

Dynamic Packet Scheduler Optimization in Wireless Relay Networks

Hussein Al-Zubaidy, Changcheng Huang, James Yan

SCE, Carleton University, Ottawa, ON, K1S 5B6 Canada.

e-mail: {hussein, huang}@sce.carleton.ca, jim.yan@sympatico.ca.

January 31, 2011

Abstract

In this article, we investigate the optimal dynamic packet scheduling policy in a wireless relay network (WRN). We model this network by a two sets of parallel queues, that represent the subscriber stations (SS) and the relay stations (RS). We model the wireless channel connectivity by a two-state random process. 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. An optimal policy minimizes, in stochastic ordering sense, the process of cost function of the SS and RS queue sizes. We use stochastic dominance and coupling arguments in our proof. We also provide a low-overhead algorithm for optimal policy implementation.

1 Introduction

Fourth generation (4G) wireless systems are high-speed cellular networks with peak download data rates of 100 Mbps. These systems deploy dedicated wireless relay nodes to achieve cooperative diversity. In such systems, wireless relay nodes 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 relay nodes 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 has attracted the attention of many researchers as well as the industry, c.f. [5] [6]. IEEE 802.16j task group recommended the use of relay nodes and user cooperation in cellular networks design [7].

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. However, performance optimization in such systems involves more than one layer and therefore requires a cross-layer design perspective.

In this work, we study the problem of cooperative communication in wireless cellular networks from new 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 and to achieve performance gain. 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 and K relay stations. L can be larger, smaller or equal to K . This network is modelled by two sets of queues with infinite size. The wireless channel in such network is varying with time and can best be described by a random process. A wireless link (that connects a SS node to a RS node or a RS node to the BS) 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 relay station) and during the second half a RS node is scheduled to transmit (to the base station).

This model is widely used to study scheduling and routing algorithms in multi-hop, wireless networks such as relay assisted, fourth generation wireless networks. Cellular networks, including 4G wireless networks, are mostly infrastructure-based networks. Therefore, a centralized approach can be used for the implementation of packet-scheduler in these networks. 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 [8]. In this case, the channel state information (CSI) as well as the queue state information (backlog) can be made available to the controller with minimal efforts.

The optimization cost that we consider in this work is a monotone, non-decreasing function of the system queue occupancy. The optimization criterion we use in this work is *stochastic dominance* [19]. Stochastic dominance is a stronger optimization notion than the expected cost minimization since the former implies the later; however, the reverse is not true [9]. This cost may represent the expected total queueing delay, since this delay can be stated as a function (i.e., the expected value) of the total number of buffered packets in the system, according to Little’s law. We prove using stochastic dominance and coupling arguments [20], that a *most balancing* (MB) policy minimizes, in stochastic ordering sense, the process of cost function mentioned above. A MB policy is a dynamic packet scheduling policy that at every scheduling opportunity, i.e., at every time frame, tries to minimize the imbalance of the SS queue sizes as well as the RS queue sizes.

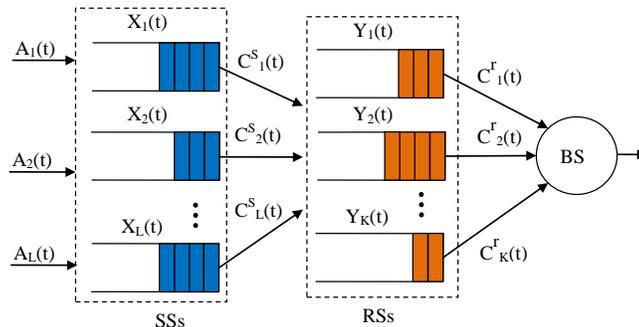


Figure 1: A model for packet scheduling in a 2-hop wireless relay networks.

1.1 Previous Work

The optimality 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 [10]. They investigated an optimal 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. An extension to their work is to include random connectivity between the two queues to represent wireless channel. This is a special case of the model we presented in Figure 1, where $L = K = 1$.

Another related work to the optimal control problem we are presenting here was reported by Tassiulas and Ephremides in [21]. 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. Bambos and Michailidis [22] worked on a model similar to that in [21] and proved that, under stationary ergodic input job flow, both MCW (Maximum Connected Workload) and LCQ allocation policies maximize the stability region for this system. The same authors proved similar results for continuous time models in [23].

The authors in [24] 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 that it stochastically minimizes a cost function of the queues' length. This policy allocates, at every time slot, the K servers to the K longest connected queues.

In previous work [25], 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) above, i.e., a server may be connected to a subset of the queues and not to the others, and we allow multiple servers to be allocated to a single queue at a given time slot. We proved, using coupling arguments, that the class of Most Balancing (MB) policies is optimal in that they minimize, in stochastic ordering sense, a cost function of the queues' sizes in the system.

The model we are presenting in this article differs from the above 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.

In [11], 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 [12] 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. They proved that the worst-case performance of the proposed algorithm compared to an optimal one is bounded.

In [13], the authors presented opportunistic relaying with decode-and-forward (DaF) and amplify-and-forward (AaF) strategies under an aggregate power constraint. They showed that opportunistic DaF relaying is outage-optimal. Furthermore, they showed that opportunistic AaF relaying is outage-optimal among single-relay selection methods. The authors in [14] proposed a utility maximization framework for wireless cooperative cellular data network with a base station and many cooperative subscribers. They claim that their strategy optimizes relay selection as well as power allocation.

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 [15] and [16] among others. Power control has been investigated by [17] and [18] 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.

1.2 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 propose a study of wireless relay networks from dynamic packet scheduling perspective. A queueing model is developed for this study, see Figure 1.
- We introduce (in Equation 15) and show the existence of the class of optimal scheduling policies (i.e., the class of MB policies) for the model of Figure 1.
- We prove the optimality of MB policies (Theorem 2) for packet scheduling in wireless relay networks with symmetrical parameters, i.e., arrivals and connectivities. The optimality criterion we use is stochastic ordering and the cost that is minimized belongs to a set of functionals of the SS and RS queue sizes.
- We provide an implementation algorithm for the MB policies. At every time frame, this algorithm schedules the longest connected SS queue to transmit during the first half of the frame, then route the transmitted packet to the shortest RS queue, update the RS queue sizes and lastly schedule the longest connected RS queue to transmit during the second half of the frame. We also prove that this algorithm results in a MB policy.

Our literature survey revealed several proposed algorithms for packet scheduling in WRN. Many of the scheduling policies resulted from these algorithms may outperform other existing policies under some operating conditions but they are not optimal. The theoretical as well as practical value of the optimal packet scheduling policy is what motivated this work.

The rest of this paper is organized as follows; In Section II, we present a detailed description of the queueing model for the relay network under investigation. In Section III, we define the “Most Balancing” policy for the presented model. In Section IV, the optimal packet scheduling policy is investigated and the optimality results are presented. Conclusions are given in Section V.

2 Model Description

We model this network using a discrete-time queueing system¹ as shown in Figure 1. The objective is to find the optimal dynamic packet scheduling policy (or equivalently, the channel allocation decision at every time slot) for this network. The optimal policy is the one that minimizes a cost function of the system queue lengths.

In this model, time is slotted into constant intervals each of which is equal to one transmission frame. Let $\mathcal{T} = \{1, 2, \dots\}$ denote the set of decision intervals (a time slot is equal to one decision interval). 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 (denote that action 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.

A scheduled packet completes transmission with probability 1 at a given time frame and therefore no retransmission is required. This is a simplifying assumption, however, the results obtained for this model can be easily adapted for a system with retransmission. 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 central-

¹In this model, the direct link between a subscriber station and the destination (the base station) can be considered as a degenerate case.

ized control provided by the base station. They deploy a dedicated control channel that can be used to communicate such information.

2.1 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 $\mathcal{L} = \{0, 1, \dots, L\}$ (respectively $\mathcal{K} = \{0, 1, \dots, K\}$) be the set of indices for the SS (respectively RS) stack of queues

For any time slot $t \in \mathcal{T}$, we define the following:

- $\mathbf{X}(t) = (X_0(t), X_1(t), \dots, X_L(t))$ is the queue length vector for SS nodes (measured in number of packets) at the beginning of time slot t , where $X_i(t) \in \{0, 1, 2, \dots\}$ and $X_0(1) = 0$, i.e., we assume that the dummy queue is initially empty.
- $\mathbf{Y}(t) = (Y_0(t), Y_1(t), \dots, Y_L(t))$ is the queue length vector for RS nodes at the beginning of time slot t , where $Y_i(t) \in \{0, 1, 2, \dots\}$. We assume that $Y_0(t) = 0$ for all t .
- $\mathbf{A}(t) = (A_0(t), A_1(t), \dots, A_L(t))$, where $A_i(t)$ is the number of exogenous arrivals to SS queue i during time slot t .
- $\mathbf{C}^s(t) = (C_0^s(t), C_1^s(t), \dots, C_L^s(t))$ (resp. $\mathbf{C}^r(t) = (C_0^r(t), C_1^r(t), \dots, C_K^r(t))$) is the channel connectivity random variable 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. $U_1(t) \in \mathcal{L}, U_2(t) \in \mathcal{K}, U_3(t) \in \mathcal{K}$, 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” at the beginning of time slot t that we denote by $\mathbf{S}(t)$.

We make the following statistical assumptions regarding the random processes in the system. The arrival processes $(A_i(t), i = 1, \dots, L)$ are assumed to be i.i.d. Bernoulli, with parameter q , i.e., packet arrival to SS queue i is given by:

$$A_i(t) = \begin{cases} 1 & \text{with probability } q, \\ 0 & \text{with probability } 1 - q \end{cases}$$

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. In either case, arrivals during time slot t will be added only after the removal of the scheduled packets at 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 the channel is idled (i.e., when there is no real packet transmission).

The connectivity processes $C_i^s(t)$ and $C_j^r(t)$, for all $i = 1, \dots, L$ and $j = 1, \dots, K$ 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.

We define next the ‘withdrawal’ and the ‘insertion’ controls as a function of the scheduler control $\mathbf{U}(t)$. These controls simplify the formulation and the proof of the optimality problem we are studying in this article.

2.2 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:

$$W_i^s(t) = \mathbb{1}_{\{U_1(t)=i\}}, \quad i = 0, 1, \dots, L, \quad (1)$$

and,

$$W_j^r(t) = \mathbb{1}_{\{U_3(t)=j\}}, \quad j = 0, 1, \dots, K, \quad (2)$$

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 .

Similarly, we define the RS *insertion vector* as follows:

$$V_j^r(t) = \mathbb{1}_{\{U_2(t)=j\}}, \quad j = 1, \dots, K, \quad U_2(t) > 0 \quad (3)$$

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 in to 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:

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

Exogenous arrivals (in Equation (4)) can only be added after the scheduling control is executed and the scheduled packet is withdrawn.

It is apparent, from Equations (4) and (5) that Equations (1) – (3) do not guarantee feasibility of the withdrawal/insertion 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*’. A formal statement of this condition is given next.

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:

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

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

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

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 slot t .

Define the set $\mathcal{U}(\mathbf{S}(t))$ as the set of all feasible scheduling controls when the system in state $\mathbf{S}(t)$. In the following subsection, we formally define packet scheduling policies for the queueing system under investigation.

2.3 Policies for Dynamic Packet Scheduling

A *packet scheduling policy* π (or policy π 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 sequence of random variables

$$\begin{aligned} \mathbf{H}(1) &= (\mathbf{X}(1), \mathbf{Y}(1)), \quad \text{and} \\ \mathbf{H}(t) &= (\mathbf{X}(1), \mathbf{Y}(1), \mathbf{C}^s(1), \mathbf{C}^r(1), \mathbf{A}(1), \dots, \mathbf{C}^s(t-1), \\ &\quad \mathbf{C}^r(t-1), \mathbf{A}(t-1), \mathbf{C}^s(t), \mathbf{C}^r(t)), \quad n = 2, 3, \dots \end{aligned} \quad (9)$$

Note that only the initial queue size value is included in the history. The queue size for the subsequent time slots can be easily determined for a given initial state and a given policy when the values of the arrival and connectivity processes are known at all preceding time slot.

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

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

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

The set of feasible² scheduling policies described in Equation (10) is denoted by Π . We are interested in a subset of Π that we will introduce in the next section, 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 Π .

3 The Class of MB Policies (Π^{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 more formal characterization of MB policies. We first define the ‘*imbalance index*’ of a vector \mathbf{x} that we denote by $\kappa(\mathbf{x})$.

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

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

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

The above definition ensures that the differences are nonnegative and a pair of components is accounted for in the summation only once; moreover, as we shall see in Lemma 2, this

²We say that a policy π is feasible if it selects a feasible scheduling control $\mathbf{U}^\pi(t) \in \mathcal{U}(\mathbf{S}(t))$ for all t .

definition allows for a straightforward calculation and comparison of various policies. 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

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

□

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, i.e., make the queue sizes more balanced.

Given a state $\mathbf{s}(t)$ and a policy π 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 size 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 $z_0(t) = w_0^s(t)$, $\hat{x}_0(t)$ may be negative. The updated queue sizes can be stated as follows:

$$\hat{x}_i(t) = x_i(t) - w_i^s(t) \quad i = 0, 1, \dots, L \quad (13)$$

and,

$$\hat{y}_j(t) = y_j(t) + v_j^r(t) - w_j^r(t) \quad j = 0, 1, \dots, K \quad (14)$$

At any given time slot t , the imbalance indices for the updated SS and RS queue length vectors $\hat{\mathbf{x}}(t) \in \mathcal{Z}_+^{L+1}$ and $\hat{\mathbf{y}}(t) \in \mathcal{Z}_+^{K+1}$ are given by $\kappa(\hat{\mathbf{x}}(t))$ and $\kappa(\hat{\mathbf{y}}(t))$. The $L + 1^{\text{st}}$ SS queue as well as the $K + 1^{\text{st}}$ RS queue are the dummy queues defined in the previous section. By convention, the SS (respectively the RS) “dummy queue”, i.e., queue 0 in either stack, will always have order $L + 1$ (respectively $K + 1$), i.e., the queue with the minimum length in the stack.

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 $M \cdot x_{[M]}$ (i.e., the first M components have the same value which is equal to that of the smallest component) which is indicative of a fully balanced system. It also follows that the maximum such value is equal to $2(M - 1)x_{[1]} - (M - 2)x_{[M]}$. This value is attained when the $M - 1$ largest components have the same size.

We are ready to define the set of MB policies. We denote by Π^{MB} the set of all MB policies in the system we described earlier. We define the elements of Π^{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.,

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

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.

The set Π^{MB} in Equation (15) is well-defined and non-empty, since the minimization is over a finite set of controls. It is also possible that the set of MB policies may have more than one element. This could happen, for example, when at a given time slot t , one of two (or more) equal size queues is scheduled to transmit during that time slot under an MB policy $\pi_1 \in \Pi^{MB}$. Then, a policy π_2 that chooses the other equal size queue will result in the same imbalance index as π_1 during that time slot. Therefore, $\pi_2 \in \Pi^{MB}$.

3.1 MB Policy Implementation

Given $\mathbf{s}(t)$, one can construct an MB policy using a direct search over all possible scheduling control vectors. For large L and K , this can be a challenging computational task. We provide

next 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 $\mathbf{u}(t) \in \mathcal{U}(\mathbf{s}(t))$ such that $u_1(t)$ is the longest connected SS queue (LCQ), $u_2(t)$ is the shortest RS queue (SQ), and $u_3(t)$ is the longest connected RS queue (LCQ). That is

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

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

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

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 in the extreme case, 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, e.g., when all connected SS queues are empty. \square

Equations (16) – (18) describe the control vector $\mathbf{u}(t) = (u_1(t), u_2(t), u_3(t))$. Equation (16) identifies the longest connected SS queue as $u_1(t)$. 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 SS queue as $u_3(t)$. Equation (18) identifies the longest connected RS queue, after adding the packet received from SS nodes during the first half of time slot t , as $u_3(t)$.

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 $\mathbf{u}(t)$. \square

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 2 and shown in Figure 1, a MB policy can be constructed using Algorithm LCQ/SQ/LCQ.*

To prove Theorem 1, we will need the following lemma. It quantifies the effect of performing a balancing interchange on the imbalance index $\kappa(\mathbf{x})$ of the $L + 1$ -dimensional vector \mathbf{x} .

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 between 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*

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

According to Equation (11), to prove Lemma 2 we have to show that

$$\begin{aligned} \sum_{i'=1}^L \sum_{j'=i'+1}^{L+1} (x_{i'}^* - x_{j'}^*) &= \sum_{i=1}^L \sum_{j=i+1}^{L+1} (x_i - x_j) \\ &\quad - 2(s - l) \cdot \mathbf{1}_{\{x_l \geq x_{s+2}\}} \end{aligned} \quad (20)$$

The full proof of Lemma 2 is presented in Appendix Appendix .1. Now we are ready to prove Theorem 1.

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 . During this time slot, 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)).

We show next that the control vector selected by Algorithm LCQ/SQ/LCQ during time slot t will result in an imbalance index that is either (i) *less* or (ii) *equal* to that of 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.

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 the contradiction, we assume (to the contrary of Theorem 1) that $\mathbf{u}(t)$ does not satisfy Equations (16) – (18). In this case, consider the following three cases corresponding to Equations (16), (17) and (18) respectively:

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 π). 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, π 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 $y_{u_2(t)}(t) + v_{u_2(t)}^r(t) > y_{s^r}(t) + v_{s^r}^r(t) + 1$. 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, π does not satisfy Equation (15) and hence it is not a MB policy. This contradiction leads us to conclude that π 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 $c_{u_2(t)}^r(t) = c_{s^r}^r(t)$ then $u_2(t)$ satisfies Equation (17) during time slot t . Otherwise, i.e., $c_{s^r}^r(t) > c_{u_2(t)}^r(t) = 0$, then $y_{u_2(t)}(t) + v_{u_2(t)}^r(t) = y_{s^r}(t) + v_{s^r}^r(t) + 1$. 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, π does not satisfy Equation (15) and hence it is not a MB policy. By contradiction π must satisfy Equation (17).

If on the other hand $u_3(t) \neq s^r$ then a policy that choses 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) $y_{u_3(t)}(t) + v_{u_3(t)}^r(t) < y_{l^r}(t) + v_{l^r}^r(t)$, 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, π does not satisfy Equation (15) and hence it is not a MB policy. This contradiction leads us to conclude that a MB policy π must satisfy Equation (18). Since l^r is the longest connected queue by assumption and given the feasibility constraint (7), the case where $y_{u_3(t)}(t) + v_{u_3(t)}^r(t) > y_{l^r}(t) + v_{l^r}^r(t)$ 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. \square

4 Optimality of MB Policies

In this section, we provide a proof for the optimality of the Most Balancing (MB) policies. The optimal policy minimizes (stochastically) a class of cost functions that correspond to a range performance criteria, including the minimization of the total number of packets in the system. We use coupling arguments in our proof. 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 investigate in this section.

4.1 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 defined in Equation (12).

Then the “preferred order” is defined as follows:

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

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 [26]. Intuitively, $\tilde{\mathbf{x}} \preceq \mathbf{x}$ if the vector $\tilde{\mathbf{x}}$ is obtained from \mathbf{x} by performing a sequence of operations including: *reductions*, *permutations of two components* and/or *balancing interchanges*.

4.2 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 \preceq . For any two vectors $\tilde{\mathbf{x}}, \mathbf{x} \in \mathcal{Z}_+^M$, a function f belongs to the class \mathcal{F} if and only if

$$\tilde{\mathbf{x}} \preceq \mathbf{x} \Rightarrow f(\tilde{\mathbf{x}}) \leq f(\mathbf{x}). \quad (21)$$

Using (21) and the definition of preferred order, we conclude that the function $f(\mathbf{x}) = x_1 + x_2 + \dots + x_M$, i.e., the total sum of all components in \mathbf{x} , 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.

4.3 The Optimality Results

Let $B \leq_{st} C$ defines the usual stochastic ordering. For two real-valued random variables B and C [19]. For the rest of this article, we say that a policy $\sigma \in \Pi$ *dominates* another policy $\pi \in \Pi$ if

$$f(\mathbf{X}^\sigma(t)) \leq_{st} f(\mathbf{X}^\pi(t)), \quad \text{AND} \quad (22)$$

$$f(\mathbf{Y}^\sigma(t)) \leq_{st} f(\mathbf{Y}^\pi(t)), \quad \forall t = 1, 2, \dots, \quad (23)$$

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

Note that from Equation (21) and the definition of stochastic ordering, $\mathbf{X}^\sigma(t) \preceq \mathbf{X}^\pi(t)$ and $\mathbf{Y}^\sigma(t) \preceq \mathbf{Y}^\pi(t)$, for all t and all sample paths in a suitable sample space, is sufficient to insure policy domination. The intended sample space is the standard one used in stochastic coupling [20]; refer to Appendix Appendix .2 for more details.

In what follows, let \mathbf{X}^{MB} and \mathbf{X}^π (respectively \mathbf{Y}^{MB} and \mathbf{Y}^π) 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 Π of all feasible scheduling policies:

1. $\Pi_\tau \in \Pi$, the set of policies that has the MB property during slots $t \leq \tau$, and are arbitrary for $t > \tau$.
2. $\Pi_\tau^{u_2} \in \Pi$, the set of policies that has the MB property during time slots $t \leq \tau - 1$ and during $t = \tau$ chooses the same actions $u_1(\tau)$ and $u_2(\tau)$ as those selected by a MB policy and an arbitrary $u_3(\tau)$. Note that for policies in this set, $u_3(\tau)$ may not be a most balancing action.

From the above definitions we have $\Pi = \Pi_0$. Note that the set Π_n for any $t = n$ is not empty, since MB policies are elements of it. For $n = 0, 1, \dots$, Π_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 Π^{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 the policy $\pi \in \Pi_{\tau-1}$, a policy $\tilde{\pi} \in \Pi_\tau^{u_2}$ can be constructed, such that $\tilde{\pi}$ dominates π .*

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

We present the proof for Lemmas 3 and 4 in Appendix Appendix .2. The proof has two parts. In the first part, we construct the policy $\tilde{\pi}$ by performing a balancing interchange to π according to Equation (12). In the second part, we prove that $\tilde{\pi}$ dominates π (see Equation (22)); we use stochastic coupling arguments for this part of the proof.

The above lemmas provide a methodology to construct a MB policy from any arbitrary policy π 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 of Lemma 3 is necessary in order to simplify the coupling arguments used in the proof of the lemma.

We are ready to present the main result of this work, i.e., the optimality of MB policies.

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

$$f(\mathbf{X}^{MB}(t)) \leq_{st} f(\mathbf{X}^{\pi}(t)), \quad \text{AND} \quad (24)$$

$$f(\mathbf{Y}^{MB}(t)) \leq_{st} f(\mathbf{Y}^{\pi}(t)), \quad \forall t = 1, 2, \dots, \quad (25)$$

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

Proof. Starting from an arbitrary policy π , we apply a series of modifications to π , using Lemmas 3 and lem:7, that result in a sequence of policies (π_1, π_2, \dots) , such that:

- (i) policy π_1 dominates the original arbitrary policy π ,
- (ii) $\pi_n \in \Pi_n$, in other words, policy π_n has the MB property during time slots $t = 1, 2, \dots, n$, and,
- (iii) π_m dominates π_n for $m > n$ (i.e., when π_m has the MB property for a period of time $m - n$ slots longer than π_n).

By definition, π is an arbitrary policy; therefore, $\pi \in \Pi_0$. We construct a policy $\tilde{\pi} \in \Pi_1^{u2}$ that dominates π 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 π . Repeating the construction

steps above and using Lemmas 3 and lem:7 again for time slots $t = 2, 3, \dots$ will result in a sequence of policies $\pi_n \in \Pi_n, n = 2, 3, \dots$ 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 π .

Lets denote the limiting policy for the sequence of constructed policies as $n \rightarrow \infty$ by π^* . 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 π^* dominates π_n , for all $n < \infty$ including the original policy π . The theorem follows since the initial policy $\pi \in \Pi$ is an arbitrary policy by assumption. \square

5 Conclusion

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 queueing system. The results presented in this article provide a concrete understanding of the optimal scheduling policy structure in homogeneous wireless relay networks as well as other multi-hop wireless network.

Appendix .1 Proof for Lemma 2 in Section 3.1

In this section, we present the full proof for Lemma 2. This lemma quantifies the effect of performing a balancing interchange on the imbalance index $\kappa(\mathbf{x})$ of a vector \mathbf{x} .

Proof for Lemma 2. To prove this lemma, we first show that:

$$\sum_{i'=1}^L \sum_{j'=i'+1}^{L+1} (x_{i'}^* - x_{j'}^*) = \sum_{i=1}^L \sum_{j=i+1}^{L+1} (x_i - x_j) - 2(s-l) \cdot \mathbf{1}_{\{x_l \geq x_s + 2\}} \quad (\text{A-1})$$

Then according to Equation (11), the above is equivalent to Equation (19) and Lemma 2 follows.

We generate the vector \mathbf{x}^* by performing a *balancing interchange* of two components, l and s (i.e., the l^{th} and the s^{th} largest components), in the vector \mathbf{x} and reorder the resulted vector in descending manner. The resulted vector \mathbf{x}^* is characterized by the following:

$$\begin{aligned} x_{l'}^* &= x_l - 1, & x_{s'}^* &= x_s + 1, & x_l &> x_s \\ x_k^* &= x_k, & \forall k &\neq l, s, l', s' \end{aligned} \quad (\text{A-2})$$

where l' (respectively s') is the new index (i.e., the order in the new vector \mathbf{x}^*) of component l (respectively s) in the original vector \mathbf{x} .

From Equation (A-2) we can identify $L-2$ elements that have the same magnitude in the two vectors \mathbf{x} and \mathbf{x}^* . Therefore, the sum of differences between these $L-2$ elements in both vectors will also be the same, i.e.,

$$\sum_{\substack{i'=1 \\ i' \notin \{l', s'\}}}^L \sum_{\substack{j'=i'+1 \\ j' \notin \{l', s'\}}}^{L+1} (x_{i'}^* - x_{j'}^*) = \sum_{\substack{i=1 \\ i \notin \{l, s\}}}^L \sum_{\substack{j=i+1 \\ j \notin \{l, s\}}}^{L+1} (x_i - x_j) \quad (\text{A-3})$$

We calculate the sums for the remaining terms (i.e., when at least one of the indices i, j belongs to $\{l, s\}$ and/or i', j' belongs to $\{l', s'\}$) next. We first assume that $x_l \geq x_s + 2$; in this case, we can easily show that $l' \leq s'$. Then, we have the following five, mutually exclusive, cases to consider:

1. When $i' = l', i = l, j' = s'$ and $j = s$. This case occurs only once, i.e., when decomposing the double sum in Equation (A-1) we can find only one term that satisfies this case. From Equation (A-2) we have

$$x_{l'}^* - x_{s'}^* = x_l - x_s - 2 \quad (\text{A-4})$$

2. When $i' = l', i = l, j' \neq s'$ and $j \neq s$. There are $L - l$ terms that satisfy this case. Analogous to case 1) we determined that

$$x_{l'}^* - x_{j'}^* = x_l - x_j - 1 \quad (\text{A-5})$$

3. When $i' \neq l', i \neq l, j' = s'$ and $j = s$. There are $s - 2$ terms that satisfy this case. In this case we can show that

$$x_{i'}^* - x_{s'}^* = x_i - x_s - 1 \quad (\text{A-6})$$

4. When $i' \neq l', s', i \neq l, s, j' = l'$ and $j = l$. There are $l - 1$ terms that satisfy this case. In this case we can show that

$$x_{i'}^* - x_{l'}^* = x_i - x_l + 1 \quad (\text{A-7})$$

5. When $i' = s', i = s, j' \neq l', s'$ and $j \neq l, s$. There are $L - s + 1$ terms that satisfy this case. In this case we have

$$x_{s'}^* - x_{j'}^* = x_s - x_j + 1 \quad (\text{A-8})$$

The above cases (i.e., Equations (A-3)-(A-8)) cover all the terms in Equation (A-1) when $x_l \geq x_s + 2$. Combining all these terms yields:

$$\begin{aligned} \sum_{i'=1}^L \sum_{j'=i'+1}^{L+1} (x_{i'}^* - x_{j'}^*) &= \sum_{i=1}^L \sum_{j=i+1}^{L+1} (x_i - x_j) \\ &- 2 \cdot (1) - 1 \cdot (L-l) - 1 \cdot (s-2) + 1 \cdot (l-1) + 1 \cdot (L-s+1) \\ &= \sum_{i=1}^L \sum_{j=i+1}^{L+1} (x_i - x_j) - 2(s-l) \end{aligned} \quad (\text{A-9})$$

Furthermore, if $x_l = x_s + 1$, then from Equation (A-2) it is clear that $x_{l'}^* = x_s$ and $x_{s'}^* = x_l$, i.e., the resulted vector is a permutation of the original one. Therefore, the sum of differences will be the same in both vectors and Equation (A-1) will be reduced to

$$\sum_{i'=1}^L \sum_{j'=i'+1}^{L+1} (x_{i'}^* - x_{j'}^*) = \sum_{i=1}^L \sum_{j=i+1}^{L+1} (x_i - x_j) \quad (\text{A-10})$$

Equation (A-1) follows from Equations (A-9) and (A-10). \square

Appendix .2 Proofs for Lemmas 3 and 4

To prove Lemmas 3 and 4 we use stochastic coupling arguments. We start by introducing the coupling method briefly.

Appendix .2.1 Stochastic Coupling Method

In order to compare probability measures on a measurable space, it is often possible to construct random elements on a common probability space with these measures as their distributions, such that this comparison can be conducted in terms of these random elements rather than the probability measures. Such construction is often referred to as *stochastic coupling* [20]. In the notation of [20], a formal definition of coupling of two probability measures on the measurable space (E, \mathcal{E}) (the state space, e.g., $E = \mathcal{R}, \mathcal{R}^d, \mathcal{Z}_+, \text{etc.}$) is given below; further details regarding coupling method and its application can be found in [20].

A random element in (E, \mathcal{E}) is a quadruple $(\Omega, \mathcal{F}, \mathbf{P}, X)$, where $(\Omega, \mathcal{F}, \mathbf{P})$ is the sample space and X is the class of measurable mappings from Ω to E (X is an E -valued random variable, s.t. $X^{-1}(B) \in \mathcal{F}$ for all $B \in \mathcal{E}$).

Definition: A coupling of the two random elements $(\Omega, \mathcal{F}, \mathbf{P}, \mathbf{X})$ and $(\Omega', \mathcal{F}', \mathbf{P}', \mathbf{X}')$ in (E, \mathcal{E}) is a random element $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{\mathbf{P}}, (\hat{\mathbf{X}}, \hat{\mathbf{X}}'))$ in (E^2, \mathcal{E}^2) such that

$$\mathbf{X} \stackrel{\mathcal{D}}{=} \hat{\mathbf{X}} \quad \text{and} \quad \mathbf{X}' \stackrel{\mathcal{D}}{=} \hat{\mathbf{X}}', \quad (\text{B-1})$$

where $\stackrel{\mathcal{D}}{=}$ denotes 'equal in distribution'. □

Stochastic coupling was initially used by mathematicians to study properties of stochastic processes (e.g., in renewal theory). Later on, coupling methods proved to be handy in proving optimality results for dynamic control of queueing systems. c.f. [27], [28], [21], [24] and many others.

In the proof of our results (Lemmas 3 and 4), we apply the coupling method as follows: For the scheduling policy π , let ω be a given sample path of the system state process. Note that the values of the sequences $\{\mathbf{X}(n)\}, \{\mathbf{Y}(n)\}$ and $\mathbf{U}(n)$ can be completely determined by ω and π . In our proof, we construct a new sample path, $\tilde{\omega}$ and a new policy $\tilde{\pi}$. The details of this construction is given in the proof of the two lemmas. To put things into perspective, in

the coupling definition (Equation (B-1)), $\hat{\omega} = (\omega, \tilde{\omega})$ and the ‘‘coupled’’ processes of interest in Equation (B-1) will be the SS queue sizes $\hat{\mathbf{X}} = \{\mathbf{X}(n)\}$ and $\hat{\mathbf{X}}' = \{\tilde{\mathbf{X}}(n)\}$ as well as the RS queue sizes $\hat{\mathbf{Y}} = \{\mathbf{Y}(n)\}$ and $\hat{\mathbf{Y}}' = \{\tilde{\mathbf{Y}}(n)\}$.

Appendix .2.2 Proof for the Results of Section 4

In this section, we provide a detailed proof for Lemmas 3 and 4. lets describe the construction process for $\tilde{\omega}$ and $\tilde{\pi}$ first.

Define ω to be the sequence of sample values of the random variables $(\mathbf{S}(1), \mathbf{A}(1), \mathbf{C}^s(2), \mathbf{C}^r(2), \mathbf{A}(2), \dots)$, i.e., $\omega \equiv (\mathbf{x}(1), \mathbf{y}(1), \mathbf{c}^s(1), \mathbf{c}^r(1), \mathbf{a}(1), \mathbf{c}^s(2), \mathbf{c}^r(2), \mathbf{a}(2), \dots)$. Let $\tilde{\omega}$ denote a sample path of the system random variables under policy $\tilde{\pi}$. Then we construct $\tilde{\omega}$ such that:

- 1) $\tilde{\mathbf{x}}(1) = \mathbf{x}(1)$ and $\tilde{\mathbf{y}}(1) = \mathbf{y}(1)$, i.e., both systems start from the same initial queue sizes.
- 2) During $t = \tau$, $\tilde{\mathbf{c}}^s(n)$ is the same as $\mathbf{c}^s(n)$, except for two elements that are exchanged.

We do the same for the other connectivity and arrival sample values.

Which elements to be exchanged will be detailed in the proof that follows. The resulted random elements from this exchange have the same distribution as their original counterparts, since the distributions of the connectivities as well as the arrival processes are i.i.d. by assumption, reordering their elements will not change their distribution.

The scheduling policy selects three control elements at every time slot, namely $u_1(t), u_2(t)$ and $u_3(t)$. The detailed construction of policy $\tilde{\pi}$ will be described in the proof. Using Equations (4) and (5), We can compute the new queue states $\mathbf{x}(\cdot), \mathbf{y}(\cdot)$ under π and $\tilde{\mathbf{x}}(\cdot), \tilde{\mathbf{y}}(\cdot)$ under $\tilde{\pi}$. Our goal is to prove that the two relations

$$\tilde{\mathbf{x}}(t) \preceq \mathbf{x}(t) \tag{B-2}$$

$$\tilde{\mathbf{y}}(t) \preceq \mathbf{y}(t) \tag{B-3}$$

are satisfied for all t . This will insure the dominance of policy $\tilde{\pi}$ over π .

A queue length vector $\tilde{\mathbf{x}}$ is preferred over \mathbf{x} (i.e., $\tilde{\mathbf{x}} \preceq \mathbf{x}$) if and only if one of the statements S1, S2 or S3 (that we introduced in Section 4.1) holds.

Proof for Lemma 3. To prove this lemma, we start from an arbitrary policy $\pi \in \Pi_{\tau-1}$ and a sample path $\omega = (\mathbf{x}(1), \mathbf{y}(1), \mathbf{c}^s(1), \mathbf{c}^r(1), \mathbf{a}(1), \dots)$, where $\mathbf{x}(\cdot), \mathbf{y}(\cdot), \mathbf{c}^s(\cdot), \mathbf{c}^r(\cdot)$ and $\mathbf{a}(\cdot)$ are

sample values of the random variables $\mathbf{X}(\cdot)$, $\mathbf{Y}(\cdot)$, $\mathbf{C}^s(\cdot)$, $\mathbf{C}^r(\cdot)$ and $\mathbf{A}(\cdot)$. For the remainder of the proof, we refer to the three statements S1, S2 and S3, that we defined in Section 4.1 as properties of the partial order (\preceq). If one of these properties is satisfied then the partial order holds.

The proof proceeds in two parts; In part 1, we construct the sample path $\tilde{\omega}$ and the policy $\tilde{\pi}$ (as stated by Lemma 3) for times up to $t = \tau$. In part 2, we do the same for $t > \tau$.

Part 1: For time $t < \tau$ we construct $\tilde{\omega}$ to coincide with ω , i.e., $\tilde{\mathbf{a}}(t) = \mathbf{a}(t)$, $\tilde{\mathbf{c}}^s(t) = \mathbf{c}^s(t)$ and $\tilde{\mathbf{c}}^r(t) = \mathbf{c}^r(t)$ for all $t < \tau$. We construct $\tilde{\pi}$ such that $\tilde{\mathbf{u}}(t) = \mathbf{u}(t)$ for all $t < \tau$. In this case, the resulting queue lengths under both policies at $t = \tau$ are the same, i.e., $\tilde{\mathbf{x}}(\tau) = \mathbf{x}(\tau)$ and $\tilde{\mathbf{y}}(\tau) = \mathbf{y}(\tau)$.

At time slot $t = \tau$, let $\tilde{\omega}$ have the same channel connectivities and arrivals as ω , i.e., let $\tilde{\mathbf{c}}^s(\tau) = \mathbf{c}^s(\tau)$, $\tilde{\mathbf{c}}^r(\tau) = \mathbf{c}^r(\tau)$ and $\tilde{\mathbf{a}}(\tau) = \mathbf{a}(\tau)$. Then we construct $\tilde{\pi}$ at $t = \tau$ by selecting the sequence of controls $\tilde{u}_1(\tau)$, $\tilde{u}_2(\tau)$ and $\tilde{u}_3(\tau)$ as follows:

1- *Construction of $\tilde{u}_1(\tau)$.* We have the following two cases to consider:

(i) The scheduling control $\mathbf{u}(t)$ satisfies Equation (16) at $t = \tau$, i.e.,

$$u_1(\tau) = l^s : l^s \in \underset{i \in \mathcal{L}: c_i^s(\tau)=1}{\operatorname{argmax}} x_i(\tau).$$

Then we set $\tilde{u}_1(\tau) = u_1(\tau)$. Note that $\tilde{u}_1(\tau)$ and $u_1(\tau)$ affect the SS queue lengths only and have no effect of the RS queue sizes. It follows that the resulting SS queue lengths $\tilde{\mathbf{x}}(\tau + 1) = \mathbf{x}(\tau + 1)$ ($\tilde{a}(\tau) = a(\tau)$ by construction), property (S1) holds true for the SS queue length vector and (B-2) is satisfied at $t = \tau + 1$.

(ii) The scheduling control $\mathbf{u}(t)$ does not satisfy Equation (16) at $t = \tau$. Then we set

$$\tilde{u}_1(\tau) = l^s : l^s \in \underset{i \in \mathcal{L}: \tilde{c}_i^s(\tau)=1}{\operatorname{argmax}} \tilde{x}_i(\tau).$$

Keeping the construction of $\tilde{\omega}$ in mind, we conclude the following (we suppress the time argument for the subscript to simplify notation):

$$\begin{aligned} \hat{\tilde{x}}_{\tilde{u}_1}(\tau) &= \hat{x}_{\tilde{u}_1}(\tau) - 1, & \hat{\tilde{x}}_{u_1}(\tau) &= \hat{x}_{u_1}(\tau) + 1, \\ \text{where } \hat{x}_{\tilde{u}_1}(\tau) &> \hat{x}_{u_1}(\tau) \end{aligned} \tag{B-4}$$

From Equation (B-4) and the construction of the exogenous arrivals (i.e., we set $\tilde{a}(\tau) = a(\tau)$) we conclude that property (S3) holds true for the SS queue length vector (i.e., $\tilde{\mathbf{x}}(\tau + 1)$ is obtained from $\mathbf{x}(\tau + 1)$ by performing a balancing interchange of two components $\tilde{u}_1(\tau)$ and $u_1(\tau)$) and (B-2) is satisfied at $t = \tau + 1$.

2- *Construction of $\tilde{u}_2(\tau)$ and $\tilde{u}_3(\tau)$.* We have the following two cases to consider:

(i) The scheduling control $\mathbf{u}(t)$ satisfies Equation (17) at $t = \tau$, i.e.,

$$u_2(\tau) = s^r : s^r \in \operatorname{argmax}_{i \in \mathcal{I}} c_i^r(\tau),$$

$$\text{where } \mathcal{I} = \operatorname{argmin}_{j \in \{1, \dots, K\}} y_j(\tau).$$

Then we set $\tilde{u}_2(\tau) = u_2(\tau)$ and $\tilde{u}_3(\tau) = u_3(\tau)$. The resulting RS queue sizes $\tilde{\mathbf{y}}(\tau + 1) = \mathbf{y}(\tau + 1)$. Property (S1) holds true for the RS queue length vector and (B-3) is satisfied at $t = \tau + 1$.

(ii) The scheduling control $\mathbf{u}(t)$ does not satisfy Equation (17) at $t = \tau$. Then we set

$$\tilde{u}_2(\tau) = s^r : s^r \in \operatorname{argmax}_{i \in \mathcal{I}} c_i^r(\tau),$$

where the set \mathcal{I} is defined in case (i) above. We also set $\tilde{u}_3(\tau) = u_3(\tau)$. In this case, the RS queue lengths satisfy (for all feasible selections of $u_3(\tau)$) the following:

$$\begin{aligned} \tilde{y}_{\tilde{u}_2}(\tau + 1) &= y_{\tilde{u}_2}(\tau + 1) + 1, & \tilde{y}_{u_2}(\tau + 1) &= y_{u_2}(\tau + 1) - 1, \\ \text{where } y_{\tilde{u}_2}(\tau + 1) &< y_{u_2}(\tau + 1). \end{aligned} \tag{B-5}$$

Equation (B-5) suggests that $\tilde{\mathbf{y}}(\tau + 1)$ is obtained from $\mathbf{y}(\tau + 1)$ by performing a balancing interchange of two components $\tilde{u}_2(\tau)$ and $u_2(\tau)$. In this case, property (S3) holds true for the RS queue length vector and (B-3) is satisfied at $t = \tau + 1$.

In cases (1-) and (2-) above, we constructed the policy $\tilde{\pi}$ for time slot $t = \tau$. We also showed that Relations (B-2) and (B-3) are satisfied for time slot $t = \tau + 1$. The above concluded the construction of policy $\tilde{\pi}$ upto time slot $t = \tau$. Next (in part 2), we will construct $\tilde{\pi}$ for time slots $t > \tau$. Furthermore, starting from a preferred state at $t = \tau + 1$, we will show using forward induction that Relations (B-2) and (B-3) are satisfied for all time slot $t > \tau$.

Part 2: We use induction to complete part 2 of our proof. The sample path $\tilde{\omega}$ and the policy $\tilde{\pi}$ are already defined for $t \leq \tau$. To complete the induction argument, we assume that $\tilde{\pi}$ and

$\tilde{\omega}$ are defined up to time $n - 1 \geq \tau$ and that relations (B-2) and (B-3) are satisfied at $t = n$, i.e., $\tilde{\mathbf{x}}(n) \preceq \mathbf{x}(n)$ and $\tilde{\mathbf{y}}(n) \preceq \mathbf{y}(n)$. We will show that at time slot n , $\tilde{\pi}$ can be constructed such that relations (B-2) and (B-3) are satisfied at $t = n + 1$. To do that, we have to show that either (S1), (S2) or (S3) holds for both $\mathbf{x}(t)$ and $\mathbf{y}(t)$ at time slot $t = n + 1$.

We consider next three cases that correspond to properties (S1), (S2) and (S3) of the vector $\mathbf{x}(n)$. For each one of these cases, we consider three sub-cases that correspond to properties (S1), (S2) and (S3) of the vector $\mathbf{y}(n)$.

1- $\tilde{\mathbf{x}}(n) \leq \mathbf{x}(n)$ (i.e., property (S1) holds for $\mathbf{x}(t)$). We set $\tilde{\mathbf{a}}(n) = \mathbf{a}(n)$ and $\tilde{\mathbf{c}}^s(n) = \mathbf{c}^s(n)$. We set $\tilde{u}_1(n) = u_1(n)$. The SS queue lengths satisfy (S1), i.e., $\tilde{\mathbf{x}}(n+1) \leq \mathbf{x}(n+1)$, and (B-2) holds at $t = n + 1$. The controls $\tilde{u}_2(n)$ and $\tilde{u}_3(n)$ are construction below and the RS queue lengths are computed as follows:

(a) $\tilde{\mathbf{y}}(n) \leq \mathbf{y}(n)$ (i.e., property (S1) holds for $\mathbf{y}(t)$). We set $\tilde{\mathbf{c}}^r(n) = \mathbf{c}^r(n)$. We also set the controls $\tilde{u}_2(n) = u_2(n)$ and $\tilde{u}_3(n) = u_3(n)$. In this case, (S1) is satisfied again and Relation (B-3) holds at $t = n + 1$.

(b) $\tilde{\mathbf{y}}(n)$ is a 2-component permutation of $\mathbf{y}(n)$ (i.e., property (S2) holds for $\mathbf{y}(t)$). Let RS queues i and j be the indices of the two permuted queues. Then let $\tilde{c}_i^r(n) = c_j^r(n)$, $\tilde{c}_j^r(n) = c_i^r(n)$ and $\tilde{c}_k^r(n) = c_k^r(n), \forall k \neq i, j$. We construct the control $u_2(n)$ as follows:

$$\tilde{u}_2(n) = \begin{cases} i & \text{if } u_2(n) = j \\ j & \text{if } u_2(n) = i \\ k & \text{if } u_2(n) = k, \forall k \neq i, j \end{cases} \quad (\text{B-6})$$

Similarly, the control $u_3(n)$ is constructed as follows:

$$\tilde{u}_3(n) = \begin{cases} i & \text{if } u_3(n) = j \\ j & \text{if } u_3(n) = i \\ k & \text{if } u_3(n) = k, \forall k \neq i, j \end{cases} \quad (\text{B-7})$$

From the construction of $\tilde{\pi}$, it can be easily shown that property (S2) is satisfied again for RS queues at time $t = n + 1$ and (B-3) follows.

(c) $\tilde{\mathbf{y}}(n)$ is obtained from $\mathbf{y}(n)$ by performing a balancing interchange as described by Equation (12) (i.e., property (S3) holds for $\mathbf{y}(t)$). Let i and j be the indices of the two RS

queues involved in the balancing interchange, such that $y_i(n) \geq y_j(n) + 1$. We consider the following two cases:

(i) $y_i(n) = y_j(n) + 1$. Then $\tilde{y}_i(n) = y_j(n)$ and $\tilde{y}_j(n) = y_i(n)$. This case corresponds to case (1b) above and the same construction of $\tilde{\omega}$ and $\tilde{\pi}$ apply. The resulted queue lengths at $t = n + 1$ will satisfy property (S2) and (B-3) follows.

(ii) $y_i(n) > y_j(n) + 1$. We set $\tilde{\mathbf{c}}^r(n) = \mathbf{c}^r(n)$ and $\tilde{u}_2(n) = u_2(n)$.

If “ $c_j^r(n) = 1, c_k^r(n) = 0, \forall k \neq j$ and $y_j(n) \leq 0$ ”³ (i.e., queue j is the only connected RS queue which happens to be empty), then $u_3(n) = 0$ (i.e., forced idling) according to feasibility constraint (7). Then we set $\tilde{u}_3(n) = j$, which is a feasible control since $\tilde{y}_j(n) = y_j(n) + 1$ according to Equation (12). The resulted RS queue length vector in this case satisfies (S1), i.e., $\tilde{\mathbf{y}}(n + 1) \leq \mathbf{y}(n + 1)$, and (B-3) follows.

Else, i.e., $y_j(n) > 0$ and/or there are other connected queues in the stack, then we set $\tilde{u}_3(n) = u_3(n)$. This action preserve property (S3) and Equation (B-3) hold at $t = n + 1$.

This concludes the construction of $\tilde{\omega}$ and $\tilde{\pi}$ for case (1) during time slot $t = n$.

2- $\tilde{\mathbf{x}}(n)$ is a 2-component permutation of $\mathbf{x}(n)$ (i.e., property (S2) holds for $\mathbf{x}(t)$).

Let i and j be the indices of the two permuted SS queues. Then let $\tilde{c}_i^s(n) = c_j^s(n)$, $\tilde{c}_j^s(n) = c_i^s(n)$ and $\tilde{c}_k^s(n) = c_k^s(n), \forall k \neq i, j$. Similarly, $\tilde{a}_i(n) = a_j(n)$, $\tilde{a}_j(n) = a_i(n)$ and $\tilde{a}_k(n) = a_k(n), \forall k \neq i, j$. We construct the control $u_1(n)$ as follows:

$$\tilde{u}_1(n) = \begin{cases} i & \text{if } u_1(n) = j \\ j & \text{if } u_1(n) = i \\ k & \text{if } u_1(n) = k, \forall k \neq i, j \end{cases} \quad (\text{B-8})$$

From Equation (B-8), it can be easily shown that, at time $t = n + 1$, property (S2) holds again for $\tilde{\mathbf{x}}(n + 1)$ and $\mathbf{x}(n + 1)$, and (B-2) is satisfied. Analogous to case (1-) above, We consider three cases for the construction of $u_2(n)$ and $u_3(n)$ that correspond to (S1), (S2) and (S3) properties of the vector $\mathbf{y}(n)$.

³Note that $j = 0$, the ‘dummy’ queue, is not excluded; therefore, $y_j(n) < 0$ is possible in that particular case.

- a) $\tilde{\mathbf{y}}(n) \leq \mathbf{y}(n)$,
- b) $\tilde{\mathbf{y}}(n)$ is a 2-component (i, j) permutation of $\mathbf{y}(n)$, and
- c) $\tilde{\mathbf{y}}(n)$ is obtained from $\mathbf{y}(n)$ by performing a balancing interchange, of components i and j , as described by Equation (12).

The construction of $\tilde{\omega}$ and $\tilde{\pi}$ in cases (2a), (2b) and (2c) is analogous to that presented in cases (1a), (1b) and (1c) respectively, and the resulted RS queue length vector $\tilde{\mathbf{y}}(n+1)$ satisfies (B-3) at $t = n+1$.

3- $\tilde{\mathbf{x}}(n)$ is obtained from $\mathbf{x}(n)$ by performing a balancing interchange as described by Equation (12) (i.e., property (S3) holds for $\mathbf{x}(t)$). Let i and j be the indices of the two SS queues involved in the balancing interchange, such that $x_i(n) \geq x_j(n) + 1$. We consider the following two cases:

- (i) $x_i(n) = x_j(n) + 1$. Then $\tilde{x}_i(n) = x_j(n)$ and $\tilde{x}_j(n) = x_i(n)$. This case corresponds to case (2-) above and the same construction of $\tilde{\omega}$ and $\tilde{\pi}$ apply. The resulted queue lengths at $t = n+1$ will satisfy property (S2) and (B-3) follows.
- (ii) $x_i(n) > x_j(n) + 1$. We set $\tilde{\mathbf{a}}(n) = \mathbf{a}(n)$ and $\tilde{\mathbf{c}}^s(n) = \mathbf{c}^s(n)$.

If “ $c_j^s(n) = 1, c_k^s(n) = 0, \forall k \neq j$ and $x_j(n) \leq 0$ ”⁴ (i.e., queue j is the only connected SS queue, which happens to be empty), then $u_1(n) = 0$ (i.e., forced idling) according to feasibility constraint (7). Then we set $\tilde{u}_1(n) = j$. This is a feasible control since $\tilde{x}_j(n) = x_j(n) + 1$ according to Equation (12). The resulted SS queue length vector in this case satisfies (S1), i.e., $\tilde{\mathbf{x}}(n+1) \leq \mathbf{x}(n+1)$, and (B-2) is satisfied at $t = n+1$.

Else, i.e., if $x_j(n) > 0$ and/or there are other connected queues in the stack, then we set $\tilde{u}_1(n) = u_1(n)$. This action preserve property (S3) and Equation (B-2) is satisfied at $t = n+1$.

In this case, as with the previous cases, there are three cases to consider for the construction of $u_2(n)$ and $u_3(n)$ which correspond to (S1), (S2) and (S3) properties of the vector $\mathbf{y}(n)$. Again, the construction of $\tilde{\omega}$ and $\tilde{\pi}$ in these cases is analogous to that presented before in

⁴ $x_j(n) < 0$ is possible only when $j = 0$.

cases (1a), (1b) and (1c). The same conclusion regarding the resulted RS queue length vector $\tilde{\mathbf{y}}(n+1)$ is drawn.

This concludes the construction of the policy $\tilde{\pi}$ at $t = n$, for $n > \tau$. We have shown that this policy resulted in queue length vectors $\mathbf{x}(n+1)$ and $\mathbf{y}(n+1)$ that satisfy Equations (B-2) and (B-3) respectively. Note that policy $\tilde{\pi} \in \Pi_\tau^{u_2}$ by construction in Part 1; its dominance over π follows from relation (21). \square

Proof for Lemma 4. The proof of this lemma is analogous to that of Lemma 3. We start from an arbitrary policy $\pi \in \Pi_\tau^{u_2}$ and a sample path $\omega = (\mathbf{x}(1), \mathbf{y}(1), \mathbf{c}^s(1), \mathbf{c}^r(1), \mathbf{a}(1), \dots)$. The proof proceeds in two parts; In part 1, we construct the sample path $\tilde{\omega}$ and the policy $\tilde{\pi}$ (as stated by Lemma 4) for $t \leq \tau$. In part 2, we do the same for $t > \tau$.

Part 1: For time $t < \tau$ we construct $\tilde{\omega}$ to coincide with ω , i.e., $\tilde{\mathbf{a}}(t) = \mathbf{a}(t)$, $\tilde{\mathbf{c}}^s(t) = \mathbf{c}^s(t)$ and $\tilde{\mathbf{c}}^r(t) = \mathbf{c}^r(t)$ for all $t < \tau$. We construct $\tilde{\pi}$ such that $\tilde{\mathbf{u}}(t) = \mathbf{u}(t)$ for all $t < \tau$. In this case, the resulting queue lengths under both policies at $t = \tau$ are the same, i.e., $\tilde{\mathbf{x}}(\tau) = \mathbf{x}(\tau)$ and $\tilde{\mathbf{y}}(\tau) = \mathbf{y}(\tau)$.

At time slot $t = \tau$, let $\tilde{\mathbf{c}}^s(\tau) = \mathbf{c}^s(\tau)$, $\tilde{\mathbf{c}}^r(\tau) = \mathbf{c}^r(\tau)$ and $\tilde{\mathbf{a}}(\tau) = \mathbf{a}(\tau)$. Then for the construction of $\tilde{\pi}$ at $t = \tau$, we set $\tilde{u}_1(\tau) = u_1(\tau)$ and $\tilde{u}_2(\tau) = u_2(\tau)$, where $u_1(\tau)$ and $u_2(\tau)$ are most balancing controls. Then property (S1) holds true for the SS queue length vector and (B-2) is satisfied at $t = \tau + 1$. We construct $\tilde{u}_3(\tau)$ as follows:

- (i) If the scheduling control $\mathbf{u}(t)$ satisfies Equation (18) at $t = \tau$, i.e.,

$$u_3(\tau) = l^r : l^r \in \underset{j \in \mathcal{K}: c_j^r(\tau)=1}{\operatorname{argmax}} (y_j(\tau) + v_j^r(\tau)).$$

Then we set $\tilde{u}_3(\tau) = u_3(\tau)$. Since $\tilde{u}_2(\tau) = u_2(\tau)$ by construction, it follows that the resulting RS queue lengths $\tilde{\mathbf{y}}(\tau+1) = \mathbf{y}(\tau+1)$; property (S1) holds true for the RS queue length vector and (B-3) is satisfied at $t = \tau + 1$.

- (ii) The scheduling control $\mathbf{u}(t)$ does not satisfy Equation (18) at $t = \tau$. Then we set

$$\tilde{u}_3(\tau) = l^r : l^r \in \underset{j \in \mathcal{K}: c_j^r(\tau)=1}{\operatorname{argmax}} (y_j(\tau) + v_j^r(\tau)).$$

The RS queue lengths in this case satisfies the following (we suppress the time argument for

the subscript to simplify notation):

$$\begin{aligned} \tilde{y}_{\tilde{u}_3}(\tau+1) &= y_{\tilde{u}_3}(\tau+1) - 1, & \tilde{y}_{u_3}(\tau+1) &= y_{u_3}(\tau+1) + 1, \\ \text{where } y_{\tilde{u}_3}(\tau+1) &> y_{u_3}(\tau+1) \end{aligned} \tag{B-9}$$

From Equation (B-9) we conclude that property (S3) holds true, i.e., $\tilde{\mathbf{y}}(\tau+1)$ is obtained from $\mathbf{y}(\tau+1)$ by performing a balancing interchange of two components $\tilde{u}_3(\tau)$ and $u_3(\tau)$, and (B-3) is satisfied at $t = \tau + 1$.

The above concludes the construction of policy $\tilde{\pi}$ for time slot $t \leq \tau$. By construction, $\tilde{\pi}$ has the MB property during time slot τ . We showed that Relations (B-2) and (B-3) are satisfied for time slot $t = \tau + 1$. Next (in part 2), we will construct $\tilde{\pi}$ for time slots $t > \tau$. Furthermore, starting from a preferred state at $t = \tau + 1$, we will show using forward induction that Relations (B-2) and (B-3) are satisfied for all time slot $t > \tau$.

Part 2: we assume that $\tilde{\pi}$ and $\tilde{\omega}$ are defined up to time $n - 1 \geq \tau$ and that relations (B-2) and (B-3) are satisfied at $t = n$, i.e., $\tilde{\mathbf{x}}(n) \preceq \mathbf{x}(n)$ and $\tilde{\mathbf{y}}(n) \preceq \mathbf{y}(n)$. We will show that at time slot n , $\tilde{\pi}$ can be constructed such that relations (B-2) and (B-3) are satisfied at $t = n + 1$.

There are three cases to be considered. These cases correspond to properties (S1), (S2) and (S3) of the vector $\mathbf{x}(n)$. For each one of these cases, we consider three sub-cases that correspond to properties (S1), (S2) and (S3) of the vector $\mathbf{y}(n)$. The construction of $\tilde{\omega}$ and $\tilde{\pi}$ for all these cases is the same as the construction carried out in Part 2 of the proof for Lemma 3 and it will not be repeated here. Same conclusions are valid here, i.e., relations (B-2) and (B-3) are satisfied at $t = n + 1$.

Part 1 and Part 2 above provide a complete description of the policy $\tilde{\pi}$ at $t = n$, for $n > \tau$. This policy resulted in queue length vectors $\mathbf{x}(n+1)$ and $\mathbf{y}(n+1)$ that satisfy Equations (B-2) and (B-3) respectively. Note that policy $\tilde{\pi} \in \Pi_\tau$ by construction in Part 1; its dominance over π follows from relation (21). \square

References

- [1] Y. Hong, W. Huang, F. Chiu, and C. C. J. Kuo, Cooperative communications in resource-constrained wireless networks, IEEE Signal Processing Magazine, vol. 24, pp. 4757, May

2007.

- [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*, 2004.
- [5] A. Sendonaris, E. Erkip, and B. Aazhang, Increasing uplink capacity via user cooperation diversity, in *Proc. of IEEE International Symposium on Information Theory*, (Cambridge, 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. 3749, Sep. 2006.
- [7] "MMR Harmonized Contribution on 802.16j (Mobile Multihop Relay) Usage Models," Document No. IEEE 802.16j-06/015, Sep. 2006.
- [8] J. P. Castro, *All IP in 3G CDMA Networks*. USA: John wiley & Sons Inc., 2004.
- [9] S. M. Ross, *Stochastic Processes*, 2nd ed. New York: Wiley, 1996.
- [10] Z. Roseburg, P. Varaiya, J. Walrand, "Optimal Control of Service in Tandem Queues," *IEEE Trans. Auto. Control*. AC27:600–610 (1982).
- [11] Y. Shi, W. Zhang, K. B. Latief, "Cooperative Multiplexing and Scheduling in Wireless Relay Networks," in *Proc. IEEE International Conference on Communications (ICC 08)*, 2008.
- [12] C.Y. Hong, A.C. Pang, "Link Scheduling with QoS Guarantee for Wireless Relay Networks," *IEEE Conference on Computer Communications (INFOCOM)*, Rio De Janeiro, Brazil, 2009.

- [13] A. Bletsas, H. Shin, and M. Z. Win, Cooperative communications with outage optimal opportunistic relaying, *IEEE Trans. on Wireless Commun.*, vol. 6, pp. 3450-3460, Sep. 2007.
- [14] T. C.-Y. Ng and W. Yu, Joint optimization of relay strategies and resource allocations in cooperative cellular networks, *IEEE Journal on Selected Areas in Commun.*, vol. 25, pp. 3283-39, Feb. 2007.
- [15] V. Sreng, H. Yanikomeroglu, and D. Falconer, Relay selection strategies in cellular networks with peer-to-peer relaying, in *IEEE Vehicular Technology Conf. (VTC03-Fall)*, Oct. 2003, pp. 1949-1953.
- [16] M. Yu and J. Li, Is amplify-and-forward practically better than decode-and-forward or vice versa? in *IEEE Inter. Conf. Acoustics, Speech, and Signal Processing, (ICASSP 05)*, vol. 3, Mar. 2005, pp. 3653-68.
- [17] A. Host-Madsen and J. Zhang, Capacity bounds and power allocation for wireless relay channels, *IEEE Trans. Information Theory*, vol. 51, no. 6, pp. 2020-2040, June 2005.
- [18] M. Chen, S. Serbetli, and A. Yener, Distributed power allocation for parallel relay networks, *IEEE Global Telecommunication Conf. (GLOBECOM 05)*, vol. 3, Nov. 2005, pp. 1177-1181.
- [19] D. Stoyan, *Comparison Methods for Queues and other Stochastic Models*, J. Wiley and Sons, Chichester, 1983.
- [20] T. Lindvall, *Lectures on the coupling method*, New York: Wiley(1992).
- [21] 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.
- [22] 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) pp. 3638-3643.

- [23] 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.
- [24] 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.
- [25] H. Al-Zubaidy, I. Lambadaris and I. Viniotis, “Optimal Resource Scheduling in Wireless Multi-service Systems With Random Channel Connectivity,” *IEEE Globecom '09*, Honolulu, HI, USA, Dec. 2009.
- [26] R. Lidl and G. Pilz, *Applied abstract algebra, 2nd edition*, Undergraduate Texts in Mathematics, Springer, (1998).
- [27] J. Walrand, “A note on optimal control of a queueing system with two heterogeneous servers,” *Systems and Control Letters*, 4: 131–134, 1984.
- [28] P. Nain, P. Tsoucas and J. Walrand, “Interchange arguments in stochastic scheduling”, *Journal of Applied Probability*, Vol 27: 815-826, 1989.