y_i(n) = \sum_{j=1}^{K} \mathbb{1}_{\{i = q_j(n)\}}, \quad i = 1, 2, \ldots, L \tag{1}$$

where $\mathbb{1}_{\{\mathcal{A}\}}$ returns 1 if $\mathcal{A}$ is true and 0 otherwise. For our mathematical and theoretical treatment, the packet withdrawal vector $\mathbf{Y}(n)$ is a more rigorous representation of the controller action. On the other hand, from an engineering standpoint, $\mathbf{q}(n)$ has a more intuitive interpretation as we will see later in the implementation of the optimal policy.

Let the vector $(\mathbf{X}(n), \mathbf{G}(n))$ denote the state of the system at the beginning of time slot n. We assume that the controller has complete knowledge of the system state information at the beginning of each time slot. We define the set $\mathcal{Y}(\mathbf{x}, \mathbf{g})$ as the set of feasible packet withdrawals (controls) in state $(\mathbf{x}, \mathbf{g})$. A *feasible* control $(\mathbf{Y}(n) \in \mathcal{Y}(\mathbf{X}(n), \mathbf{G}(n)))$ is the one that designate one queue per server (at any given time slot) such that it satisfies the following constraints:

$$0 \le Y_i(n) \le \min\left(X_i(n), \sum_{j=1}^{K} G_{i,j}(n)\right), \quad \forall\, i,\, n, \tag{2}$$

$$\text{such that} \quad \sum_{i=1}^{L} Y_i(n) \le K, \quad \forall\, i,\, n. \tag{3}$$

In other words, the controller can only withdraw a total of up to K packets from the connected queues in the system and no more than the number of packets in the scheduled queues at any given time slot. We allow more than one server to be allocated to one queue when connected. The constraints in equations (2) and (3) are necessary but not sufficient, since they do not force the controller to assign a server only to one queue at a given time slot. That is why we needed to state that explicitly in the definition of the feasible control above.

For any feasible control $(\mathbf{Y}(n))$, the system described above evolves according to the following equation

$$\mathbf{X}(n+1) = \mathbf{X}(n) - \mathbf{Y}(n) + \mathbf{Z}(n), \qquad n = 1, 2, \ldots \tag{4}$$

We assume that arrivals during time slot n can only be added after removing served packets. Therefore, packets that arrive during time slot n have no effect on the controller decision at that time slot and may only be withdrawn during $t = n + 1$ or later.

III. Most Balancing (MB) Policies for Server Allocation

A server allocation policy π is a sequence of controls that represents a family of mappings which determine $Y_i(n)$, for all i and n, as a function of the past history and current state of the system $\mathbf{H}(n)$, and such that $\mathbf{Y}(n)$ is a feasible withdrawal vector. The history is given by

$$\begin{aligned} \mathbf{H}(1) &= (\mathbf{X}(1)), \quad \text{and} \\ \mathbf{H}(n) &= (\mathbf{X}(1), \mathbf{G}(1), \mathbf{Z}(1), \ldots, \mathbf{G}(n-1), \mathbf{Z}(n-1), \mathbf{G}(n)), \\ & \qquad n = 2, 3, \ldots \end{aligned} \tag{5}$$

Notice that we did not include the past controls, $Y_i(t), t < n$ in the history, and we included only the initial queue length vector $\mathbf{X}(1)$. This is because for any given policy π, these quantities can be recovered using $\mathbf{H}(n)$ and equation (4) above.

Let $\mathcal{H}_n$ be the set of all histories up to time slot n. Then a policy π can be formally defined as the sequence of measurable functions

$$\begin{aligned} & u_n : \mathcal{H}_n \longmapsto \mathcal{Z}_+^L, \\ \text{s.t.} \quad & u_n(\mathbf{H}(n)) \in \mathcal{Y}(\mathbf{X}(n), \mathbf{G}(n)), \quad n = 1, 2, \ldots \end{aligned} \tag{6}$$

where $\mathcal{Z}_+$ is the set of non-negative integers and $\mathcal{Z}_+^L = \mathcal{Z}_+ \times \ldots \times \mathcal{Z}_+$, where the Cartesian product is taken L times.

At each time slot, the following sequence of event happens: first, the connectivities $\mathbf{G}(n)$ and the queue lengths $\mathbf{X}(n)$ are observed. Second, the packet withdrawals $\mathbf{Y}(n)$ are determined according to the policy in effect. Finally, the new arrivals $\mathbf{Z}(n)$ are added to determine the next queue length vector $\mathbf{X}(n+1)$.

We will show in Section IV that the *Most Balancing* policies we introduce next are optimal: they minimize (in the stochastic ordering sense presented in the next section) a range of cost functions including the process of total number of packets in the system.

A. The Class of Most Balancing Policies

Given a state $(\mathbf{x}(n), \mathbf{g}(n))$ and a policy π that chooses the feasible control $\mathbf{y}(n)$ at time slot n, define $\hat{x}_i(n) = x_i(n) - y_i(n)$ as the size of queue $i, i = 1, \ldots, L$, after applying the control $y_i(n)$ and just before adding the arrivals during time slot n. The MB policies are the policies that minimize the total queue lengths differences (i.e., $\hat{x}_i(n) - \hat{x}_j(n)$, $\forall i, j$) between the queues in the system. In other words, the set of MB policies (Π^{MB}) is the set that satisfies the following relationship:

$$\Pi^{MB} = \underset{\mathbf{y}(n) \in \mathcal{Y}(\mathbf{x}, \mathbf{g})}{\operatorname{argmin}} \sum_{i=0}^{L} \sum_{j=i}^{L} |\hat{x}_i(n) - \hat{x}_j(n)|, \ \forall n = 1, 2, \ldots \tag{7}$$

where $x_0(n) = 0$ is the length of a dummy queue we introduced to simplify the mathematical treatment for this model.

Intuitively, the MB policies "attempt to balance the lengths of all queues in the system *as much as possible*, at every time slot n". The following example explains this behavior:

Let $L = 5$ and $K = 4$. Assume that at time slot n, we have $\mathbf{x}(n) = (5, 1, 2, 3, 1)$ and let $g_{i,j}(n) = 1$, $\forall i, j$. Then the MB policy will choose one of the following server allocations: $\mathbf{q}(n) = (1, 1, 1, 4)$, $\mathbf{q}(n) = (1, 1, 4, 1)$, $\mathbf{q}(n) = (1, 4, 1, 1)$ or $\mathbf{q}(n) = (4, 1, 1, 1)$. These are the only controls that will result in a packet withdrawal vector that satisfies (7). The withdrawal vector in this case will be $\mathbf{Y}(n) = (3,0,0,1,0)$, and $\mathbf{X}(n) - \mathbf{Y}(n) = (2,1,2,2,1)$. On the other hand, if the controller chooses $\mathbf{q}'(n) = (1,2,3,4)$, then $\mathbf{Y}'(n) = (1,1,1,1,0)$, and $\mathbf{X}(n) - \mathbf{Y}'(n) = (4,0,1,2,1)$. It is obvious that the last control does not satisfy (7) at time slot n.

B. LCSF/LCQ Policy

We introduce the *Least Connected Server First/Longest Connected Queue* (LCSF/LCQ) policy as a low complexity approximation of MB policies. LCSF/LCQ is the server scheduling policy that allocates each one of the K servers to its longest connected queue (not counting the packets already scheduled for service) starting with the least connected server (i.e., the server with the minimum number of queues connected to it).

Let $\mathbb{K}_j$ be the set of the non-empty queues that are connected to server j at time slot t, i.e., $\mathbb{K}_j = \{i : i = 1,2,\ldots,L, g_{i,j}(t) = 1, x_i(t) > 0\}$. Let $\mathbb{K}_{[i]}$ be the i^{th} element in the sequence $(\mathbb{K}_1,\ldots,\mathbb{K}_K)$, when ordered in ascending manner according to their size, i.e., $|\mathbb{K}_{[l]}| \geq |\mathbb{K}_{[m]}|$ if $l > m$. Ties are resolved arbitrarily. Then under the LCSF/LCQ policy, at every time slot t, the K servers are allocated as follows:

Algorithm 1.

```
1. $for \quad t = 1,2,\ldots \quad do \quad \{$
2.    Input: $\mathbf{X}(t), \mathbf{G}(t)$. Calculate $\mathbb{K}_{[l]},\ l = 1,\ldots,K$.
3.    $\mathbf{X}'(t) \longleftarrow \mathbf{X}(t),\ \mathbf{Y}(t) \longleftarrow \mathbf{0},\ \mathbf{q}(t) \longleftarrow \mathbf{0}$
4.    $for\ j = 1 \quad to \quad K \quad \{$ ; allocate servers sequentially
5.       $Q_{[j]}(t) = \min\left(l : l \in \left\{\underset{k:k\in\mathbb{K}_{[j]}}{\operatorname{argmax}}(X'_k(t) | X'_k(t) > 0)\right\}\right)$
6.       $for\ i = 1 \quad to \quad L \quad \{$
7.          $Y_i(t) = Y_i(t) + \mathbb{1}_{\{i = Q_{[j]}(t)\}}$
8.          $X'_i(t) = X_i(t) - Y_i(t) \quad \} \quad \}$
9. $\}$      ; End of Algorithm 1.
```

In the example we presented earlier, the LCSF/LCQ policy will choose the control $\mathbf{q}(n) = (1,1,1,4)$. Note that in line 5 of Algorithm 1, if the set $\mathbb{K}_{[j]}$ is empty, then the argmax returns the empty set. In this case, the j^{th} order server will not be allocated (i.e., will be idle during time slot t). Algorithm 1 produces two outputs, when it is run at $t = n$: $\mathbf{y}(n) \longleftarrow \mathbf{Y}(t)$ and $\mathbf{q}(n) \longleftarrow \mathbf{Q}(t)$. In accordance to the definition of a policy in Equation (6), the LCSF/LCQ policy can be formally defined as the sequence of time-independent mappings $u(\mathbf{x}(n), \mathbf{g}(n))$ that produce the withdrawal vector $\mathbf{y}(n)$ described in line 7.

IV. Optimality of MB Policies

In this section, we outline the main result of this work, that is, the optimality of the Most Balancing (MB) policies. We will establish the optimality of MB policies for a wide range of performance criteria including the minimization of the total number of packets in the system. We introduce the following definition.

A. Definition of Preferred Order

To prove the optimality of MB policies, we will need a methodology that enables comparison of the joint queue lengths under different policies. Definition D below defines the partial order $\prec_p$ (*Preferred over*) on the set $\mathcal{Z}_+^L$, where $\mathcal{Z}_+^L$ is as defined in section III.

Definition D: (Preferred Order). Let $\tilde{x}, x \in \mathcal{Z}_+^L$. We say that $\tilde{x} \prec_p x$ ($\tilde{x}$ is preferred over x) if one of the following statements holds:

1) $\tilde{x} \leq x$;
2) $\tilde{x}$ is a permutation of x. The two vectors differ only in two components i and j, such that $\tilde{x}_i = x_j$ and $\tilde{x}_j = x_i$;
3) $\tilde{x}$ is obtained from x by performing a *"balancing interchange"*. The two vectors differ in two components i and j only, where $x_j \leq \min(\tilde{x}_i, \tilde{x}_j) \leq \max(\tilde{x}_i, \tilde{x}_j) \leq x_i$, such that: $\tilde{x}_i = x_i - 1$ and $\tilde{x}_j = x_j + 1$.

Case D1 defines a partial order on $\mathcal{Z}_+^L$ such that $\tilde{x}_i \leq x_i$ for all i. In case D2, $\tilde{x}$ can be obtained from x by permuting components i and j. Case D3 describes a balancing interchange. We say that $\tilde{x}$ is more balanced than x, and can be obtained from x by moving one packet from a larger queue to a smaller one. For example, if $L = 2$ and $x = (5,2)$ then a balancing interchange will result in $x = (4,3)$. In summary, $\tilde{x}$ is preferred over x ($\tilde{x} \prec_p x$) if $\tilde{x}$ can be obtained from x by performing a sequence of permutations, balancing interchanges or packets removals.

B. The class $\mathcal{F}$ of cost functions

Let $\tilde{x}, x \in \mathcal{Z}_+^L$ be two vectors of the queue lengths. Then we denote by $\mathcal{F}$, the class of real-valued functions on $\mathcal{Z}_+^L$ that are monotone, non-decreasing with respect to the partial order $\prec_p$, that is, $f \in \mathcal{F}$ if and only if

$$\tilde{x} \prec_p x \Rightarrow f(\tilde{x}) \leq f(x) \tag{8}$$

It can be easily shown, from (8) and the definition of preferred order, that $f(x) = x_1 + x_2 + \ldots + x_L$ belongs to $\mathcal{F}$. This function corresponds to the total number of packets in the system.

The following theorem states the optimality of the MB policies, with respect to $f(\mathbf{X}(n))$ for all $n \in \mathcal{Z}_+$ and $f \in \mathcal{F}$ among the class of work conserving policies for server allocation in the symmetric system of parallel queues that was presented earlier. The full proof is presented in [19]. We provide only an outline here.

Theorem 1: A MB policy dominates any arbitrary work conserving policy in the system described earlier. That is,

$$\{f(\mathbf{X}^{MB}(t))\} \leq_{st} \{f(\mathbf{X}^{\pi}(t))\} \tag{9}$$

For a complete description of stochastic ordering the reader may refer to [1].

C. Outline of The Proof

We use the stochastic coupling method [2] in our proof. It is sufficient to show that $\mathbf{X}^{MB}(t) \prec_p \mathbf{X}^{\pi}(t)$ for all t and all sample paths, where $\{\mathbf{X}^{MB}(t)\}$ and $\{\mathbf{X}^{\pi}(t)\}$ are the queue lengths processes in the system under a MB policy and the system under any arbitrary policy π respectively, and when both systems start from the same initial state. Hence, for any $f \in \mathcal{F}$, we have $f(\mathbf{X}^{MB}(t)) \leq f(\mathbf{X}^{\pi}(t))$, for all t and all sample paths. Then using the definition of stochastic dominance, the stochastic process $\{f(\mathbf{X}^{MB}(t))\} \leq_{st} \{f(\mathbf{X}^{\pi}(t))\}$, that is, a MB policy *dominates* any other arbitrary policy π.

Proof of Theorem 1: For $n \geq 1$, let Π_n denote the set of policies that have the MB property at time slots $t = 1, 2, \ldots, n$. Let $\Pi_\infty = \bigcap_{n=1}^{\infty} \Pi_n$ denote the set of policies that have MB property at all times.

Denote by $\mathcal{Q}_n^{\pi} \equiv \{q_j(n) : j = 1, \ldots, K\}$ the set of queues served by the K servers under policy π during time slot n. Then we say that π deviates from MB at time n by at most $h > 0$, if at most h server reallocations are required to make π have the MB property (i.e., $\mathcal{Q}_n^{MB}$ differs from $\mathcal{Q}_n^{\pi}$ by at most h elements). Let Π_n^h be the set of policies that have the MB property before time slot n and deviate from MB at time slot n by at most h, i.e., $\Pi_n^h \subseteq \Pi_{n-1}$. Then $\Pi_n^0 = \Pi_n$, is the policy that has the MB property up to time slot n.

To prove the optimality of π^{MB} we start with an arbitrary policy π and apply a sequence of modifications that result in a sequence of policies $(\pi_1, \pi_2, \ldots)$. Each of these policies dominates the previous one, and is such that if $\pi_i \in \Pi_n$ and $i < j$ then $\pi_j \in \Pi_m$ and $n < m$. This means that π_j has the MB property for a longer period of time than π_i, and π_j dominates π_i. We will need the following lemma to complete the proof of Theorem 1.

Lemma 1: For any policy $\pi \in \Pi_\tau^h$ and $h > 0$, a policy $\tilde{\pi} \in \Pi_\tau^{h-1}$ can be constructed such that $\tilde{\pi}$ dominates π.

The proof of Lemma 1 is omitted due to space limitations; it can also be found in [19].

Now, we will proceed with the proof of Theorem 1. Let π be any arbitrary policy. We construct a sequence of policies starting from π by applying lemma 1 repeatedly. Each of these policies dominates the previous one. According to lemma 1, we obtain policies that belong to $\Pi_1^K, \Pi_1^{K-1}, \ldots, \Pi_1^0$. The last such policy belongs to Π_2^K. If we continue in such manner, we will obtain a policy π^n that belongs to Π_n^0 for increasing value of n. From the construction of π^n, we can see that it agrees with the preceding policy $\pi^{n-1} \in \Pi_{n-1}^0$ until time $n-1$.

For any arbitrarily large value of n, this sequence of policies defines a limiting policy π^* that for every n agrees with π^n until time n. This means that π^* acts similar to π^{MB} at all times and dominates all the previous policies including π. □

The optimality of MB policies is intuitively apparent since any such policy will "minimize the probability of server idling". This is because the MB policies serves the longest connected queues in the system and try to keep packets spread between all the queues in the system. Hence, minimizing the probability that servers will end up with connected queues that are all empty, and be forced to idle.

V. The Least Balancing Policies

The *Least Balancing* (LB) policies are the work conserving server allocation policies that at every time slot ($n = 1, 2, \ldots$), choose a packet withdrawal vector $\mathbf{y}(n) \in \mathcal{Y}(\mathbf{x}, \mathbf{g})$ that maximizes the differences between queues lengths in the system. The LB policies can do that by minimizing the length of the shortest non-empty queue in the system. Obviously, this will maximize the number of empty queues in the system. Hence, maximizing the chance that servers are forced to idle in future time slots because they are connected to empty queues only.

In other words, if Π^{WC} is the set of feasible work conserving policies for the system under consideration, and Π^{LB} is the set of all LB policies then

$$\Pi^{LB} \subset \ \Pi^{WC}, \ \textit{such that}$$
$$\Pi^{LB} = \underset{\mathbf{y}(n) \in \mathcal{Y}(\mathbf{x},\mathbf{g})}{\operatorname{argmax}} \sum_{i=0}^{L} \sum_{j=i}^{L} |\hat{x}_i(n) - \hat{x}_j(n)|, \ \forall n = 1, 2, \ldots \tag{10}$$

Intuitively, this algorithm will lead to the least balancing configuration of the queue lengths in the system. As such, it will yield the worst performance among all work conserving policies. The next theorem states this intuition formally. Its proof is analogous to that of Theorem 1 and will not be presented.

Theorem 2: A LB policy is dominated by any arbitrary work conserving policy.

We present next an approximate implementation of the LB policies.

A. MCSF/SCQ Policy

The *Most Connected Server First/Shortest Connected Queue* (MCSF/SCQ) policy is the service sharing policy that allocates each one of the K servers to its shortest connected queue (not counting the packets already scheduled for service) starting with the most connected server first. A feasible implementation algorithm of MCSF/SCQ policy is to allocate the K servers according to the following algorithm:

Algorithm 2.

```
for  $t = 1, 2, \ldots$  do  {
  Input: $\mathbf{X}(t), \mathbf{G}(t)$. Calculate  $\mathbb{K}_{[l]}, l = 1, \ldots, K$.
  $\mathbf{X}'(t) \longleftarrow \mathbf{X}(t), \ \mathbf{Y}(t) \longleftarrow \mathbf{0}, \ \mathbf{q}(t) \longleftarrow \mathbf{0}$
  for $j = K$  to  1  { ; Servers in descending order
      $Q_{[j]}(t) = \min \left( l : l \in \left\{ \underset{k:k \in \mathbb{K}_{[j]}}{\operatorname{argmin}} (X'_k(t) | X'_k(t) > 0) \right\} \right\}$
      for  $i = 1$  to  $L$  {
          $Y_i(t) = Y_i(t) + \mathbb{1}_{\{i = Q_{[j]}(t)\}}$
          $X'_i(t) = X_i(t) - Y_i(t)$  }  }
}      ; End of Algorithm 2.
```

On the long run, Algorithm 2 may lead to the least balancing configuration of the queues lengths in the system. Therefore, the MCSF/SCQ policy is a good approximation of the class of LB policies.

VI. Simulation Results

We used simulation to study the performance of the system described earlier under several work conserving policies. We focused our simulations on two themes. In the first one (representative results are shown in Figures 2 through 6), we study how various parameters affect system performance. In the second theme, we explored whether MB policies would still outperform in systems where the mathematical assumptions were relaxed. The metric we used in the simulation is EQ, the long-run average of the total number of queued packets in the system. Representative results are shown in Figures 7(a) through 7(c). The results are not exhaustive, but they do verify the strong intuition that MB policies would be optimal again.

The policies used in this simulation are: MB, LB, randomized, MCSF/LCQ, and LCSF/SCQ policies. The MB/LB policies provide "upper and lower" performance limits for all work conserving policies. In the simulation we used LCSF/LCQ, as described in section III-B, to approximate an MB policy; we used the MCSF/SCQ policy, as described in section V-A, to approximate an LB policy. The randomized policy is the one that at each time slot allocates each server, randomly and with equal probability, to one of its connected queues. The Most Connected Server First/Longest Connected Queue (MCSF/LCQ) policy differs from the LCSF/LCQ in the order that it allocates the servers. While an LCSF/LCQ policy allocates the servers after ordering them according to their connectivities to queues, starting with the least connected server first, the MCSF/LCQ use the reverse order, starting the allocation with the most connected server and ending it with the least connected one. However, it resembles LCSF/LCQ policies in that it allocates the server to its longest connected queue. The last policy that we studied is LCSF/SCQ (Least Connected Server First/Shortest Connected Queue). It allocates each server, starting from the one with the least number of connected queues, to its shortest connected queue. The difference from the LCSF/LCQ policy is obviously the allocation to the shortest connected queue. This policy will result in greatly unbalanced queues lengths and hence a performance that is closer to the LB policies.

Figure 2 shows the (average) total queue occupancy (total number of queued packets in the system) versus arrival rate under the five different policies. The system in this simulation is a symmetrical system with 16 parallel queues ($L = 16$), 16 identical servers ($K = 16$) and i.i.d. Bernoulli queue-to-server (channel) connectivity with parameter $p = 0.2$, where $p = P[G_{i,j}(t) = 1]$, $\forall\, i, j, t$.

The curves in figure 2 follow a distinctive shape that starts flat (actually, increasing with small slope) and ends with a rapid incline. This sudden increase happens at a point where the system crosses a "stability threshold". In this case, the queue lengths in the system will grow uncontrollably and the system becomes unstable. The graph shows that the MB policy outperforms[2] all other policies. It minimizes the (average) system queue occupancy and hence the queuing delay. We also noticed that it maximizes the system stability region and hence maximizes the system throughput. Another observation is that a LB policy is inferior to all the other policies we studied. As expected, the performance of the other three policies lies between that of the MB and the LB policies.

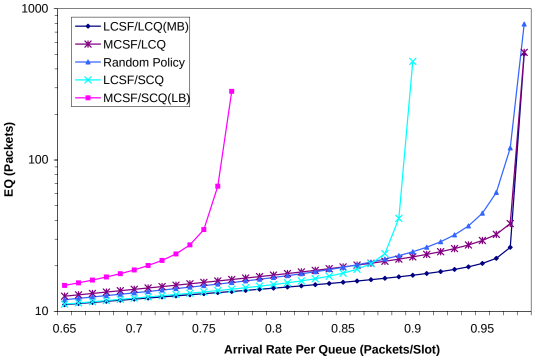


Fig. 2. Total queue occupancy versus load under different policies, $L = 16$, $K = 16$ and $p = 0.2$.

The MCSF/LCQ and LCSF/SCQ policies are variations of the MB and LB policies respectively. The performance of MCSF/LCQ policy is close to that of the MB policy. The difference in performance is due to the order of server allocation. On the other hand, the LCSF/SCQ policy shows a large performance improvement on that of the LB policy. This improvement is a result of the reordering of allocations of servers.

Figure 2 also shows that the randomized policy behaves very close to the optimal policy. However, it deviates more as the number of servers in the system increases, as the next set of experiments shows.

A. The Effect of The Number of Servers

In this section, we study the effect of the number of servers on policy performance. Figures 3 and 4 show the average total queue occupancy versus arrival rate per queue under the policies presented earlier, in a symmetrical system with $L = 16$ and $p = 0.2$. The two figures are corresponding to $K = 8$ and $K = 4$ respectively.

Comparing these two graphs to the one in figure 2, we notice the following: First, the performance advantage of the MB policy over the other policies increases as the number of servers in the system increases. More servers means larger action space. Selecting the optimal option, over any arbitrary policy, out of a larger space will certainly produce higher performance advantage than that selected from a smaller space. Second, the stability region of the system becomes narrower when less servers are used. This is true because less resources (servers) are available to be allocated by the working policy

[2] We have calculated 99% confidence intervals for all our experiments; we do not show them here for clarity of the graphs.

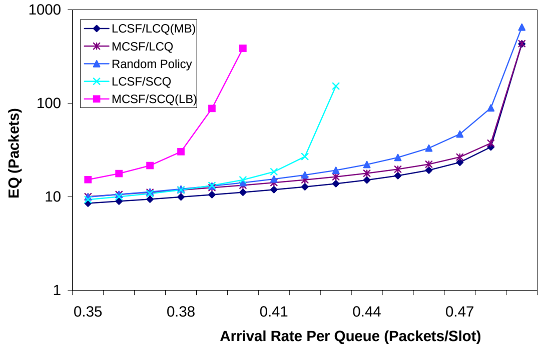


Fig. 3. Total queues occupancy versus load, $L = 16, K = 8$ and $p = 0.2$.

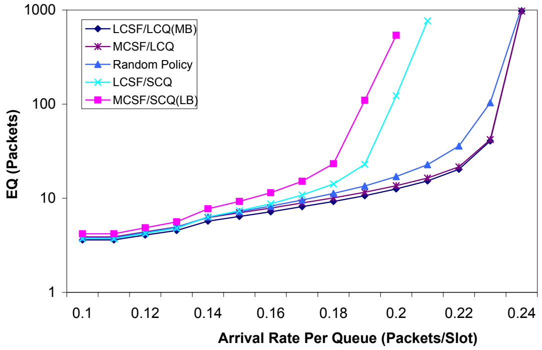


Fig. 4. Total queues occupancy versus load, $L = 16, K = 4$ and $p = 0.2$.

in this case. Finally, we noticed that the MCSF/LCQ performs very close to the MB policy in the case of $K = 4$. Apparently, when K is small, the order of server allocation does not have a big impact on the policy performance.

B. The Effect of Channel Connectivity

In this section we investigate the effect of channel connectivity on the performance of the different policies presented previously. Figures 5 and 6 show this effect under two different setups. Several observations are worth mentioning here.

First, we noticed that at higher channel connection probabilities ($p \geq 0.9$), the effect of the policy behavior on the system performance became less significant. Therefore, all five policies perform very close to each other. The MB policy still has very small advantage over the rest of the policies. LB still has the worst performance. As p increases, the probability that a server will end up connected only to a group of empty queues will be very small regardless of the policy in effect. Actually, when the servers have full connectivity to all queues (i.e., $p = 1.0$) then any work conserving policy will minimize the total number of packets in the symmetrical system we studied and they all will have identical performance. In other words, any work conserving policy will be optimal in a system with full connectivity.

Second, we noticed from all experiments that we conducted (as seen from the resulted simulation graphs) that there is an upper limit for the service provided by the system under investigation that is a function of the channel connectivity. Therefore, in order to keep the system stable, the average number of packets arrivals should never exceed that limit. Furthermore, in the symmetrical system that we studied, this limit is intuitively given by

$$\alpha < \frac{K}{L}\left(1 - (1 - p)^L\right) \qquad (11)$$

where α is the arrival rate, e.g., for Bernoulli arrival, $\alpha = P[Z_i(t) = 1], \forall\, i, t$. In other words, the average number of packets entering the system (αL) must be less than the rate they are being served. The service rate is proportional to the number of servers and the probability that a server is at least connected to one queue ($\sum_{i=1}^{K}(1 - (1 - p_i)^L) = K(1 - (1 - p)^L)$ when $p_i = p$, $\forall i$). When $p = 1.0$, the stability condition in (11) will be reduced to $\alpha L < K$, which makes intuitive sense in a system with deterministic service such as this one.

Finally, we noticed that the MCSF/LCQ policy performs very close to the MB policy. However, its performance deteriorates in systems with higher number of servers and lower channel connectivity. The intuition here is that the more servers are there the more effect the order of allocations of servers have on the performance. Since MCSF/LCQ differs from MB only by the order of servers allocation, therefore, more servers means more performance deviation from the MB policy. Also, the lower the connection probability the higher the probability that a server will end up with no connectivity to any non-empty queue, and hence be forced to idle.

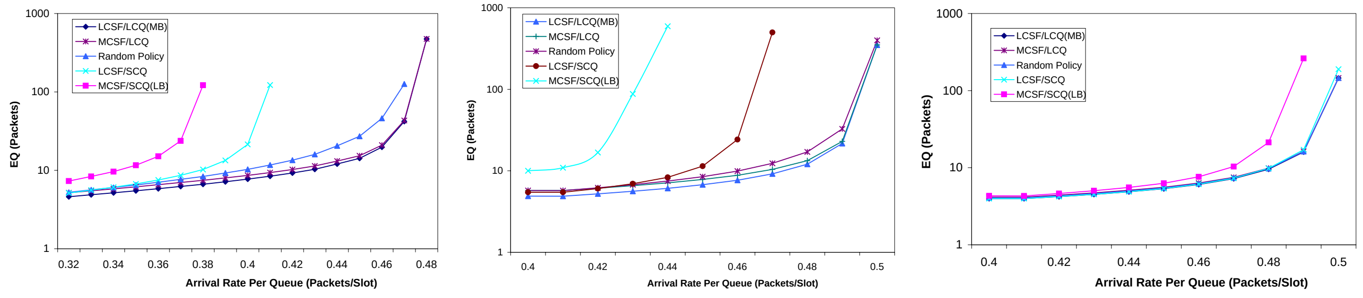
C. Batch Arrival With Random Batch Size

We studied the performance of the presented policies in the case of batch arrivals with uniformly distributed batch size. Figure 7 shows the total number of packets in the system versus load for the case of batch arrival with random batch size. The MB policy clearly dominates all the other policies. However, the performance of the other policies, including the LB one, approaches that of the MB policy as the average batch size increases. The performance of all the policies deteriorates when the arrivals become more bursty, i.e., the batch size increases. We conjecture that these trends will still apply for any arrival distribution.

VII. Conclusion

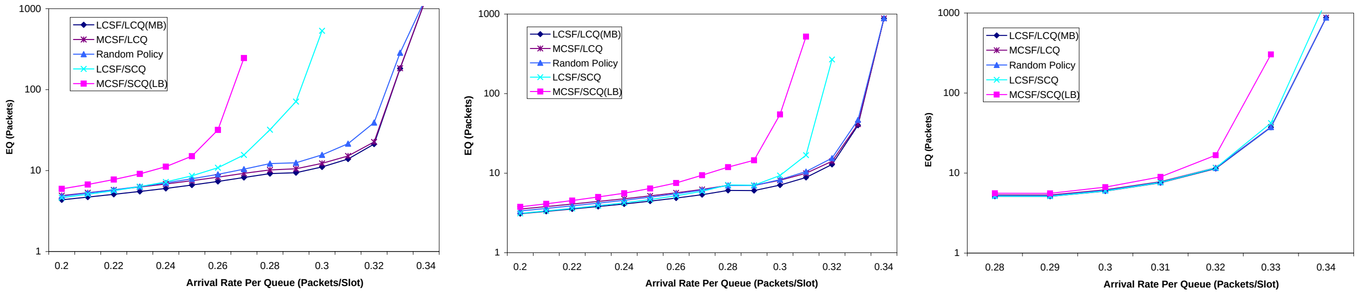
In this work, we presented a model for service sharing in emerging wireless systems. We investigated the class of Most Balancing policies. These policies serve the longest connected queues in the system in an effort to “equalize” the queue occupancies in the long run. A theoretical proof of the optimality of MB policies using stochastic dynamic coupling argument was presented. The LCSF/LCQ policy was designed as a low complexity approximation of MB policies.

A simulation study was conducted to study the performance of five different work conserving policies including the optimal one. The results showed that the Most Balancing policies outperformed all other policies, even when statistical assumptions were relaxed. As expected, the Least Balancing policy performed the worst. We also found that a randomized policy can perform very close to the optimal one in most situations.



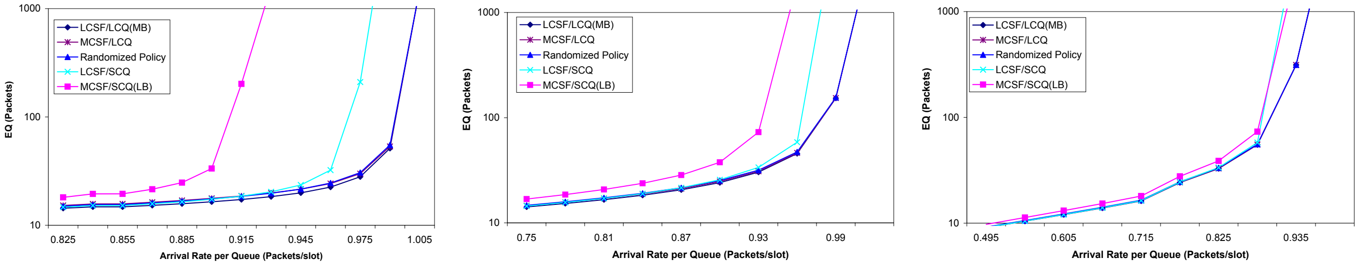
(a) $p = 0.3$ (b) $p = 0.5$ (c) $p = 0.9$

Fig. 5. Total number of packets in the system versus load under different policies, $L = 8$ and $K = 4$.



(a) $p = 0.3$ (b) $p = 0.5$ (c) $p = 0.9$

Fig. 6. Total number of packets in the system versus load under different policies, $L = 12$ and $K = 4$.



(a) $p = 0.5$, batch size =U(2) (b) $p = 0.6$, batch size =U(5) (c) $p = 0.8$, batch size =U(10)

Fig. 7. Total number of packets versus load, batch arrivals, $L = 16$ and $K = 16$.

REFERENCES

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

[2] T. Lindvall, *Lectures on the coupling method,* New York: Wiley(1992).

[3] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA-HDR: A bandwidth-efficient high-speed wireless data service for nomadic users", *IEEE Communication Magazine*, pages 7077, July 2000

[4] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Transactions on Information Theory*, Vol. 39, Issue 2, Page(s):466 - 478, March 1993.

[5] N. Bambos and G. Michailidis, "On the stationary dynamics of parallel queues with random server connectivities," *Proceedings of 34th Conference on Decision and Control*, (CDC), New Orleans, LA (1995).

[6] N. Bambos and G. Michailidis, "On parallel queueing with random server connectivity and routing constraints," *Probability in Engineering and Informational Sciences,* v.16, pp. 185-203, 2002.

[7] A. Ganti, E. Modiano and J. N. Tsitsiklis, "Optimal transmission scheduling in symmetric communication models with intermittent connectivity," *IEEE Transactions on Information Theory*, Vol. 53, Issue 3, March 2007.

[8] C. Lott and D. Teneketzis, "On the optimality of an index rule in multichannel allocation for single-hop mobile networks with multiple service classes," *Probability in the Engineering and Information Services*, vol. 14, no. 3, pp. 259-297, July 2000.

[9] G. Koole, Z. Liu and R. Righter, "Optimal transmission policies for noisy channels", *Operations Research*, Vol. 49, No. 6 (Nov. 2001).

[10] X. Liu, E.K.P. Chong, N.B. Shroff, "Optimal opportunistic scheduling in wireless networks", *IEEE 58th Vehicular Technology Conference,* Vol.3, Oct. 2003.

[11] X. Liu, E. Chong, and N. B. Shroff, "A framework for opportunistic scheduling in wireless networks" *Computer Networks*, 41(4), 2003.

[12] R. Agrawal and V. Subramanian, "Optimality of certain channel aware scheduling policies", *The 40th Annual Allerton Conference on Communication, Control, and Computing*, Monticello, Illinois, October 2002.

[13] M. Andrews, "Instability of the proportional fair scheduling algorithm for HDR", *IEEE Transactions on Wireless Communications*, 3(5), 2004.

[14] M. Andrews and L. Zhang, "Scheduling over a time-varying user-dependent channel with applications to high speed wireless data", *Journal of the ACM* (JACM), Volume 52 , Issue 5, September 2005.

[15] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length based scheduling and congestion control", *IEEE INFOCOM 05*, Miami, FL, March 2005.

[16] A. Stolyar, "On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation", *Operations Research*, 53:1225, 2005.

[17] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks", *IEEE/ACM Transactions on Networking*, 7(4):473-489, 1999.

[18] S. Shakkottai and A. Stolyar, "Scheduling for multiple flows sharing a time-varying channel: The exponential rule", *American Mathematical Society Translations*, Series 2, Vol. 207, 2002.

[19] H. Al-Zubaidy, "Optimal allocation of channel resources in wireless networks", *Technical report: SCE-09-02*, SCE, Carleton Univ, Feb 2009.