# Dynamic Packet Scheduler Optimization in Wireless Relay Networks

Hussein Al-Zubaidy*, Changcheng Huang[†], James Yan[†]
\* ECE, University of Toronto; † SCE, Carleton University.
hzubaidy@comm.utoronto.ca, huang@sce.carleton.ca, jim.yan@sympatico.ca.

*Abstract*—In this work[1], we investigate the optimal dynamic packet scheduling policy in a wireless relay network (WRN). We model this network by two sets of parallel queues, that represent the subscriber stations (SS) and the relay stations (RS), with random link connectivity. An optimal policy minimizes, in stochastic ordering sense, a cost function of the SS and RS queue sizes. We prove that, in a system with symmetrical connectivity and arrival distributions, a policy that tries to balance the lengths of all the system queues, at every time slot, is optimal. We use stochastic dominance and coupling arguments in our proof. We also provide a low-overhead algorithm for optimal policy implementation.

Keywords: Optimal Scheduling, Wireless Relay Network, Coupling Arguments, Stochastic Ordering, Most Balancing.

## I. INTRODUCTION

Fourth generation (4G) wireless systems are high-speed cellular networks with peak download data rates of 100 Mbps. IEEE 802.16j task group recommended the use of relay nodes and cooperation in 4G networks design [1]. In these systems, dedicated wireless relay nodes are deployed in order to achieve cooperative diversity. Wireless relays are spread over the coverage range of the cell. They have the effect of increasing coverage within a cell and facilitating the targeted data rate for 4G mobile users. These relays usually possess limited functionality and have low power consumption. Consequently, they are significantly cheaper than a full-scale base station. Early studies of cooperative communication were initiated by [2], [3] and [4]. Since then, the subject attracted the attention of many researchers, cf. [5] [6].

Most of the existing work in this area aimed at exploiting the diversity and multiplexing gain to improve some performance criteria, e.g., capacity and bandwidth utilization, outage probability, error rate, etc. These are often achieved through the use of adaptive modulation techniques, distributed space-time coding, or error-correction coding. In this work, we study this problem from a different perspective, the dynamic packet scheduling perspective. We are interested in the scheduling of packets on the uplink of a wireless relay network (WRN). Each of the subscriber or the relaying nodes is assumed to have a time-varying channel that can be modelled as a random process. We present a queueing model that captures the packet buffering, scheduling and routing processes as well as the intermittent channel connectivity in such network. We then use this model to study dynamic packet scheduling in such networks.

Dynamic packet scheduling enables the redistribution of the available resources to improve network performance. Furthermore, optimal packet scheduling policies can be determined under various operating constraints to optimize various performance criteria. This motivated the investigation of the optimal control problem that we present here. The inherent randomness of the wireless channel and the dynamic configuration of the nodes in wireless networks create a formidable challenge to such investigation. Therefore, it is wise and often necessary in such cases to make simplifying assumptions that result in mathematically tractable problem formulations. Otherwise, optimality results may not be attainable.

In this article, we investigate an optimal dynamic packet scheduling policy in a wireless relay network (WRN) composed of a base station (BS), $L$ subscriber stations (SS) and $K$ relay stations (RS), for any arbitrary $L$ and $K$. This network is modelled by two sets of queues with infinite size (see Figure 1). The wireless channel in such network is varying with time and can best be described by a random process. This assumption is widely used in literature, cf. [11], [12] and many other. A wireless link is assumed to be 'connected' with probability $p$ and 'not connected' otherwise. We further assume that the connectivity processes are independent across the wireless links. The transmission frame is divided into two halves; during the first half a SS node is scheduled to transmit (to a selected RS) and during the second half a RS node is scheduled to transmit to the base station. This model can be used to study scheduling and routing algorithms in multi-hop, wireless networks such as relay assisted, fourth generation wireless networks.

The optimization cost that we consider in this work is a monotonic, non-decreasing function of the system queue length. We prove using coupling arguments [10], that a *most balancing* (MB) policy minimizes, in stochastic ordering sense[2] [9], the cost function.

### A. Previous Work

The problem we are investigating lies in the area of optimal control in queueing networks. A related problem was first studied by Roseburg et al in [8]. They investigated an optimal

---

[2]Stochastic dominance is a stronger optimization notion than the expected cost minimization since the former implies the latter; however, the reverse is not true [7].
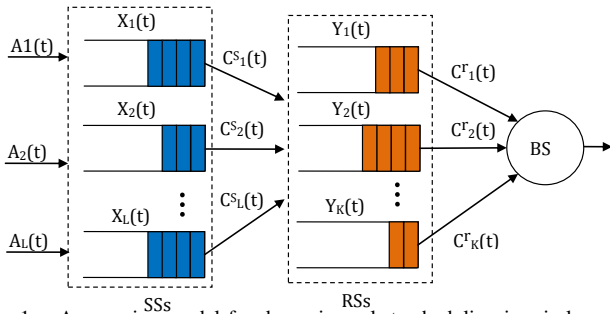
Fig. 1. A queueing model for dynamic packet scheduling in wireless relay networks.

control policy of service rate in a system of two queues in tandem. They proved, using dynamic programming argument, that a threshold policy is optimal in that it minimizes the expected cost function of the queue occupancy.

Another work related to the optimal control problem we are presenting here was reported by Tassiulas and Ephremides in [11]. They considered a model of parallel queues with a single server; they showed that a LCQ policy, a policy that allocates the server to its longest connected queue, minimizes the total number of packets in the system. The authors in [12] studied a satellite node with $K$ transmitters. They modelled the system by a set of parallel queues with symmetrical statistics competing for $K$ identical servers. At any given time slot in this model: (i) a server is connected to either all or no queues at all, (ii) at most one server can be allocated to each scheduled queue. Using stochastic coupling arguments, the authors proved that an LCQ (Longest, Connected Queue) policy is optimal.

In previous work [13], we studied the problem of optimal scheduling in a multi-server system of parallel queues with random queue-server connectivity. We relaxed assumptions (i) and (ii) in [12] above. We proved, using coupling arguments, that a Most Balancing (MB) policy is optimal in that it minimizes, in stochastic ordering sense, a cost function of the queue sizes in the system.

In [14], the authors proposed a cooperative multiplexing and scheduling algorithm for a wireless relay network with a single relay node. They showed that this algorithm outperforms the traditional opportunistic techniques in terms of spectral efficiency. The authors in [15] studied link scheduling in WRN with bandwidth and delay guarantees. They modelled the system using simple directed graph. They proposed an efficient algorithm to provide delay guarantee over WRN.

For a wireless relay network, the choices of relay node, relay strategy, and the allocation of power and bandwidth for each user are important design parameters which were investigated thoroughly in recent literature. Relay selection and cooperation strategies for relay networks have been investigated by [16] and [17] among others. Power control has been investigated by [18] and [19] and many others. However, the modern wireless networks are mostly IP-based, and therefore the optimization problem may be reduced to finding the optimal dynamic packet scheduling policies in these networks.

## B. Our Contributions

In this work, we developed a queueing model to study the process of packet scheduling in wireless relay network. Our main contributions can be summarized by the following:

- We develop a queueing model to study packet scheduling in WRN, see Figure 1.
- We introduce (in Equation 15) and show the existence of the class of optimal scheduling policies (i.e., the MB policies). We prove their optimality (Theorem 2) for packet scheduling in this model. The optimality criterion we use is stochastic ordering and the cost that is minimized belongs to a set of functions of the SS and RS queue lengths.
- We provide an implementation algorithm for packet scheduling policies in WRN. We also prove that this algorithm results in a MB policy.

The model we are presenting in this article differs from the previous work (in section I-A) in that it contains two sets of parallel queues in tandem rather than just one. At every time frame, the scheduler in this case must decide which SS node transmits during the first half of the frame, which RS queue receives the transmitted packets and which RS node transmits (to BS) during the second half of the frame. The task of solving this problem is quite challenging. The dependency between the two sets of queues (i.e., SS and RS), mainly the dependency of the scheduling controls (to be defined shortly in section II below) $U_2(t)$ and $U_3(t)$ on $U_2(t)$ and $U_1(t)$, added more complexity to the solution of this optimization problem. The approach we used in our proof (section IV) addressed this issue rigorously. The model and results we present here will help provide a sound theoretical ground to the problem of packet scheduling in WRN and can also be used to study multi-hop wireless networks in general.

The rest of this article is organized as follows; in Section II, we present a detailed description of the queueing model under investigation. In Section III, we define the class of "Most Balancing" policies. We present the optimality results in Section IV. Conclusions are given in Section V. Due to space limitation, we moved some of the proofs to a technical report (please refer to [23]).

## II. MODEL DESCRIPTION

We model the WRN by a discrete-time queueing system as shown in Figure 1. The objective is to find the optimal dynamic packet scheduling policy for this network. The optimal policy is the one that minimizes a cost function of the queue lengths (to be defined shortly).

In this model, time is slotted into constant intervals each of which is equal to one transmission frame. At every time slot, the following sequence of events happen: (a) the system state (queue sizes and connectivities) is observed, (b) a scheduler action (or control decision) is selected, and (c) the exogenous arrivals are added to their respective SS queues. The scheduler action involves (i) selecting a SS node to transmit to a RS node (denoted by $U_1(t)$), (ii) selecting the RS node that the scheduled packet is routed to (denoted by $U_2(t)$), and (iii) selecting a RS node to transmit to the base station (denoted

by $U_3(t)$). These actions are sequentially executed with order $U_1(t), U_2(t)$ then $U_3(t)$. A packet that arrives during the current time slot can only be considered for transmission in the subsequent time slots. We assume that the scheduler has complete knowledge of the system state when the decision time arrives. This is a realistic assumption for most infrastructure-based networks since they use a centralized control provided by the base station. They deploy a dedicated control channel that can be used to communicate such information[3]. Note that a SS that is not connected to the cell's pilot channel is outside its coverage area and will not be considered by the scheduler.

### A. Formulation and Statistical Assumptions

We define the following notation that we use to describe the model under investigation. Throughout this paper, we will use UPPER CASE, **bold face** and lower case letters to represent random variables, vector/matrix quantities and sample values respectively. In our notation, we define two dummy queues, one SS and one RS, that we denote by the index '0'. These queues are used to represent the idling action, i.e., a dummy packet is removed from queue 0 when no real packet from real queue is scheduled for transmission. We assume that the dummy queues have full connectivity at all times and initial sizes of 0. The dummy queues are required in order to facilitate the mathematical formulation of this optimal control problem. Let $\mathcal{L} = \{0, 1, \ldots, L\}$ (respectively $\mathcal{K} = \{0, 1, \ldots, K\}$) be the set of indices for the SS (respectively RS) stack of queues. For any time slot $t = 1, 2, \ldots$, we define the following:

- $\mathbf{X}(t) = (X_0(t), X_1(t), \ldots, X_L(t))$ is the queue length vector for SS nodes (measured in number of packets) at the beginning of time slot $t$, where $X_i(t) \in \{0, 1, 2, \ldots\}$ and $X_0(1) = 0$, i.e., we assume that the dummy queue is initially empty.
- $\mathbf{Y}(t) = (Y_0(t), Y_1(t), \ldots, Y_L(t))$ is the queue length vector for RS nodes at the beginning of time slot $t$, where $Y_i(t) \in \{0, 1, 2, \ldots\}$. We assume that $Y_0(t) = 0$ for all $t$.
- $\mathbf{A}(t) = (A_0(t), A_1(t), \ldots, A_L(t))$, where $A_i(t)$ is the number of exogenous arrivals to SS queue $i$ during time slot $t$.
- $\mathbf{C^s}(t) = (C_0^s(t), C_1^s(t), \ldots, C_L^s(t))$ (resp. $\mathbf{C^r}(t) = (C_0^r(t), C_1^r(t), \ldots, C_K^r(t))$) is the data channel connectivity for SS (resp. RS) nodes during time slot $t$, where $C_0^s(t) = C_0^r(t) = 1, \forall t$.
- $\mathbf{U}(t) = (U_1(t), U_2(t), U_3(t))$, s.t. $U_1(t) \in \mathcal{L}, U_2(t), U_3(t) \in \mathcal{K}$, is the scheduler decision (or control), where $\mathbf{U}(t) = (i, j, k)$ means that SS node $i$ is scheduled to transmit to RS node $j$ during the first half of time slot $t$, and RS node $k$ is scheduled to transmit to the BS during the second half of time slot $t$.

For ease of reference, we refer to the state of queue lengths and connectivities, i.e., the tuple $(\mathbf{X}(t), \mathbf{Y}(t), \mathbf{C^s}(t), \mathbf{C^r}(t))$, as the system "state" (denote by $\mathbf{S}(t)$) at time slot $t$.

We make the following statistical assumptions regarding the random processes in the system. The arrival processes $(A_i(t), i = 1, \ldots, L)$ are assumed to be i.i.d. Bernoulli[4], with parameter $q$. However, the arrivals to any RS queue at time $t$ is equal to the number of packets transmitted from a SS node to that RS node during that time slot. For convenience, we define $A_0(t) = W_0^s(t)$, where $W_0^s(t)$ is the number of packets withdrawn from queue 0 during time slot $t$, in order to ensure that $X_0(t) = 0$ for all $t$. Furthermore, transmitted dummy packets (i.e., fictitious packets from dummy queues) will not be added to the receiver queue (the RS queue that the packet is routed to). This assumption is intuitively correct, since fictitious packet is generated only when there is no real packet transmission.

The connectivity processes $C_i^s(t)$ and $C_j^r(t)$, for all $i = 1, \ldots, L$ and $j = 1, \ldots, K$ are assumed to be independent 2-state channels with connection probability $p$. It is further assumed that the connectivity and arrival processes are independent of each other.

Some of the statistical assumptions that we enforce are necessary for the tractability of the solution for this problem. Others can be relaxed for the cost of more complexity. For future work, we propose to relax some of these assumptions.

We define next the 'withdrawal' and the 'insertion' controls as a function of the scheduler control $\mathbf{U}(t)$ in order to simplify problem formulation and the proof of our results.

### B. Feasible Withdrawal/Insertion Control Vectors

Let $\mathbb{1}_{\{B\}}$ denote the indicator function for condition $B$. At any given time slot $t$, we define the SS (respectively the RS) *withdrawal vector* $\mathbf{W}^s(t)$ (respectively $\mathbf{W}^r(t)$) as follows:

$$W_i^s(t) = \mathbb{1}_{\{U_1(t)=i\}}, \quad \forall i \in \mathcal{L}, \quad and, \quad (1)$$
$$W_j^r(t) = \mathbb{1}_{\{U_3(t)=j\}}, \quad \forall j \in \mathcal{K}, \quad (2)$$

where $W_i^s(t)$ (respectively $W_j^r(t)$) represents the number of packets withdrawn from SS queue $i$ (respectively RS queue $j$) during time slot $t$. We also define the RS *insertion vector* as:

$$V_j^r(t) = \mathbb{1}_{\{U_2(t)=j\}}, \quad j = 1, \ldots, K, \quad (3)$$

where $V_j^r(t)$ represents the number of packets inserted to RS queue $j$ during time slot $t$. Note that we do not allow real packets to be inserted in the dummy queue. Similarly, we do not allow dummy packets to be inserted into real queues.

In the system described above and for any *feasible* withdrawal/insertion controls, the queue length for any SS node evolves according to the following relation:

$$\mathbf{X}(t+1) = \mathbf{X}(t) - \mathbf{W}^s(t) + \mathbf{A}(t) \quad (4)$$

Similarly, the queue length evolution for any RS node is given by the following relation:

$$\mathbf{Y}(t+1) = \mathbf{Y}(t) + \mathbf{V}^r(t) - \mathbf{W}^r(t) \quad (5)$$

It is apparent, from Equations (4) and (5) that Equations (1) – (3) do not guarantee feasibility of the withdrawal/insertion

---

[3]The model can be modified by allowing the scheduler to schedule multiple packets, or multiple time slots, at a time. This will reduce the required frequency for the state updates and hence the control overhead. We conjecture that our optimality results will hold for this case and under the added constraints.

[4]This assumption is widely used in the literature for analytic studies and optimization of wireless networks [11], [12] and [13].

vectors and hence the scheduling control vector $\mathbf{U}(t)$. Therefore, we provide the following feasibility condition:

A vectors $\mathbf{U}(t)$ is said to be a '*feasible scheduling control*' if the following condition is satisfied: '*a packet may only be withdrawn from a connected, non-empty queue*'. Formally, given the system state $\mathbf{S}(t)$ during time slot $t$, a scheduling control vector $\mathbf{U}(t)$ is *feasible* if and only if the resulted withdrawal/insertion vectors satisfy the following feasibility constraints:

$$0 \leq \quad W_i^s(t) \quad \leq \mathbb{1}_{\{X_i(t)>0\}} \cdot C_i^s(t), \quad \forall i \neq 0, \quad (6)$$

$$0 \leq \quad W_j^r(t) \quad \leq \mathbb{1}_{\{Y_j(t)>0\}} \cdot C_j^r(t), \quad \forall j \neq 0, \quad (7)$$

$$\sum_{i=0}^{L} W_i^s(t) = 1, \qquad \sum_{j=0}^{K} W_j^r(t) = 1. \quad (8)$$

According to Constraint (6), a packet is withdrawn from a SS queue $i$ only if queue $i$ is connected and non-empty, i.e., $X_i(t) > 0$ and $C_i^s(t) = 1$. Similarly, according to Constraint (7), a packet can only be withdrawn from a connected, non-empty RS queue. Constraint (8) insures that only one SS node and one RS node are allowed to transmit at any given time $t$. Let $\mathcal{U}(\mathbf{S}(t))$ be the set of all feasible scheduling controls when the system in state $\mathbf{S}(t)$.

### C. Policies for Dynamic Packet Scheduling

A *packet scheduling* policy $\pi$ (or policy $\pi$ for short) is a rule that determines the feasible control vectors $\mathbf{U}(t)$ for all $t$ as a function of the past history and current state of the system, where the state history $\mathbf{H}(t)$ is given by the following sequence of random variables

$$\begin{aligned} \mathbf{H}(1) &= (\mathbf{X}(1), \mathbf{Y}(1)), \quad \text{and for} \quad t \geq 2: \\ \mathbf{H}(t) &= (\mathbf{X}(1), \mathbf{Y}(1), \mathbf{C^s}(1), \mathbf{C^r}(1), \mathbf{A}(1), \dots, \\ &\quad \mathbf{C^s}(t{-}1), \mathbf{C^r}(t{-}1), \mathbf{A}(t{-}1), \mathbf{C^s}(t), \mathbf{C^r}(t)) \quad (9) \end{aligned}$$

Let $\mathcal{H}_t$ be the set of all state histories up to time slot $t$ and $\mathcal{Z}_+$ be the set of non-negative integers. Then a policy $\pi$ can be formally defined as the sequence of measurable functions

$$g_t : \mathcal{H}_t \longmapsto \mathcal{Z}_+^3, \quad \text{s.t. } g_t(\mathbf{H}(t)) \in \mathcal{U}(\mathbf{S}(t)), \; t = 1, 2, \dots \; (10)$$

The set of feasible[5] scheduling policies described in Equation (10) is denoted by $\Pi$. We are interested in a subset of $\Pi$, namely the class of *Most Balancing* (MB) policies. The main objective of this work is to prove the optimality of MB policies among all policies in $\Pi$.

### III. THE CLASS OF MB POLICIES ($\Pi^{MB}$)

In this section, we provide a description and mathematical characterization of the class of MB policies. Intuitively, the MB policies attempt to balance the sizes (leftover) of the SS queues as well as the RS queues in the system. This can be achieved by minimizing the queue length differences for the two sets of queues, at every time slot $t$. We present next a

---

[5]We say that a policy $\pi$ is feasible if it selects a feasible scheduling control $\mathbf{U}^\pi(t) \in \mathcal{U}(\mathbf{S}(t))$ for all $t$.

more formal characterization of MB policies. We first define the '*imbalance index*' ($\kappa(\mathbf{x})$) of a vector $\mathbf{x}$.

Let $\mathbf{x} \in \mathcal{Z}_+^M$ be an $M$-dimensional vector. The imbalance index of $\mathbf{x}$ is defined as follows:

$$\kappa(\mathbf{x}) : \mathcal{Z}_+^M \longmapsto \mathcal{Z}_+, \quad \kappa(\mathbf{x}) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} (x_{[i]} - x_{[j]}), \quad (11)$$

where $[k]$ denotes the index of the $k^{th}$ longest component in the vector $\mathbf{x}$.

The above definition ensures that the differences are non-negative and a pair of components is accounted for in the summation only once. We define next the "*balancing interchange*" for the vector $\mathbf{x}$. We use this operation in the proof for the optimality of MB policies.

**Definition: Balancing Interchange**: Given vectors $\mathbf{x}, \mathbf{x}^* \in \mathcal{Z}_+^M$, we say that $\mathbf{x}^*$ is obtained from $\mathbf{x}$ by performing a *balancing interchange* if the two vectors differ in two components $i > 0$ and $j \geq 0$ only, where

$$x_i^* = x_i - 1, \quad x_j^* = x_j + 1, \quad s.t. \quad x_i \geq x_j + 1. \quad (12)$$

For example, let $\mathbf{x} = (6, 3, 4)$ then $\mathbf{x}^* = (5, 4, 4)$ is obtained by choosing $i = 1$ and $j = 2$, and $\mathbf{x}^* = (5, 3, 5)$ is obtained by choosing $i = 1$ and $j = 3$.

To put the above definition into perspective, if the vector $\mathbf{x}$ represents a queue sizes vector then a balancing interchange would involve the removal of one packet from a larger queue $i$ and the insertion of that packet to a smaller queue $j$. We will show later (in Lemma 2) that such an interchange will decrease the imbalance index of the vector.

Given a state $\mathbf{s}(t)$ and a policy $\pi$ that chooses the feasible scheduling control $\mathbf{u}(t) \in \mathcal{U}(\mathbf{s}(t))$ at time slot $t$; define the "updated" queue sizes, $\hat{x}_i(t)$ and $\hat{y}_j(t)$, as the sizes of these queues after applying the control $\mathbf{u}(t)$ and just before adding the exogenous arrivals during time slot $t$. Note that because we let $a_0(t) = w_0^s(t)$, $\hat{x}_0(t)$ may be negative. The updated queue sizes can be stated as follows:

$$\hat{x}_i(t) = x_i(t) - w_i^s(t), \qquad i \in \mathcal{L}, \quad and, \quad (13)$$

$$\hat{y}_j(t) = y_j(t) + v_j^r(t) - w_j^r(t), \quad j \in \mathcal{K} \quad (14)$$

At any given time slot $t$, the imbalance indices for the updated SS and RS queue length vectors $\hat{\mathbf{x}}(t)$ and $\hat{\mathbf{y}}(t)$ are given by $\kappa(\hat{\mathbf{x}}(t))$ and $\kappa(\hat{\mathbf{y}}(t))$. The $L + 1^{st}$ SS queue as well as the $K + 1^{st}$ RS queue are the dummy queues defined in the previous section. From Equation (11), it follows that the minimum possible value of the imbalance index for a $M + 1$-dimensional vector $\mathbf{x}$ is equal to $M \cdot x_{[M]}$ which is indicative of a fully balanced system.

Denote by $\Pi^{MB}$ the set of all MB policies. We define the elements of $\Pi^{MB}$ as follows:

**Definition: Most Balancing Policies:** A *Most Balancing* (MB) policy is a policy $\pi \in \Pi$ that, at every $t = 1, 2, \dots$, chooses feasible scheduling control vector $\mathbf{u}(t) \in \mathcal{U}(\mathbf{s}(t))$ such that both imbalance indices $\kappa(\hat{\mathbf{x}}(t))$ and $\kappa(\hat{\mathbf{y}}(t))$ are minimized, i.e.,

$$\Pi^{MB} = \left\{ \pi \in \Pi : \operatorname*{argmin}_{\mathbf{u}(t) \in \mathcal{U}(\mathbf{s}(t))} \kappa(\hat{\mathbf{x}}(t)) \bigcap \operatorname*{argmin}_{\mathbf{u}(t) \in \mathcal{U}(\mathbf{s}(t))} \kappa(\hat{\mathbf{y}}(t)), \forall t \right\} \; (15)$$

In Equation (15), two sets of policies are defined through the two argmin functions. Policies in the first (respectively the second) set minimize the imbalance index for the SS (respectively the RS) queue length vector. The intersection of the two sets results in a set of policies that minimize the imbalance index for both vectors. We say that a policy has the "*MB property*" during time slot $n$, if it chooses a control that satisfies Equation (15) at $t = n$. Then a MB policy can be defined as the policy that has the MB property at every time slot.

Consider a system with $L = 3$, $K = 2$ and full connectivity (i.e., all channels are connected). Let $\mathbf{x}(t) = (5, 6, 2)$ and $\mathbf{y}(t) = (3, 2)$. A policy that chooses either $\mathbf{U}(t) = (2, 2, 1)$ or $\mathbf{U}(t) = (2, 2, 2)$, has the MB property at $t$.

The set $\Pi^{MB}$ in (15) is well-defined and non-empty, since the minimization is over a finite set of controls. Furthermore, the set of MB policies may have more than one element.

*A. MB Policy Implementation*

In this section, we provide a low-complexity heuristic algorithm (LCQ/SQ/LCQ) to implement MB policies. This algorithm is defined next:

**Definition: Algorithm LCQ/SQ/LCQ:** For every time slot $t$, Algorithm LCQ/SQ/LCQ selects the feasible vector $\mathbf{u}(t)$ such that $u_1(t)$ is the longest connected SS queue, $u_2(t)$ is the shortest RS queue, and $u_3(t)$ is the longest connected RS queue. That is

$$u_1(t) = l^s : l^s \in \operatorname*{argmax}_{i \in \mathcal{L}: c_i^s(t)=1} x_i(t) \tag{16}$$

$$u_2(t) = s^r : s^r \in \operatorname*{argmax}_{i \in \mathcal{I}} c_i^r(t), \; \mathcal{I} = \operatorname*{argmin}_{j \in \{1,\dots,K\}} y_j(t) \tag{17}$$

$$u_3(t) = l^r : l^r \in \operatorname*{argmax}_{j \in \mathcal{K}: c_j^r(t)=1} (y_j(t) + v_j^r(t)) \tag{18}$$

where $\mathbf{v}^r(t)$ is the RS insertion vector at time slot $t$. For SS queue 0 we add one extra condition for the sake of mathematical accuracy, that is: "*If $u_1(t) = 0$ then $u_2(t) = 0$.*" This may happen when the controller is forced to idle during the first half of the frame. □

Equation (17) identifies the shortest RS queue; if there are more than one RS queue that satisfy this condition, one of which (at least) is connected, then the connected one is the one selected as $u_2(t)$. Otherwise, $u_2(t)$ will be the shortest non-connected RS queue. The reason behind this extra condition is a special case where all the RS queues have the same size, then RS queue $u_2(t)$ will be the longest RS queue after adding the packet transmitted from SS queue $u_1(t)$. Selecting a connected RS queue in this case will provide the opportunity for the scheduler to select the longest RS queue as $u_3(t)$.

Cellular networks, including 4G wireless networks, are mostly infrastructure-based networks. Therefore, a centralized approach can be used for the implementation of packet-scheduler (i.e., in BS). Furthermore, in modern cellular networks a pilot channel is used to estimate, among other things, the channel signal-to-noise ratio by measuring the received signal power at the receiving end. In this case, the channel state information (CSI) as well as the queue state information can be made available to the controller with minimal efforts.

**Lemma 1.** *Algorithm LCQ/SQ/LCQ results in a feasible control vector $\mathbf{u}(t)$ for any $t$.*

*Proof:* According to Equations (16) – (18), packets are withdrawn from connected queues only. Furthermore, packets are withdrawn from the longest connected queue for both SS and RS stack of queues. This will insure that as long as there is at least a single connected, non-empty queue then the LCQ will not be empty. Therefore, Equations (6) and (7) are satisfied. Furthermore, Equation (8) is satisfied by definition of the scheduler control $\mathbf{u}(t)$. ∎

The following theorem states that the policy resulted from the proposed implementation algorithm is indeed a MB policy.

**Theorem 1.** *For the operation of the system presented in Section II and shown in Figure 1, a MB policy can be constructed using Algorithm LCQ/SQ/LCQ.*

To prove Theorem 1, we need Lemma 2 below. It quantifies the effect of performing a balancing interchange on the imbalance index $\kappa(\mathbf{x})$ of the $L+1$-dimensional vector $\mathbf{x}$. The proof of the lemma can be found in [23].

**Lemma 2.** *Let $\mathbf{x}$ and $\mathbf{x}^*$ be two $L + 1$-dimensional ordered vectors (in descending order); suppose that $\mathbf{x}^*$ is obtained from $\mathbf{x}$ by performing a balancing interchange of two components, $l$ and $s$, of $\mathbf{x}$, where $x_l > x_s$, such that, $s > l; x_l > x_a, \forall a > l$ and $x_s < x_b, \forall b < s$. Then*

$$\kappa(\mathbf{x}^*) = \kappa(\mathbf{x}) - 2(s - l) \cdot \mathbb{1}_{\{x_l \geq x_s + 2\}} \tag{19}$$

*1) Proof for Theorem 1:* We prove Theorem 1 by contradiction. We assume that a MB policy selects a control $\mathbf{u}(t)$ at $t$ that does not satisfy Equations (16) – (18). The control vector selected by Algorithm LCQ/SC/LCQ is feasible according to Lemma 1. Then using Lemma 2 we show that applying the controls selected by Equations (16) – (18) will result in imbalance indices $\kappa(\hat{\mathbf{x}}(t))$ and $\kappa(\hat{\mathbf{y}}(t))$ that are smaller than those under the MB policy which contradicts Equation (15). Therefore, $\mathbf{u}(t)$ must satisfy Equations (16) – (18) and the theorem follows.

*Proof for Theorem 1:* Given the system state $\mathbf{s}(t) = (\mathbf{x}(t), \mathbf{y}(t), \mathbf{c}^s(t), \mathbf{c}^r(t))$ at time slot $t$; let $l^s$ be the index of the longest connected SS queue (as in Equation (16)) and $s^r$ be the index of the shortest RS queue *before* executing the control $\mathbf{u}(t)$ that satisfies Equation (17); let $l^r$ be the index of the longest connected RS queue *after* executing the controls $u_1(t)$ and $u_2(t)$ and just before executing the control $u_3(t)$ (as in Equation (18)). Let $\pi \in \Pi^{MB}$ be a MB policy that selects the scheduler control $\mathbf{u}(t) \in \mathcal{U}(\mathbf{s}(t))$ during time slot $t$. To show a contradiction, we assume (to the contrary of Theorem 1) that $\mathbf{u}(t)$ does not satisfy Equations (16) – (18).

We show next that in this case, the control vector selected by Algorithm LCQ/SQ/LCQ during time slot $t$ will result in an imbalance index that is either (i) *less than* or (ii) *equal to* that obtained under a MB policy. Case (i) contradicts Equation (15); therefore, the MB policy must satisfy Equations (16) – (18). Case (ii) insures that LCQ/SQ/LCQ satisfies Equation (15). In either case, Theorem 1 will follow.

Consider the following three cases corresponding to Equations (16), (17) and (18):

1) $x_{u_1(t)}(t) < x_{l^s}(t)$, i.e., $u_1(t)$ does not satisfy Equation (16) during time slot $t$. Then $\hat{x}_{u_1(t)}(t) < \hat{x}_{l^s}(t) - 1$ (under $\pi$). According to Equation (12) we can perform a balancing interchange between components $u_1(t)$ and $l^s$ that will reduce the imbalance index $\kappa(\hat{\mathbf{x}}(t))$. Therefore, $\pi$ does not satisfy Equation (15) and hence it is not a MB policy. This contradicts the original assumption that $\pi \in \Pi^{MB}$. Therefore, we conclude that a MB policy must satisfy Equation (16). Note that $x_{u_1(t)}(t) > x_{l^s}(t)$ is not possible since queue $u_1(t)$ must be connected (feasibility constraint (6)) and queue $l^s$ is the longest connected queue by assumption.

2) $y_{u_2(t)}(t) > y_{s^r}(t)$, i.e., $u_2(t)$ does not satisfy Equation (17) during time slot $t$. Then $y_{u_2(t)}(t) + v^r_{u_2(t)}(t) > y_{s^r}(t) + v^r_{s^r}(t) + 1$. Similar to the previous case, we can perform a balancing interchange between queues $u_2(t)$ and $s^r$. Again this will reduce the imbalance index $\kappa(\mathbf{y}(t) + \mathbf{v}^r(t))$. Therefore, $\pi$ does not satisfy Equation (15) and hence it is not a MB policy. This contradiction leads us to conclude that $\pi$ must satisfy Equation (17). Since $s^r$ is the shortest queue by assumption, then $y_{u_2(t)}(t) < y_{s^r}(t)$ is not possible. However, if $y_{u_2(t)}(t) = y_{s^r}(t)$ s.t. $u_2(t) \neq s^r$; in this case, if $c^r_{u_2(t)}(t) = c^r_{s^r}(t)$ then $u_2(t)$ satisfies Equation (17) during time slot $t$. Otherwise, i.e., $c^r_{s^r}(t) > c^r_{u_2(t)}(t) = 0$, then $y_{u_2(t)}(t) + v^r_{u_2(t)}(t) = y_{s^r}(t) + v^r_{s^r}(t) + 1$. In this case, if $u_3(t) = s^r$ then $\hat{y}_{u_2(t)}(t) = \hat{y}_{s^r}(t) + 2$. A balancing interchange between queues $u_2(t)$ and $s^r$ will reduce the imbalance index $\kappa(\hat{\mathbf{y}}(t))$. Therefore, $\pi$ does not satisfy Equation (15) and hence it is not a MB policy. By contradiction $\pi$ must satisfy Equation (17).

If on the other hand $u_3(t) \neq s^r$ then a policy that chooses either $u_2(t)$ or $s^r$ while keeping $u_1(t)$ and $u_3(t)$ the same will result in the same imbalance index. Since $\pi \in \Pi^{MB}$ by assumption, then LCQ/SQ/LCQ $\in \Pi^{MB}$ as well.

3) $y_{u_3(t)}(t) + v^r_{u_3(t)}(t) < y_{l^r}(t) + v^r_{l^r}(t)$, i.e., $u_3(t)$ does not satisfy Equation (18) during time slot $t$. Then $\hat{y}_{u_3(t)}(t) < \hat{y}_{l^r}(t) - 1$. Again we can perform a balancing interchange between queues $u_3(t)$ and $l^r$ that will result in a reduction of the imbalance index $\kappa(\hat{\mathbf{y}}(t))$. Therefore, $\pi$ does not satisfy Equation (15) and hence it is not a MB policy. This contradiction leads us to conclude that a MB policy $\pi$ must satisfy Equation (18). Since $l^r$ is the longest connected queue by assumption and given the feasibility constraint (7), the case where $y_{u_3(t)}(t) + v^r_{u_3(t)}(t) > y_{l^r}(t) + v^r_{l^r}(t)$ is not possible.

The above cases are the only possible cases. We conclude that a MB policy $\pi \in \Pi^{MB}$ satisfies Equations (16) – (18) and Theorem 1 follow. ∎

## IV. OPTIMALITY OF MB POLICIES

In this section, we provide a proof for the optimality of MB policies. We start by defining a partial order to facilitate the comparison of the cost functions under different policies. We also define the class of cost functions for the optimality problem that we are investigating.

### A. Definition of the Partial Order

In order to prove the optimality of MB policies, we devise a methodology that enables comparison of the queue lengths under different policies. The idea is to define an order that we call the "preferred order" and use it to compare queue length vectors, for the SS queues as well as the RS queues, under different policies. We start by defining the relation $\sqsubseteq$ on $\mathcal{Z}_+^M$ for some $M > 0$ as follows; we say that the two vectors $\tilde{\mathbf{x}}$ and $\mathbf{x}$ are related via $\tilde{\mathbf{x}} \sqsubseteq \mathbf{x}$ if:

S1- $\tilde{x}_i \leq x_i$ for all $i$ (i.e., point wise comparison),
S2- $\tilde{\mathbf{x}}$ is a 2-component permutation of $\mathbf{x}$; the two vectors differ only in two components $i$ and $j$, such that $\tilde{x}_i = x_j$ and $\tilde{x}_j = x_i$, or
S3- $\tilde{\mathbf{x}}$ is obtained from $\mathbf{x}$ by performing a *"balancing interchange"* as in Equation (12).

**Definition:** The *preferred order* ($\preccurlyeq$) is defined as the transitive closure of the relation $\sqsubseteq$ on the set $\mathcal{Z}_+^M, M > 0$. □

The transitive closure of $\sqsubseteq$ on the set $\mathcal{Z}_+^M$ is the smallest transitive relation on $\mathcal{Z}_+^M$ that contains the relation $\sqsubseteq$ [20]. Intuitively, $\tilde{\mathbf{x}} \preccurlyeq \mathbf{x}$ if the vector $\tilde{\mathbf{x}}$ is obtained from $\mathbf{x}$ by performing a sequence of *reductions*, *permutations of two components* and/or *balancing interchanges*.

### B. Definition of the Class of Cost Functions $\mathcal{F}$

We denote by $\mathcal{F}$ the class of real-valued functions on the set $\mathcal{Z}_+^M$ that are monotone and non-decreasing with respect to the partial order $\preccurlyeq$. Given any two vectors $\tilde{\mathbf{x}}, \mathbf{x} \in \mathcal{Z}_+^M$, a function $f \in \mathcal{F}$ if and only if

$$\tilde{\mathbf{x}} \preccurlyeq \mathbf{x} \Rightarrow f(\tilde{\mathbf{x}}) \leq f(\mathbf{x}). \tag{20}$$

Using (20) and the definition of preferred order, we conclude that the function $f(\mathbf{x}) = x_1 + x_2 + \cdots + x_M$ belongs to $\mathcal{F}$. If $\mathbf{x}$ is a queue length vector, then this function corresponds to the total number of queued packets in the system.

### C. The Optimality Results

Let $B \leq_{st} C$ defines the usual stochastic ordering for two real-valued random variables $B$ and $C$ [9]. For the rest of this article, we say that a policy $\sigma \in \Pi$ '*dominates*' another policy $\pi$ if

$$f(\mathbf{X}^\sigma(t)) \leq_{st} f(\mathbf{X}^\pi(t)), \text{ AND } f(\mathbf{Y}^\sigma(t)) \leq_{st} f(\mathbf{Y}^\pi(t)), \tag{21}$$

for all $t = 1, 2, \ldots$, for all cost functions $f \in \mathcal{F}$; where $\mathbf{X}^\sigma$, respectively $\mathbf{Y}^\sigma$, is the SS (respectively RS) queue length vector under policy $\sigma$.

Note that from Equation (20) and the definition of stochastic ordering, $\mathbf{X}^\sigma(t) \preccurlyeq \mathbf{X}^\pi(t)$ and $\mathbf{Y}^\sigma(t) \preccurlyeq \mathbf{Y}^\pi(t)$, for all $t$ and all sample paths in a suitable sample space, is sufficient for policy domination. The sample space is the standard one used in stochastic coupling [10].

In what follows, let $\mathbf{X}^{MB}$ and $\mathbf{X}^\pi$ (respectively $\mathbf{Y}^{MB}$ and $\mathbf{Y}^\pi$) represent the SS queue sizes (respectively RS queue sizes) under $\pi^{MB} \in \Pi^{MB}$ and an arbitrary policy $\pi \in \Pi$. To prove the optimality of MB policies (i.e., Theorem 2), we need the following definitions and results. Define the following subsets of the set $\Pi$ of all feasible scheduling policies: (a) $\Pi_\tau \in \Pi$, the set of policies that has the MB property during slots $t \leq \tau$, and are arbitrary for $t > \tau$. (b) $\Pi_\tau^{u_2} \in \Pi$, the set of policies that has the MB property during time slots $t \leq \tau - 1$ and during $t = \tau$ choose the same controls $u_1(\tau)$ and $u_2(\tau)$ as

those selected by a MB policy and an arbitrary $u_3(\tau)$. Note that $u_3(\tau)$ may not be a MB control.

From the above definitions we have $\Pi = \Pi_0$. Note that the set $\Pi_n$ for any $t = n$ is not empty, since MB policies are elements of it. For $n = 0, 1, \ldots$, $\Pi_n$ form a monotone sequence of subsets, such that $\Pi_n \subseteq \Pi_n^{u_2} \subseteq \Pi_{n-1}$. In light of the above, the set $\Pi^{MB}$ can be defined as $\Pi^{MB} = \bigcap_{n=1}^{\infty} \Pi_n$. We will need the following lemmas to complete the proof of Theorem 2.

**Lemma 3.** *Given* $\pi \in \Pi_{\tau-1}$, *a policy* $\tilde{\pi} \in \Pi_\tau^{u_2}$ *can be constructed, such that* $\tilde{\pi}$ *dominates* $\pi$.

**Lemma 4.** *Given* $\pi \in \Pi_\tau^{u_2}$; *a policy* $\tilde{\pi} \in \Pi_\tau$ *can be constructed, such that* $\tilde{\pi}$ *dominates* $\pi$.

To prove Lemmas 3 and 4, we construct the policy $\tilde{\pi}$ as described in the lemmas statements. Then we use stochastic coupling arguments to prove that the two relations

$$\tilde{\mathbf{x}}(t) \preccurlyeq \mathbf{x}(t) \tag{22}$$
$$\tilde{\mathbf{y}}(t) \preccurlyeq \mathbf{y}(t) \tag{23}$$

are satisfied for all $t$. This will insure the dominance of policy $\tilde{\pi}$ over $\pi$. Please refer to the technical report in [23] for the full proofs for Lemmas 3 and 4.

The above lemmas provide a methodology to construct a MB policy from any arbitrary policy $\pi$ using stepwise improvements (i.e., by constructing policies that has the MB property for one extra time slot at every subsequent step) on the original policy while maintaining policy domination. The intermediate construction step (Lemma 3) is necessary in order to simplify the coupling arguments used in the proof of the lemma.

**Theorem 2.** *Consider the system presented in Figure 1 and described in Section II. A Most Balancing policy* $\pi^{MB} \in \Pi^{MB}$ *dominates any arbitrary policy for this system operation, i.e.,*

$$f(\mathbf{X}^{MB}(t)) \leq_{st} f(\mathbf{X}^\pi(t)), \quad AND \tag{24}$$
$$f(\mathbf{Y}^{MB}(t)) \leq_{st} f(\mathbf{Y}^\pi(t)), \quad \forall\, t = 1, 2, \ldots, \tag{25}$$

*for all* $\pi \in \Pi$ *and all cost functions* $f \in \mathcal{F}$.

*Proof:* Starting from an arbitrary policy $\pi$, we apply a series of modifications to $\pi$, using Lemmas 3 and 4, that result in a sequence of policies $(\pi_1, \pi_2, \ldots)$, such that: (i) policy $\pi_1$ dominates the original arbitrary policy $\pi$, (ii) $\pi_n \in \Pi_n$, in other words, policy $\pi_n$ has the MB property during time slots $t = 1, 2, \ldots, n$, and, (iii) $\pi_m$ dominates $\pi_n$ for $m > n$ (i.e., when $\pi_m$ has the MB property for a period of time $m - n$ slots longer than $\pi_n$).

By definition, $\pi$ is an arbitrary policy; therefore, $\pi \in \Pi_0$. We construct a policy $\tilde{\pi} \in \Pi_1^{u_2}$ that dominates $\pi$ according to Lemma 3. Using Lemma 4 we construct a second policy $\pi_1 \in \Pi_1$ that has the MB property during time slot $t = 1$ and dominates $\pi$. Repeating the construction steps above and using Lemmas 3 and 4 again for time slots $t = 2, 3, \ldots$ will result in a sequence of policies $\tilde{\pi}_n \in \Pi_n, n = 2, 3, \ldots$ that satisfy (i) – (iii) above, i.e., each subsequently constructed policy has the MB property for one more time slot (than the previous one)

and dominates all the previous policies including the original policy $\pi$.

Denote the limiting policy for the sequence of constructed policies as $n \longrightarrow \infty$ by $\pi^*$. In that case, $\pi^* \in \Pi^{MB}$ since it has the MB property at all time. Furthermore, we can conclude from the previous construction that $\pi^*$ dominates $\pi_n$, for all $n < \infty$ including the original policy $\pi$. The theorem follows since the initial policy $\pi \in \Pi$ is assumed to be arbitrary. $\blacksquare$

## V. CONCLUSION AND FUTURE WORK

In this work, we studied the wireless relay networks optimization problem from dynamic packet scheduling perspective. We provided a queueing model for these networks that takes into consideration the randomness of the wireless channel connectivity. We introduced a class of packet scheduling policies, the *most balancing* (MB) policies. We proved, using stochastic dominance and coupling method, that MB policies dominate all other policies in that they minimize, in stochastic ordering sense, a class of cost functions of the system queue lengths including the total number of packets in the system. We proposed an implementation algorithm and proved that it will produce a MB policy for the proposed system. The results presented in this article provide a concrete understanding of the optimal scheduling policy structure in homogeneous wireless relay networks and multi-hop wireless networks in general.

This work can be extended in several directions. One such extension is to use multiple servers in the model presented in Figure 1 to represent multiple OFDMA or CDMA channels each serving one or multiple packets at every time slot. Another extension is to use finite-state Markov channel model instead of the two-state model we are using in this work. It is worth noting that these extensions are not trivial and some may render the model intractable.

### REFERENCES

[1] "MMR Harmonized Contribution on 802.16j (Mobile Multihop Relay) Usage Models," Document No. IEEE 802.16j-06/015, Sep. 2006.
[2] A. Sendonaris, E. Erkip and B. Aazhang, "User Cooperation Diversity–Part I: System Description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.
[3] N. Laneman, D.N.C. Tse and G.W. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behaviour," *IEEE Trans. Inform. Theory*, vol. 50, pp. 3062–3080, 2004.
[4] A. Nosratinia, T.E. Hunter, A. Hedayat, "Cooperative Communication in Wireless Networks," *IEEE communications Magazine*, Vol. 42, pp. 74–80, Oct. 2004.
[5] A. Sendonaris, E. Erkip, and B. Aazhang, "Increasing uplink capacity via user cooperation diversity," in *Proc. of IEEE Int. Symp. on Inform. Theory*, Cambridge, MA, USA, Aug. 1998.
[6] V. Stankovic, A. Host-Madsen, and Z. Xiong, "Cooperative diversity for wireless ad hoc networks," *IEEE Signal Processing Magazine*, vol. 23, pp. 37–49, Sep. 2006.
[7] S. M. Ross, *Stochastic Processes.* 2nd ed. New York: Wiley, 1996.
[8] Z. Roseburg, P. Varaiya, J. Walrand, "Optimal Control of Service in Tandem Queues," *IEEE Trans. Auto. Control.* AC27, pp. 600–610, 1982.

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

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

[11] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. on Inf. Theory*, Vol. 39, Issue 2, pp.466–478, Mar. 1993.

[12] A. Ganti, E. Modiano and J. N. Tsitsiklis, "Optimal Transmission Scheduling in Symmetric Communication Models With Intermittent Connectivity," *IEEE Trans. on Inform. Theory,* Vol. 53, Issue 3, pp. 998–1008, Mar. 2007.

[13] H. Al-Zubaidy, I. Lambadaris and I. Viniotis, "Optimal Resource Scheduling in Wireless Multi-service Systems With Random Channel Connectivity," in *Proc. IEEE Globecom '09*, Honolulu, HI, USA, Dec. 2009.

[14] Y. Shi, W. Zhang, K. B. Latief, "Cooperative Multiplexing and Scheduling in Wireless Relay Networks," in *Proc. IEEE Int. Conf. on Commun.* (ICC'08), 2008.

[15] C.Y. Hong, A.C. Pang, "Link Scheduling with QoS Guarantee for Wireless Relay Networks," in Proc. *IEEE Conf. on Computer Commun.* (INFOCOM), Rio De Janeiro, Brazil, 2009.

[16] V. Sreng, H. Yanikomeroglu, and D. Falconer, "Relay selection strategies in cellular networks with peer-to-peer relaying," in *Proc. IEEE Vehic. Tech. Conf.*, pp. 1949–1953, Oct. 2003.

[17] M. Yu and J. Li, "Is amplify-and-forward practically better than decode-and- forward or vice versa?" in *Proc. IEEE Inter. Conf. Acoustics, Speech, and Signal Processing,* (ICASSP' 05), vol. 3, pp. 365–368, Mar. 2005.

[18] A. Host-Madsen and J. Zhang, "Capacity bounds and power allocation for wireless relay channels," in *Proc. IEEE Trans. Information Theory,* vol. 51, no. 6, pp. 2020–2040, June 2005.

[19] M. Chen, S. Serbetli, and A. Yener, "Distributed power allocation for parallel relay networks," in *Proc. IEEE Global Telecom. Conf.* (GLOBECOM' 05), vol. 3, pp. 1177–1181, Nov. 2005.

[20] R. Lidl and G. Pilz, *Applied abstract algebra*, 2nd ed., Undergrad. Texts in Math., Springer, 1998.

[21] J. Walrand, "A note on optimal control of a queuing system with two heterogeneous servers," *Systems and Control Letters*, Vol. 4, pp. 131–134, 1984.

[22] P. Nain, P. Tsoucas and J. Walrand, "Interchange arguments in stochastic scheduling", *Journal of Applied Probability*, Vol 27, pp. 815–826, 1989.

[23] H. Al-Zubaidy, C.C. Huang and J. Yan, "Dynamic Packet Scheduler Optimization in Wireless Relay Networks," technical report arXiv:1104.3165v1 [cs.NI], http://arxiv.org/pdf/1104.3165.

**ChangCheng Huang** received his B. Eng. in 1985 and M. Eng. in 1988 both in Electronic Engineering from Tsinghua University, Beijing, China. He received a Ph.D. degree in Electrical Engineering from Carleton University, Ottawa, Canada in 1997. From 1996 to 1998, he worked for Nortel Networks, Ottawa, Canada where he was a systems engineering specialist. He was a systems engineer and network architect in the Optical Networking Group of Tellabs, Illinois, USA during the period of 1998 to 2000. Since July 2000, he has been with the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada where he is currently an associate professor. Dr. Huang won the CFI new opportunity award for building an optical network laboratory in 2001. He was an associate editor of IEEE Communications Letters from 2004 to 2006. Dr. Huang is a senior member of IEEE.

**James Yan** is currently an adjunct research professor with the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada. Dr. Yan received his B.A.Sc., M.A.Sc., and Ph.D. degrees in electrical engineering from the University of British Columbia, Vancouver, Canada. From 1976 to 1996 with Bell-Northern Research (BNR) and from 1996 to 2004 with Nortel, he was a telecommunications systems engineering manager responsible for projects in performance analysis of networks and products, advanced technology research, planning new network services and architectures, development of network design methods and tools, and new product definition. From 1988 to 1990, he participated in an exchange program with the Canadian Federal Government, where he was project prime for the planning of the evolution of the nationwide federal government telecommunications network. Dr. Yan is a member of IEEE and Professional Engineers Ontario.

**Hussein Al-Zubaidy** received his PhD from Carleton University, Ottawa, Canada, in 2010, majoring in Electrical and Computer Engineering. He got his B.Sc. (with honors) and M.Sc. from the University of Technology - Electrical Engineering department - Baghdad, Iraq. His research interests include: stochastic modelling and optimization, algorithms design, scheduling in emerging wireless networks, optimal resource allocation, mesh and Ad-Hoc networks, cooperative communication networks and security in wireless networks. He was the recipient of many honours and awards including: Ontario Graduate Scholarship (OGS), NSERC Visiting Fellowship, NSERC Summer Program in Taiwan and NSERC Post-Doctoral Fellowship. Dr. Al-Zubaidy is a member of IEEE. Currently, Dr. Al-Zubaidy is a post-doctoral fellow at ECE, University of Toronto, Toronto, Canada.