future internet

MDPI

Article

Scheduling for Multi-User Multi-Input Multi-Output Wireless Networks with Priorities and Deadlines

Li-on Raviv * and Amir Leshem *

Faculty of Engineering, Bar-Ilan University, Ramat Gan 5290002, Israel

* Correspondence: li.raviv@hotmail.com (L.o.R.); leshem.amir2@gmail.com (A.L.)

Received: 25 June 2019; Accepted: 26 July 2019; Published: 5 August2019

Abstract: The spectral efficiency of wireless networks can be significantly improved by exploiting spatial multiplexing techniques known as multi-user MIMO. These techniques enable the allocation of multiple users to the same time-frequency block, thus reducing the interference between users. There is ample evidence that user groupings can have a significant impact on the performance of spatial multiplexing. The situation is even more complex when the data packets have priority and deadlines for delivery. Hence, combining packet queue management and beamforming would considerably enhance the overall system performance. In this paper, we propose a combination of beamforming and scheduling to improve the overall performance of multi-user MIMO systems in realistic conditions where data packets have both priority and deadlines beyond which they become obsolete. This method dubbed Reward Per Second (RPS), combines advanced matrix factorization at the physical layer with recently-developed queue management techniques. We demonstrate the merits of the this technique compared to other state-of-the-art scheduling methods through simulations.

Keywords: scheduling; beamforming; WLAN; OFDM; resource allocation; EDF; priority; deadline; queuing; EDF; ZFBF

1. Introduction

In recent years, the deployment of media-rich applications for mobile devices has increased. These applications consume higher bandwidths from each mobile device. At the same time, the number of devices has grown rapidly and is expected to be on the order of tens of billions when including Internet-of-Things devices, vehicular networks, and personal communication [1,2]. This growth will lead to larger bandwidth requirements in the near future. According to Cisco's Networking Visual Index report [3], data traffic is expected to triple in volume by 2022. New critical applications such as autonomous cars demand very low latency for packet transmission. Enforcing low latency means that each packet has a deadline that needs to be met. This challenge can only be addressed if channel throughput is increased and a proper scheduling mechanism that manages both priorities and deadlines is introduced into the network. These challenges exist both in the Internet network cores and in cellular networks, whether these are C-RAN-based [4] or massive MIMO base-station-based networks [5]. Methods to perform effective network resource orchestration to minimize the bandwidth and network resource consumption were presented in [6]. However, to achieve the benefits of SDN orchestration, the base-stations or the C-RAN need to cross optimize the scheduling mechanism together with the physical layer beamforming [4,7–9] Moreover, dealing with both priority and deadline constraints requires a new approach to user scheduling and resource allocation, which ensures that the scheduler and the physical layer are harmonized to optimize performance. We can no longer expect that maximizing the spectral efficiency, using greedy allocation, will be optimal given the overall system constraints. To be optimal in the context of a multi-user MIMO system, we need a different approach to the physical layer beamforming and coding design.

The downlink of the MU-MIMO channel is known in the information theoretic context as the Broadcast Channel (BC). Several methods have been proposed to improve data rates over the broadcast channel. The Dirty Paper Coding (DPC) concept [10] for non-causally known multi-user interference at the transmitter was proven to eliminate the interference perfectly without power penalty.

The main drawback of this method is that finding the resulting optimal transmit co-variances has very high computational complexity. To reduce this computational complexity, several other beamforming techniques have been proposed. Zero-Forcing (ZF) beamforming was introduced as a simple alternative to DPC. In [11], the relationship between the linear ZF precoding design and generalized inverses in linear algebra was presented. Typically, the ZF method is sub-optimal. However, it has been shown to suffer a constant loss relative to capacity achieving approaches such as DPC in the case of large numbers of users and a high signal-to-noise ratio [12,13].

To overcome the shortcomings of ZF beamforming, where transmission to non-orthogonal users suffers significant power penalty, the proper scheduling and grouping of packets are required [13] to achieve optimal behavior. Indeed, the work in [7,13,14] proposed opportunistic or greedy grouping of users. This is indeed a special case of the general problem of joint scheduling and beamforming. This problem is known to be NP-hard [9]. Several approaches have been taken in order to propose scheduling with lower complexity beyond the greedy solutions above. Convex approximation algorithms were proposed in [9]. In the case of large numbers of MSs, opportunistic policies achieve near DPC performance in terms of the average delay [7]. Different policies handle the power constraint issue [15,16], trying to maximize the overall throughput. Other approaches to maximize the throughput used an arbitrary set of orthogonal beamforming in the case of low-rate beamforming feedback [17]. In [5], partitioning the users into groups with the approximately similar channel co-variance eigenvectors method was presented. This method enables the "massive MIMO" gains. In [8], a unified urgent weight was proposed by taking into account QoS requirements, such as delay deadline, minimum data rate, Queue State Information QSI, and the user fairness requirement. Various methodologies have been used to approach the aforementioned joint optimization task in the downlink of MU-MIMO communication systems [18]. In [17], Per User Unitary and Rate Control (PU2RC) was analyzed, and in [19], a generic channel covariance-based beamforming scheme was presented. In [4], the authors described a method to maximize the system utility, subject to the diverse QoS requirements of users and the power constraints. In [20], Joint Opportunistic Scheduling and Receiver Design (JOSRD) was analyzed for MU-MIMO. All these studies, considered scheduling policies to improve the resource block allocation while maximizing the SINR or the achievable overall system throughput. However, improving capacity is not enough to support new applications. These techniques cannot be applied when both deadlines and constraints are required. While other network management techniques exist for other network models such as device-to-device interference management [21] or ad-hoc networks [22], these techniques are inapplicable in the context of base-station scheduling.

Many IoT systems are assumed to be hard real-time systems. In hard real-time systems, if a packet fails to be delivered before its deadline expires, it is considered to be lost. The hard real-time systems problem has been widely discussed in queuing theory [23–26]. The Earliest Deadline First (EDF) scheduling policy is one of the most common methods to schedule packets in a hard real-time environment [25]. The EDF is optimal in many queuing models [25,27–31]; however, in the presence of prioritized packets, it can be sub-optimal [32]. At the same time, applications differ in terms of their importance. Application priority normally reflects their importance. The priority is attached to the application's packet. Priority becomes a reward upon successful delivery of a packet. The reward is considered to benefit the network if the packet is delivered on time [33]. Scheduling mechanisms that consider both rewards and deadlines were presented in [34]. Another scheduling policy is based on the $c\mu_k/\theta_k$-rule presented in [35–37]. The $c\mu_k/\theta_k$-rule is based on having a finite number of queues, each queue presenting a possible packet's priority. The policy selects the queue to be served. The decision aims to reduce the summation of $c\mu_k/\theta_k$ over the queues. In $c\mu_k/\theta_k$, c presents a cost function of the queue, which is similar to a reward summation, μ_k presents the k^{th} queue packets

arrival rate, and θ_k presents the abandon rate from the queue. The policy selection process takes into consideration the queue rewards, the arrival rate expectation, the abandon rate expectation, and the queue length.

A similar approach to prioritization can be found in WLAN standard 802.11e [38], which implements a queue per priority and assigns different idle time bounds according to the queue priority. Recently, the authors developed a significantly improved solution for managing queues with both priorities and deadlines [32].

The above queuing theoretic papers discuss queue management policies independent of the underlying physical layer, assuming a single server with a deterministic or random service rate. In contrast, managing queues of packets aimed at independent users with different communication channels requires a cross-layer approach, as priorities and deadlines need to be measured against the effect of each user on other users grouped together. A simplified approach was presented in [39] where packets with priorities and deadlines were scheduled to static beams. It is the goal of this paper to propose a cross-layer approach for scheduling and beamforming. A typical application of the proposed approach would be for WLAN networks; for example, WLAN equipment supporting the IEEE 802.11ac standard planned for a maximum throughput of at least 500 Mb/s for a single user and at least 1 Gb/s for multiple users [40,41]. The IEEE 802.11ac Multiple Access Control (MAC) layer extended the IEEE 802.11n standard to accommodate an MU-MIMO [42]. In order to increase the capacity, new bands were allocated, such that the IEEE 802.11ac standard aims for a WLAN working at multi-user transmission at 60 GHz. This standard was defined to increase the throughput of next-generation WLANs via both analog and digital beamforming [43–45]. The 802.11 standard indeed contains a priority mechanism at the medium control level; however, these are insufficient to support delay-critical applications. In this paper, we use a scheduling policy combined with the ZF beamforming technique to improve systems with traffic under a hard real-time environment, as well as priorities.

The remainder of this paper is organized as follows. Section 2 introduces the system model. Section 3 describes the RPS algorithm. Section 4 presents simulation results, and conclusions are drawn in Section 5.

2. System Model

Consider a system composed of a WLAN Access Point (AP) and a finite number of Mobile Stations (MSs), as depicted in Figure 1. Data packets arrive at the AP and need to be transmitted to their designated MS. The AP has multiple antennas, whereas each MS has a single antenna (multi-user MIMO). This problem is a variant of the well-known broadcast channel problem. The AP needs to transmit the packets to the MS. The arriving packets have both priority and deadlines. The system is defined to be a hard real-time system; i.e., packets that miss their deadline do not earn their priority as rewards. The goal is to maximize the sum of the rewards over the complete system.

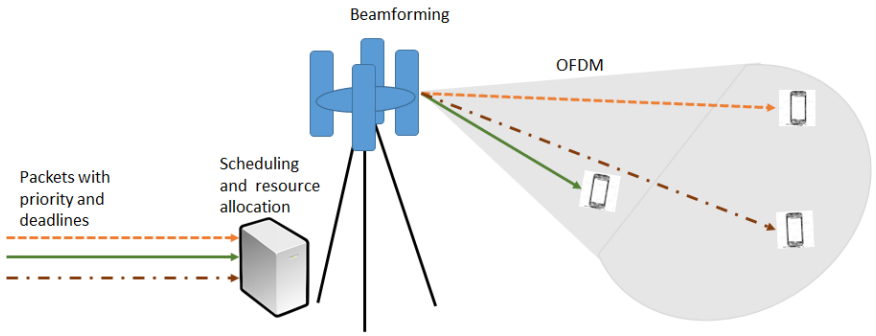

Figure 1. System.

2.1. Queuing Model

This section summarizes the standard queuing model. The complete set of assumptions for the queuing model can be found in [32].

A queuing model is composed of jobs, scheduling policy queues, and servers. The jobs entering the system are described by a renewal process with different attributes. The attributes may be assigned upon arrival or later on. The scheduling policy is responsible for allocating the servers to the jobs and choosing the job to be processed out of the queue when the server is idle. The queuing model of the data packets assumes a renewal reward process [33]. We use the extended Kendall [32,46] notation $A/B/C-D/P$ to characterize the input random process and the service mechanism.

As defined below:

- A refers to the inter-arrival time of the renewal process.
- B refers to the random process of service time required by the packets.
- C refers to the number of servers.
- D refers to the deadline random process.
- P refers to the priority or reward random process.

Each packet arriving at the AP has its destination MS, payload, priority, and deadline. Let J_i ($i \in \mathbb{N}$) be the i^{th} packet that arrives to the system. Let A_i, B_i, C_i, D_i, and P_i be random variables defining the job, J_i. The packet arrival time, denoted t_i, is defined by:

$$t_i = t_{i-1} + A_i = \sum_{j=1}^{i} A_j. \tag{1}$$

Let the tuple $J_i =< a_i, b_i, e_i, p_i >$ represent a packet i with its random parameters where a_i, b_i, e_i, and p_i are realizations of A_i, B_i, E_i, and P_i.

Following are the queuing model assumptions:

A1 The pair (A_i, P_i) is a renewal reward process.
A2 Packets' deadlines are measured with respect to the end of transmission.
A3 The reward is obtained only if packets arrived on time (hard real-time system requirements).
A4 b_i, e_i, and p_i are known upon arrival of J_i to the queue.
A5 The scheduling policy is non-preemptive, and forcing idle time is not allowed.

Let S_t^π be the set of packets that were successfully delivered to their destinations by policy π up to time t. By definition, the renewal reward process provides a mechanism to analyze the performance of the system. The cumulative rewards' function is a simple way to compare the performance of different algorithms.

Definition 1. *The cumulative reward function for time t and policy π is:*

$$U_t^\pi = \sum_{J_i \in S_t^\pi} p_i. \tag{2}$$

The objective is to find a policy π that maximizes the cumulative reward function.

2.2. ZF Beamforming for MU-MIMO Wireless Networks

Consider the downlink of an MU-MIMO system with a single Access Point (AP) operating over a single band using N transmit antennas and a single antenna receiver for each of the M Mobile Stations (MSs). The combined vector channel can be described as:

$$\mathbf{y} = \mathbf{HWx} + \mathbf{z}. \tag{3}$$

where $\mathbf{y}, \mathbf{x}, \mathbf{z} \in \mathbb{C}^{M\times 1}$, a channel gain matrix $\mathbf{H} \in \mathbb{C}^{M\times N}$, and a beamforming matrix $\mathbf{W} \in \mathbb{C}^{N\times M}$.

Let:

$$\mathbf{H}^* = \mathbf{QR} \tag{4}$$

be a **QR** decomposition of $\mathbf{H}^*$ [47]. Let $\mathbf{L} = \mathbf{R}^*$, and let $\mathbf{H} = \mathbf{LQ}^*$. Assume that $\mathbf{W} = \mathbf{Q}$. The equivalent channel is given by:

$$\mathbf{y} = \mathbf{Lx}. \tag{5}$$

Note that at this stage, each user is not subject to interference by users with a higher index [48]. To complete the ZF beamforming, we need to invert $\mathbf{L}$ and normalize the total transmit power. Practically, since $\mathbf{L}$ is a lower triangular matrix, we can use forward substitution to compute the transmitted vector. Furthermore, we can normalize the total power of the transmitted vectors prior to beamforming with $\mathbf{Q}$, since $\mathbf{Q}$ is unitary. Let $\mathbf{s} \in \mathbb{C}^{M\times 1}$ be the vector of symbols that are required to be transmitted to the different mobile stations.

We also use the following assumptions:

A6 The AP has perfect Channel State Information (CSI).
A7 Transmitted data symbols (s_k) are uncorrelated
A8 The power allocated to each user is constant. To simplify notation, we assume that the same power is allocated to all users, i.e., $E\{s_k^2\} = 1$ and $E\{x_k^2\} = \mathcal{P}$.

Let $\mathbf{x} = \mathbf{L}^{-1}\mathbf{G}\,\mathbf{s}$ where $\mathbf{G}$ is a diagonal $M \times M$ real gain matrix.
Since $L\mathbf{x} = \mathbf{Gs}$, then $\mathbf{x}$ and $\mathbf{G}$ can be calculated using forward substitution as described below:

$$\sum_{j=1}^{k} l_{kj}x_j = g_{kk}s_k \implies x_k = \frac{1}{l_{kk}}\left(g_{kk}s_k - \sum_{j=1}^{k-1} l_{kj}x_j\right). \tag{6}$$

Assuming A8, then:

$$E\{x_k^2\} = \frac{1}{l_{kk}^2}E\left\{(g_{kk}s_k - \sum_{j=1}^{k-1} l_{kj}x_j)^2\right\} = \frac{1}{l_{kk}^2}\left(g_{kk}^2 + \mathcal{P}\sum_{j=1}^{k-1} l_{kj}^2\right) = \mathcal{P}. \tag{7}$$

From Equation (7), it can be derived that:

$$g_{kk} = \sqrt{\mathcal{P}\left((l_{kk}^2 - \beta_k \sum_{j=1}^{k-1} l_{kj}^2\right)}, \tag{8}$$

where $\beta_k \in [0, 1]$ and $g_{k,k}$ is always real.

Define $\mathbf{x}$ by:

$$x_k = \frac{1}{l_{kk}}\left(\sqrt{\mathcal{P}(l_{kk}^2 - \beta_k \sum_{j=1}^{k-1} l_{kj}^2)}s_k - \sum_{j=1}^{k-1} l_{kj}x_j\right). \tag{9}$$

The energy that is allocated to transmit s_k is:

$$\mathcal{P}\left(1 - \frac{\beta_k}{l_{kk}^2}\sum_{j=1}^{k-1} l_{kj}^2\right). \tag{10}$$

Therefore, the SINR as a function of β_k is given by:

$$SINR(\beta_k) = \frac{\mathcal{P}\left(1 - \frac{\beta_k}{l_{kk}^2}\sum_{j=1}^{k-1} l_{kj}^2\right)}{l_{kk}^2(1-\beta_k)\sum_{j=1}^{k-1} l_{kj}^2\mathcal{P} + \sigma_k^2}. \tag{11}$$

The optimal β_k solves the following problem:

$$\beta_k = arg\max_{\beta_k \in [0,1]} (SINR(\beta_k)). \tag{12}$$

The achievable bit rate for MS k is given by:

$$r_k = \log_2(1 + SINR(\beta_k)). \tag{13}$$

2.3. Combining Scheduling and Beamforming

We now describe the overall system model. In the queuing model presented above, a known service time is assumed. In the combined model, service time or the packet transmission time should be derived from the packet length and the channel throughput at the time of the transmission. In contrast to the standard queuing model, this implies that the transmission time is affected by the packets already scheduled for transmission and calls for a cross-layer approach to scheduling and beamforming.

In order to keep the model simple, the following changes are made. In the combined model, the reference to an MS is omitted; instead, the channel gain vector is added to the packet ($J_i = < a_i, b_i, e_i, p_i, d_i, \mathbf{h}_i >$); in other words $\mathbf{w}_k$, $\mathbf{x}_k$, and $\mathbf{s}_k$ are referred to accordingly. This approach is similar to the original model with respect to the system objectives. The original service time of a packet b_i refers now to the packet length. The actual service is defined by the packet length and the available data rate for a specific subscriber at a specific time.

The additional assumptions are:

A9 Each packet is addressed to a specific MS (multicasting will be considered in an extension of this work).
A10 $\mathbf{h}_i$ is known upon arrival of J_i to the queue.
A11 Channel coherence time is longer than the packet service time.
A12 The queuing model is $G/D/n - /G/B$ where $n = N$.

As in standard queuing models, this model also has three possible events:

E1 The packet arrives at the system.
E2 The beginning of packet transmission.
E3 The end of packet transmission.

The system state machine is depicted in Figure 2 . Packets that arrive at the system (Event E1) need to either wait in the queue or begin transmission (Event E2). At this stage, the scheduler can drop expired packets, select a queue for the packet (for a multiple queue model), and implement a queue insertion policy. The beginning of packet transmission (Event E2) is a result of both the existence of one or more eligible packets (either in the queue or arriving) and available resources at the transmitter. Based on resource availability, the scheduler selects the packet to be transmitted from the queue. The packet is dropped if its deadline has expired, and a new packet is selected. Then, the scheduler calculates the packet beamformer and transmits the new packets with the existing packets. The end of packet transmission (Event E3) frees resources. The transmitter keeps transmitting packets for which their transmission did not end and adds new packets for transmission (Event E3). The first case may require a precoder recalculation to achieve better performance after the resources are released.

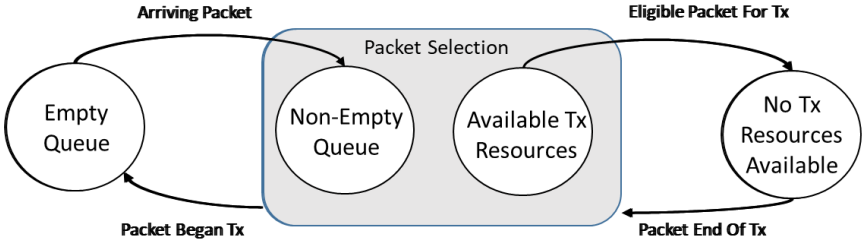

Figure 2. System state machine.

Communication systems always use a discrete set of modulation and coding schemes. Therefore, there is a minimal data rate that the system can transmit. For such a data rate, there is a minimal SINR requirement for proper decoding.

In order to support variable matrices and vectors over time, we introduce the notation of $\mathbf{H}_t, \mathbf{Q}_t, \mathbf{L}_t, \mathbf{s}_t$, and $\mathbf{x}_t$.

As is common in high spectral efficiency coherent transmission, we assume that the channel is under-spread [49], i.e., fixed along the transmission of each packet.

3. RPS Scheduling Policy

User selection is crucial for ZFBF in particular and linear ZF in general. Hence, the interaction between the scheduler and the physical layer is a crucial element in any scheduler for MU-MIMO systems. Moreover, the addition of new users can reduce the rate of already allocated users. Full combinatorial search for optimal user selection even without deadlines and priorities is prohibitively complex. Under deadline constraints, adding users to a set of transmitting users can result in missing the deadlines of these users by reducing their rates. Therefore, a scheduling policy that decouples this dependence should significantly reduce the complexity of user selection, while allowing the scheduler to allocate the users independently of previously-allocated users. The proposed RPS policy achieves this through a linear algebraic decomposition of the channel matrix in a way that allows for a sequential addition of users without harming already transmitting users.

In this section, we present a new scheduling policy called RPS that combines ZFBF and reward per transmission time ordering. The RPS algorithm gives priority to packets with a higher reward per second. The arrival queue is ordered according to EDF, and the search for the packet with the highest RPS is limited to the first N packets. The RPS scheduling policy uses ZFBF as the precoding mechanism. The ZFBF by nature provides resource allocation in decreasing order from the first row of the channel matrix to the last row. The RPS takes advantage of this characteristic to provide better bandwidth to selected packets. Packets that cannot achieve the required bandwidth to meet their deadlines are not selected for transmission. Consequently, several packets with the same destination or packets that should use the same steering vector are not transmitted together, and only one packet is transmitted at a time. In any case, such packets are not transmitted until either they get the required bandwidth or they miss their deadline and are dropped from the queue. Figure 3 depicts the new scheduling policy.

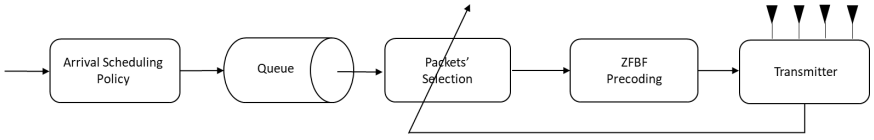

Figure 3. RPS policy packet flow.

The scheduling policy is composed of four algorithms:

M1 New packet insertion into the arrival queue, Algorithm 1.

M2 Selection of the packet for transmission in case there is an eligible packet and there are available transmission resources, Algorithm 2.
M3 Precoder update for additional new packet transmission, Algorithm 3.
M4 Precoder downdate for a packet that ended its transmission, Algorithm 4.

Algorithm 1 responds to a packet arrival event (Event E1). If there are available resources for transmission and the queue is not empty, the selection algorithm is activated (Algorithm 2). If the selection process ends successfully, Algorithm 3 runs to prepare the new precoding. At this point, the packets that are eligible for transmission and were precoded are transmitted. For packets that have just begun their transmission, this event is Event E2. At the end of transmission, i.e., Event E3, Algorithm 4 is activated in order to downdate the precoding matrices. Figure 2 depicts the state machine that activates the algorithms.

Algorithm 1 New packet queue insertion.

Let J_i be the packet that arrives at the system at time t_i.

1: Drop J_i if its deadline has expired
2: Add J_i to the queue according to the EDF arrival policy.

The following algorithm selects a packet from the arrival queue to be transmitted.

Algorithm 2 Select new packet for transmission.

Let H_{t-1} be the channel matrix in the previous stage, and let Q_t be the arrival queue at the current time. Let $Q_t(i).h$, $Q_t(i).length$ and $Q_t(i).deadline$ be the channel vector, the length, and the deadline of the packet, which is in the i^{th} position in queue Q_t. Let $currentTime$ be the actual time as defined by the index time t. Let K be the packet selection window size.

1: $windowSize = min(length(arrivalQin), K)$;
2: i=1;
3: **while** $i < windowSize$ **do**
4: $Htemp = [H_{t-1}, Q(i).h]$
5: Perform L and Q updating
6: Calculate the expected transmission $rate(i)$ for packet $Q_t(i)$ using Equations (7), (11), and (13)
7: **if** $\frac{Q_t(i).length}{rate(i)} + currentTime < Q_t(i).deadline$ **then**
8: $RewardPerUnitTime(i) = \frac{Q_t(i).priority * rate(i)}{Q_t(i).length}$
9: **else**
10: $RewardPerUnitTime(i) = 0$
11: $j = \underset{i}{argmax} RewardPerUnitTime(i)$
12: **if** $RewardPerUnitTime(j) == 0$ **then**
13: return -1
14: **else**
15: return j

Algorithm 3 Precoder update.

1: $H_t = [H_{t-1}, Q(i).h]$
2: Perform L and Q updating
3: Calculate **x** using Equations (7), (11), (9), and (13)

For additional details about QRupdate, please refer to [50].

Algorithm 4 Precoder downdate.

1: Update H_t by removing the column of the packet that ended its transmission from H_{t-1}
2: Perform L and Q downdating
3: Calculate $\mathbf{x}$ using Equations (7), (11), (9), and (13)

In Algorithm 4, $\mathbf{L}$ and $\mathbf{Q}$ are downdated; alternatively, with additional computational effort, it is possible to calculate L and Q from the beginning. In this case, packets that were added to the transmission after the packet that ended may improve their bit rate. Algorithms 3 and 4 perform QR factorization updating and downdating. These operations can be carried out more efficiently as presented in [51,52] and [50]. In the case of massive MIMO, the probability of having almost orthogonal channels is very high.

Computational Complexity of RPS

The computational complexity of RPS is composed of three components:

- Complexity of processing a new packet.
- Selection of a transmitted packet.
- Transmission complexity.

Upon arrival, the RPS adds the packet to the queue according to the earliest deadline first order with a complexity of $O(log|Q|)$ where $|Q|$ is the size of the arrival queue; when the buffer size is limited, this is always less than $log(buffersize)$. By our assumptions, the size of the arrival queue is bounded by:

$$Q_{max} = \frac{\text{maximal deadline} \times \text{minimal packet length}}{\text{maximal bandwidth} \times \text{maximal number of beams}}. \tag{14}$$

This complexity is similar to other sorted policies like EDF and priority greedy. In contrast, LIFO and FIFO have a $O(1)$complexity of packet insertion. However, even the logarithmic complexity is very moderate.

The computational complexity of adding a new packet to a ZF beamformer (i.e., QR update) using the Gram–Schmidt process or Householder transformation is $2N^2$ [47]. This computation is required regardless of the scheduling policy. After this step, we keep L and Q to avoid new computation. Selecting a new packet for transmission in RPS requires $K-1$ partial QR updates per packet on average. Each partial QR update has a computational complexity of $2N$, i.e., if the packet is not selected and a different packet is selected, the L and Q matrices are updated accordingly. When a packet ends, we perform a simple QR downdate by removing the relevant column and row, which has complexity $2N$. Therefore, the total average complexity for processing a packet is bounded by:

$$log(Q_{max}) + 2N^2 + 2N(K-1). \tag{15}$$

Note that a packet transmission complexity for any scheduling policy with a sorted queue is $log(Q_{max})$. Therefore, the updating of a ZF beamformer is $2N^2 + 2N(\hat{K}-1)$ where $\hat{K}$ is the average number of packets processed until an eligible packet is found. Therefore, the total average complexity of the beamformer processing per packet is bounded by:

$$log(Q_{max}) + 2N^2 + 2N(\hat{K}-1). \tag{16}$$

Optimal resource allocation even without beamforming is an NP hard problem [6] and [32]. This implies that our policy is likely to be sub-optimal. However, with a moderate polynomial complexity, we outperform all prior scheduling algorithms that are not designed jointly with the beamformer.

4. Simulations

In this section, we present numerical simulations comparing the performance of FIFO, EDF, priority greedy (Greedy), and the cross-layer RPS scheduling policies.

The simulation software was written using MATLAB R2019a. The software was built of two modules. The first module is called the Packet Generator (PG) and is responsible for packet generation. The second module is called Policy Runner (PR) and is used to simulate the scheduling policies behavior and measure their performance. At the beginning, the PG generates a configurable number of tests. Each test randomly picks a configurable number of MS locations. Then, the PG generates a configurable number of packets, which simulates the arrival process. After the PG generates the tests and their packets, the PR processes them according to the different policies' schemes. All policies receive as an input the same stream of packets with the same MSs location. The PR measures the cumulative rewards, the number of packets that were delivered on time, the number of packets that cannot be delivered because of the short period between the arrival and the deadline, packets that expired while waiting in the queue and packets that were delivered after the deadline expired. The calculation of the available bandwidth for transmission was based on (13). We ran the simulation using a personal computer equipped with an Intel(R) Core(TM) i5-4670S 3.1-GHz CPU and 16 GB RAM memory. The average CPU utilization was 27%, and the average memory utilization was 1.34 GB. The processing environment impacted mainly the simulation, which was used here in order to quantify the computational complexity and not for the actual processing time of the beamformer.

The data for the channel simulation were generated as follows: An AP was located at the origin with and eight-element uniform linear antenna array operating at a frequency of 2.4 GHz ($\lambda = 12.5$ cm). We generated 200 sets of locations of $N = 32$ MSs and channels for each of these. The multipath effect was simulated by a superposition of a line of site channel and seven i.i.d. Rayleigh fading taps, with excess delay and relative power according to Extended Pedestrian A. The mobile stations were located randomly 2–20 m away from the AP. Figure 4 depicts one of the MS's location realization of one test, while Figure 5 depicts the realization of 200 tests.

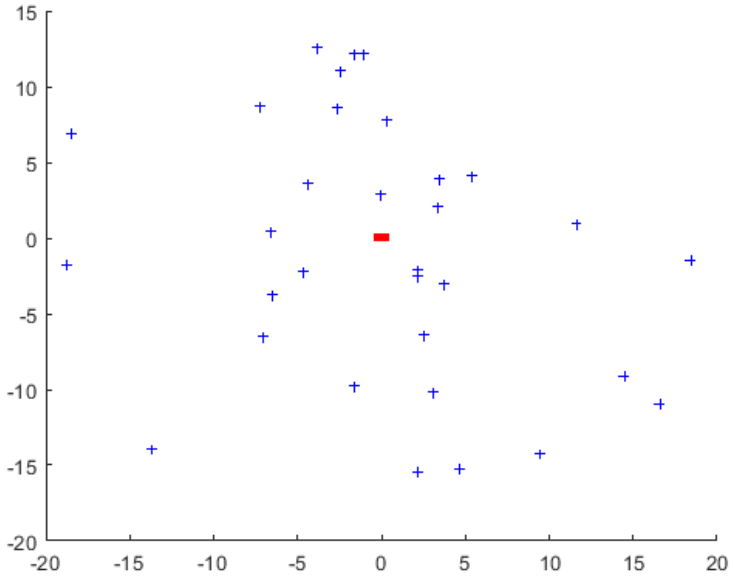

Figure 4. Access point (red) and mobile station (blue) locations for a single test.

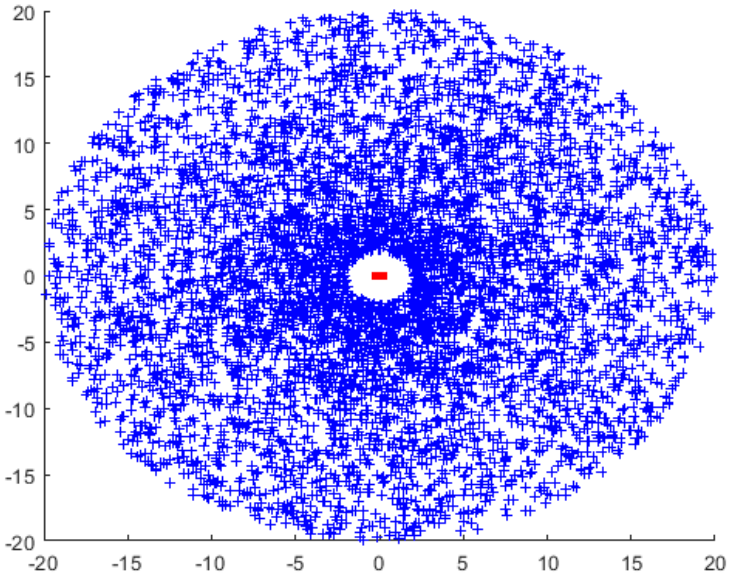

Figure 5. Access point (red) and mobile station (blue) locations for 200 tests.

The transmission power was set to 20 dBm. The total power of the AWGN was set to -101 dBm. We modeled the large-scale fading using a path loss of $(\frac{4\pi d}{\lambda})^{\alpha}$ where $\alpha = -3.5$ and d is the distance between the antenna to the MS.

The arrival process contained 2000 packets. The different random processes were set as follows:

1. The arrival-time process was exponentially distributed with $\lambda_a = 42{,}000$, 46,000, 50,200, 54,400, 58,000 and $62,000$.
2. The packets size was set to be similar to the Internet packet size distribution [53]. Forty percent were short packets with a length of 64 bytes; 40% were long packets with a length of 1500 bytes. The length of the rest of the packets (medium packets) was uniformly distributed between the short and the long packet lengths$\sim U[64, 1500]$.
3. The deadline was exponentially distributed with $\lambda_d = 15$ from the arrival to the end of service.
4. The reward was an integer that was distributed according to packet length. Short packets' reward was uniformly distributed $\sim U[5, 9]$. Long packets' reward was uniformly distributed $\sim U[1, 6]$, and medium packets' reward was uniformly distributed $\sim U[1, 10]$.
5. The destination of the packet was distributed uniformly between the MSs $\sim U[1, 32]$.

We compared four scheduling policies: FIFO, priority greedy, EDF, and RPS. In order to generate a fair environment, the FIFO, priority greedy, EDF, and RPS scheduling policies were implemented similarly in the following manner:

1. Packets whose deadline expired were removed from the arrival queue.
2. Packets with less than a 1-Kpbs rate did not begin transmission.
3. Packets were selected only from a window of K packets at the prefix of the queue.
4. All scheduling policies used ZFBF with a sequential insertion approach.
5. Matrices **L** and **Q** were downdated after packets ended their transmission.

The naive implementation of FIFO, EDF, and priority greedy selects a packet from the head of the queue and waits until this packet has enough bandwidth to be transmitted. We first explore the possibility to allow these policies to look for an eligible packet out of the first K packets in the queue. We ran simulations with different window sizes $K = \{1, 2, 4, 6, 8, 10, 12\}$ in order to set the appropriate window size. Figures 6–8 present the results of the number of delivered packets, the cumulative rewards, and the processing time using different values of K, respectively. The results showed that

increasing the window size improved the performance of the all policies while consuming more processing time. The RPS achieved an optimal performance already when $K = 2$. In other policies, the larger the window, the better the performance achieved, as well as the higher the CPU run time. In the subsequent simulations, we used a $K = 8$ in which FIFO, priority greedy, and RPS had a similar CPU consumption per packet.

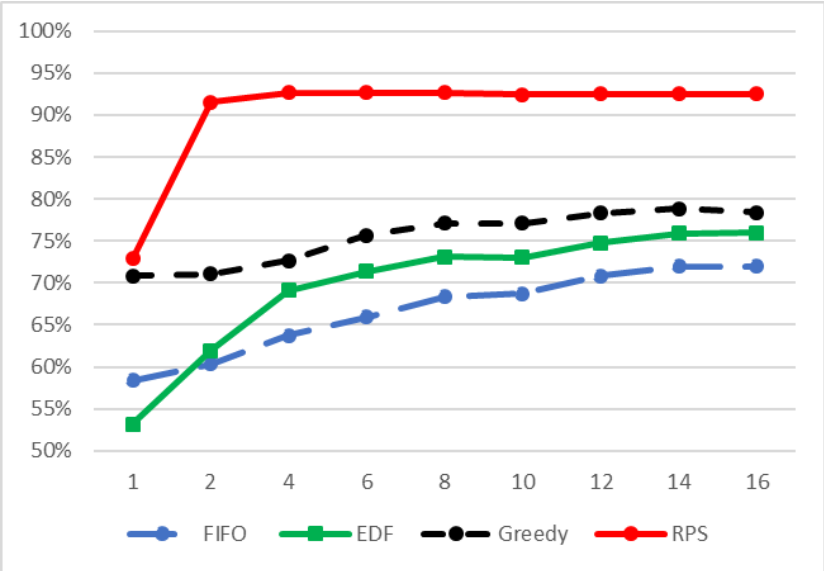

Figure 6. Percentage of delivered packets with different window sizes ($\lambda_a = 56{,}000$).

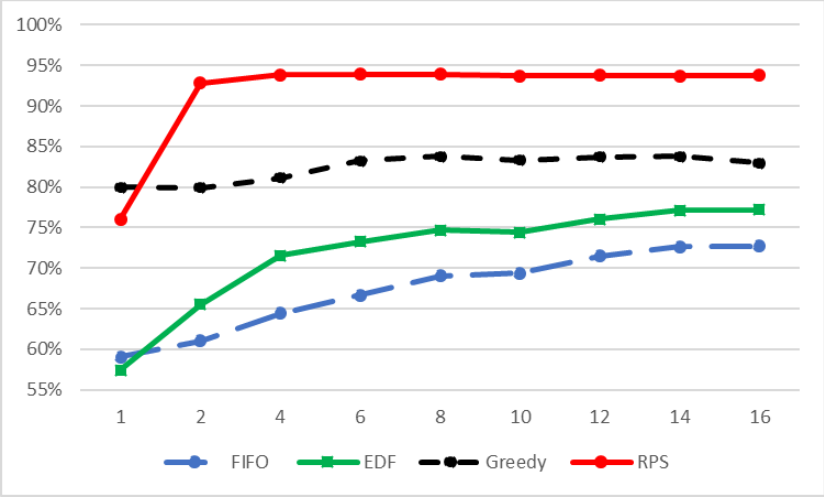

Figure 7. Cumulative rewards' percentage with different window sizes ($\lambda_a = 56{,}000$).

Figures 9 presents the cumulative reward of the different policies at different arrival rates. Figure 10 presents the number of packets that were delivered on time at different arrival rates. The general trend was as the traffic load became higher, less packets were delivered on time; however, the RPS packet loss was significantly lower than the other policies. In all measurements, the RPS policy outperformed significantly the other scheduling policies.

Figures 11 and 12 present the overall simulation processing time and the average processing time per a delivered packet at different arrival rates. As expected, increasing the traffic load increased the processing time linearly. The processing time of FIFO, priority greedy, and RPS were similar. EDF showed worse performance. In Section 3, we present computational analysis of a single packet's handling. The results here emphasize that joint scheduling kept the overall performance low with less redundant processing time of packets that were not delivered on time.

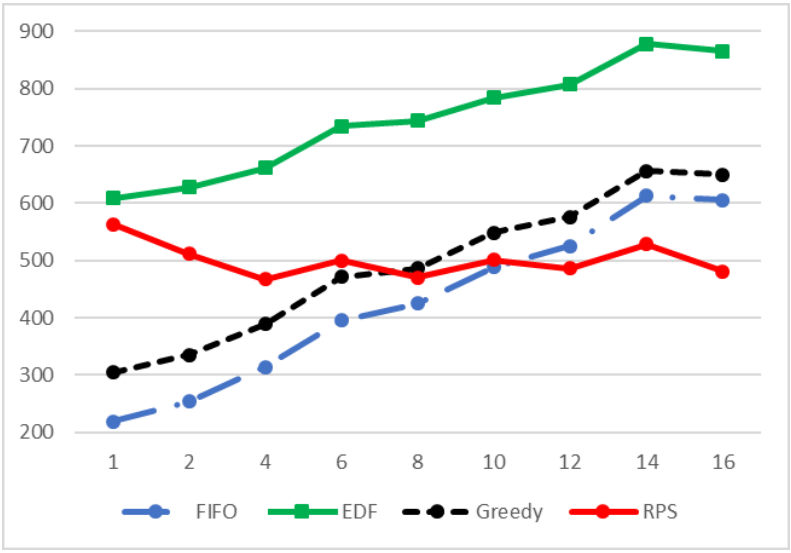

Figure 8. Processing time in seconds with different window sizes ($\lambda_a = 56{,}000$).

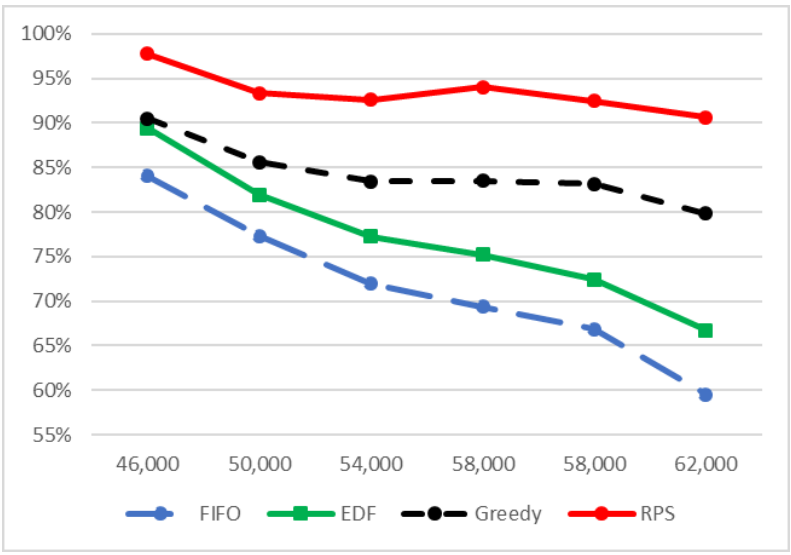

Figure 9. Percentage of cumulative reward with different arrival rates.

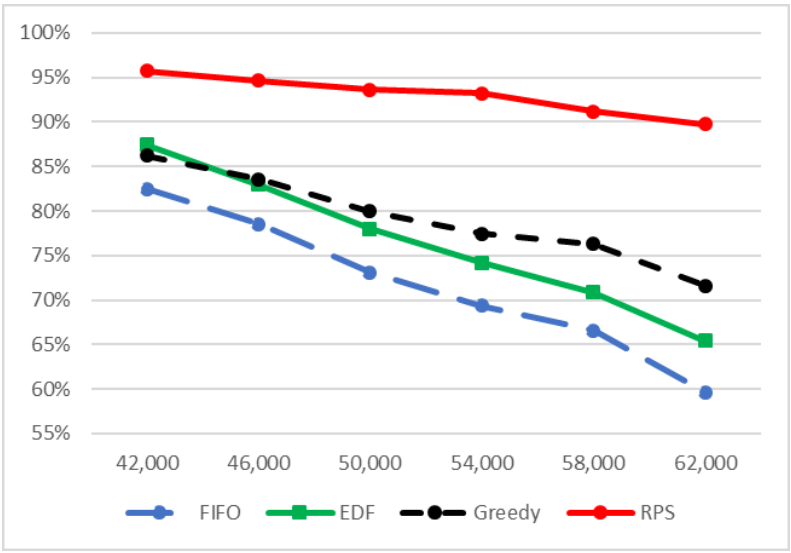

Figure 10. Percentage of delivered packets with different arrival rates.

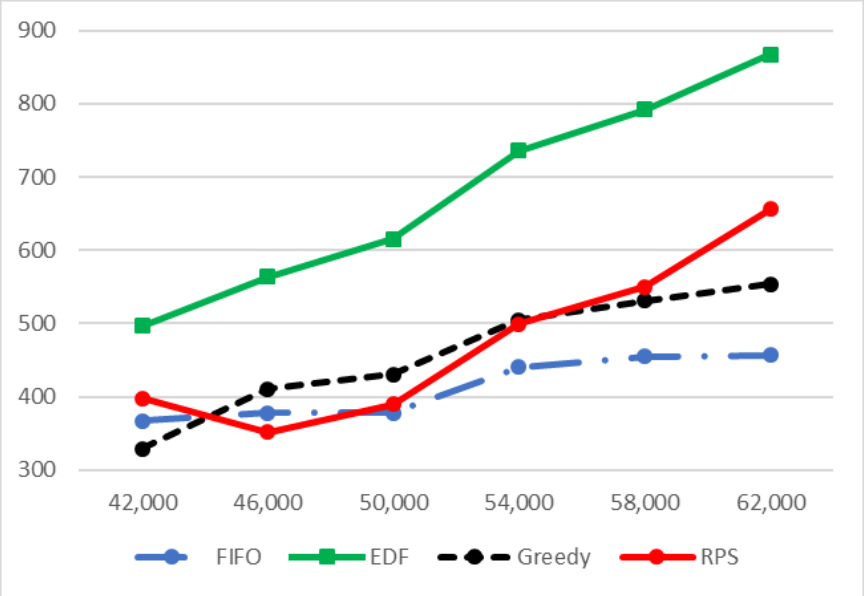

Figure 11. Processing time in seconds with different arrival rates.

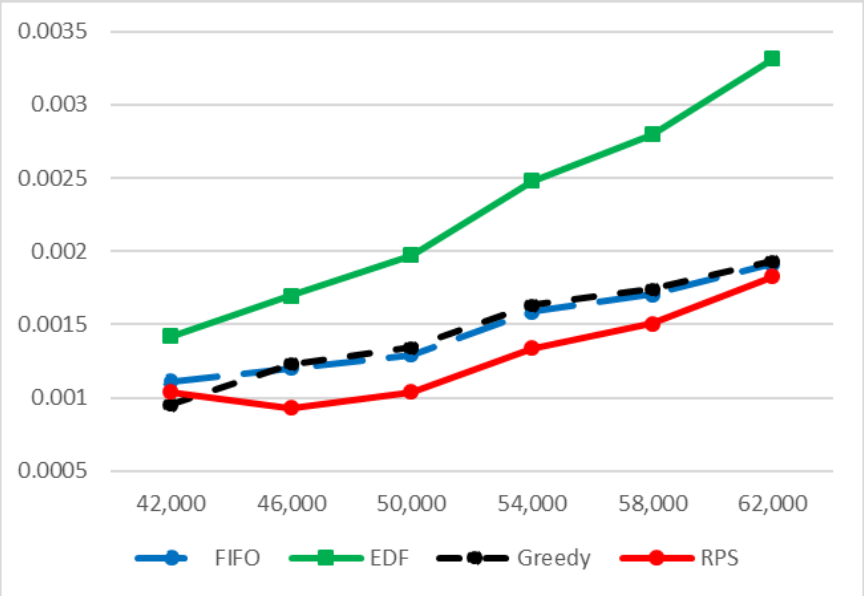

Figure 12. Delivered packet average processing time in seconds with different arrival rates.

Figures 13 and 14 present the CDF of the packets that were delivered on time. Figures 15 and 16 present the CDF of the cumulative reward. The 10% outage point at $\lambda_a = 56{,}000$ for RPS was 30% better than all other scheduling policies, both in the cumulative reward and in the number of delivered packets. As the arrival rate was higher, all policies collected less rewards and delivered less packets on time. The degradation of the service was not similar for all policies; the FIFO policy suffered the most; the EDF policy and the priority greedy policies suffered less; while the RPS policy had a minor degradation of the performance. The RPS policy cumulative reward was 35% better at a 10% outage than the priority greedy policy and was 40% better than the priority greedy policy in the expected total number of delivered packets.

One should note that the RPS policy also provided a significantly more robust packet delivery, since the expected reward CDF was very concentrated compared to the other scheduling policies.

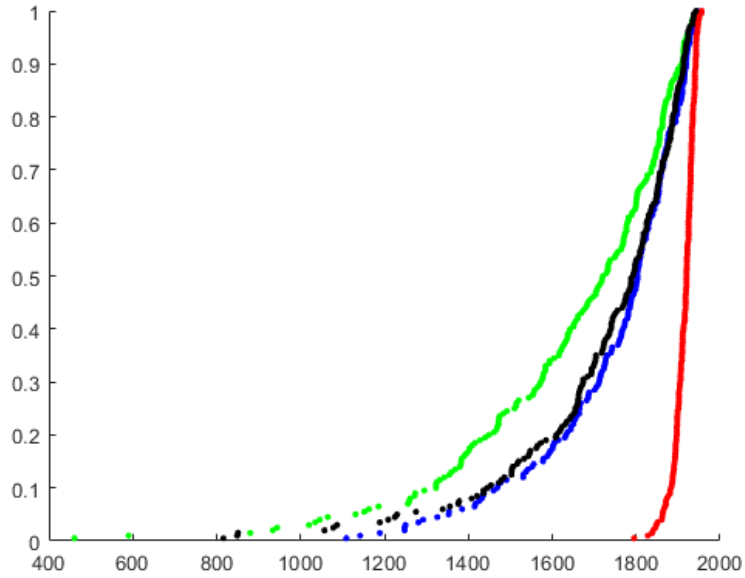

Figure 13. CDF of number of delivered packets ($\lambda_a = 42{,}000$). FIFO; , EDF; , Greedy; , RPS.

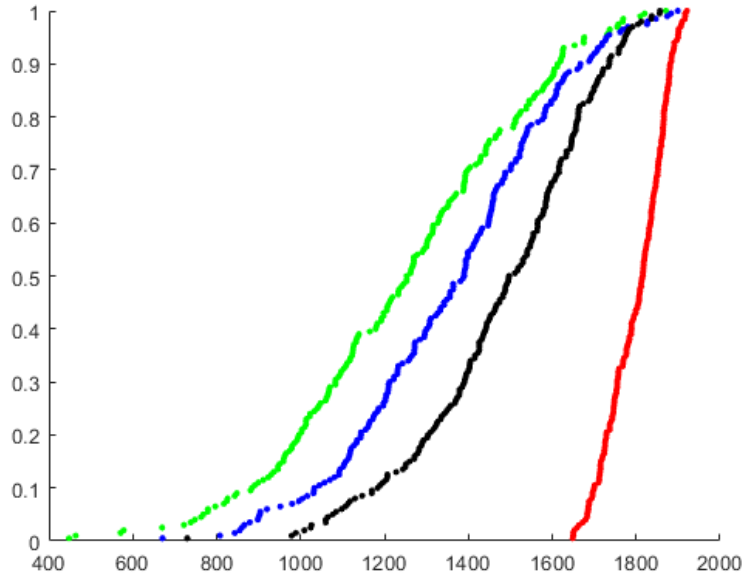

Figure 14. CDF of number of delivered packets ($\lambda_a = 66{,}000$). FIFO; , EDF; , Greedy; , RPS.

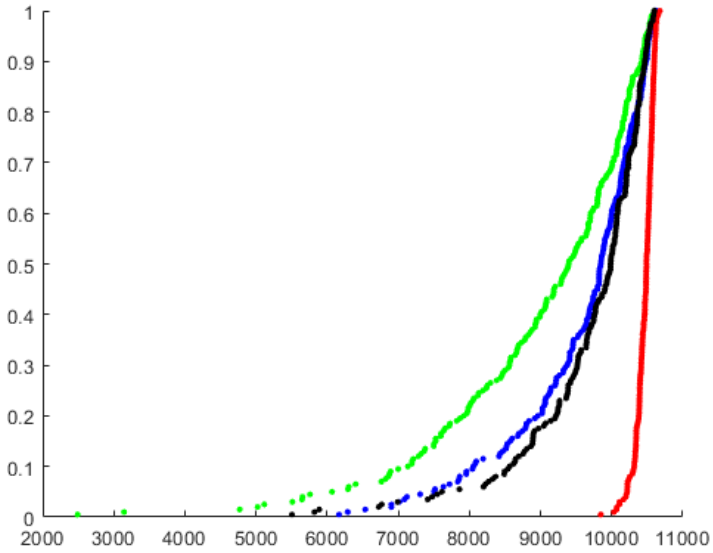

Figure 15. CDF of cumulative rewards ($\lambda_a = 42{,}000$). FIFO; , EDF; , Greedy; , RPS.

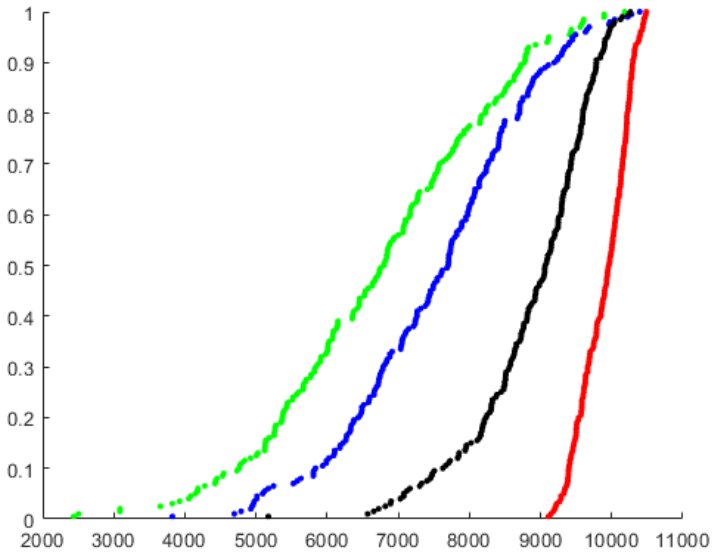

Figure 16. CDF of cumulative rewards ($\lambda_a = 66{,}000$). FIFO; , EDF; , Greedy; , RPS.

5. Discussion and Conclusions

We presented a cross-layer queuing model for packets with priority and deadlines that is applicable in an MU-MIMO downlink operating over a single band. We described the RPS scheduling policy that combines maximizing the reward per second approach with ZFBF precoding to design jointly the beamformer and the scheduling algorithm. Simulations demonstrated that the proposed technique performed significantly better in this context compared to existing techniques. Thus, designing a cross-layer, joint scheduling, and precoding can significantly improve the performance of the transmission of packets with deadlines and priorities. Extension of the proposed technique to OFDM systems (here, beamforming was done independently at each frequency) is an interesting research direction. However, the insights provided by the current paper allow one to consider joint scheduling and beamforming in the downlink of OFDM systems such as 5G and WLAN.

Author Contributions: conceptualization, A.L. and L.R.; methodology, A.L. and L.R.; software, L.R.; validation, L.R.; formal analysis, L.R.; investigation, L.R.; resources, L.R.; data curation, L.R.; writing–original draft preparation, L.R.; writing–review and editing, L.R and A.L.; visualization, L.R.; supervision, A.L.; project administration, A.L.; funding acquisition, A.L..

Funding: This paper was partially supported by the Israeli Innovations Authority as part of the HERON 5G Magnet Consortium and ISF-NRF Research Grant No. 2277/2016.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Terms	
AP	Access Point
AWGN	Additive White Gaussian Noise
BC	Broadcast Channel
C-RAN	Cloud Radio Access Network
DPC	Dirty Paper Coding
EDF	Earliest Deadline First
GBC	Gaussian Broadcast Channel
LOS	Line Of Sight
MIMO	Multiple Input Multiple Output
MISO	Multiple Input Single Output
MS	Mobile Station
MCS	Modulation and Coding Scheme
MU	Multi User
MU-MIMO	Multi User MIMO
OFDM	Orthogonal Frequency-Division Multiplexing
PG	Packet Generator
QoS	Quality of Service
QSI	Queue State Information
PR	Policy Runner
RPS	Reward Per Second
SINR	Signal-to-Interference plus Noise Ratio
SDN	Software-Defined Network
SNR	Signal-to-Noise Ratio
WLAN	Wireless Local Area Network
ZFBF	Zero Forcing Beam Forming
Notations	
A_i	Arrival random variable
B_i	Packet length random variable
C	Number of servers
D_i	Deadline random variable
J_i	A packet that arrived at time t_i.
P_i	Reward or priority random variable
t_i	The time of the i^{th} arrival
K	Defines the queue prefix size for eligible packets (window size)
M	The number of antennas
N	The number of mobile stations
$\mathbf{H}$	The channel state information
$\mathbf{H}^*$	The conjugate transpose of matrix H
$\mathbf{L}$	A lower triangular matrix
$\mathcal{P}$	The transmission power
$\mathbf{Q}$	An orthogonal matrix
$\mathbf{R}$	An upper triangular matrix
$\mathbf{W}$	The beamforming matrix
$\mathbf{x}$	A vector presenting the transmitted signal
$\mathbf{y}$	A vector presenting the received signal at the MS
$\mathbf{z}$	A vector presenting the AWGN

References

1. Corson, M.S.; Laroia, R.; Li, J.; Park, V.; Richardson, T.; Tsirtsis, G. Toward proximity-aware internetworking. *IEEE Wirel. Commun.* **2010**, *17*, 26–33.
2. Maeder, A.; Rost, P.; Staehle, D. The challenge of M2M communications for the cellular radio access network. In Proceedings of the 11th Würzburg Workshop IP Joint ITG ITC Euro-NF Workshop Vis. Future Gener. Netw. (EuroView) , Würzburg, Germany, 1–2 August 2011;pp. 1–2.
3. Cisco, V. Cisco Visual Networking Index: Forecast and Trends, 2017–2022. Available online: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html (accessed on 5 August 2019).
4. Huang, X.; Xue, G.; Yu, R.; Leng, S. Joint scheduling and beamforming coordination in cloud radio access networks with QoS guarantees. *IEEE Trans. Veh. Technol.* **2015**, *65*, 5449–5460.
5. Nam, J.; Adhikary, A.; Ahn, J.Y.; Caire, G. Joint spatial division and multiplexing: Opportunistic beamforming, user grouping and simplified downlink scheduling. *IEEE J. Sel. Top. Signal Process.* **2014**, *8*, 876–890.
6. Eramo, V.; Lavacca, F.G.; Catena, T.; Polverini, M.; Cianfrani, A. Effectiveness of Segment Routing Technology in Reducing the Bandwidth and Cloud Resources Provisioning Times in Network Function Virtualization Architectures. *Future Internet* **2019**, *11*, 71.
7. Kobayashi, M.; Caire, G. Joint beamforming and scheduling for a MIMO downlink with random arrivals. In Proceedings of the 2006 IEEE International Symposium on Information Theory, Seattle, WA, USA, 9–14 July 2006; pp. 1442–1446.
8. Sun, K.; Wang, Y.; Wang, T.; Chen, Z.; Hu, G. Joint channel-aware and queue-aware scheduling algorithm for multi-User MIMO-OFDMA Systems with downlink beamforming. In Proceedings of the 2008 IEEE 68th Vehicular Technology Conference, Calgary, AB, Canada, 21–24 September, pp. 1–5.
9. Matskani, E.; Sidiropoulos, N.D.; Luo, Z.Q.; Tassiulas, L. Convex approximation techniques for joint multiuser downlink beamforming and admission control. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 2682–2693.
10. Costa, M. Writing on dirty paper (corresp.). *IEEE Trans. Inf. Theory* **1983**, *29*, 439–441.
11. Wiesel, A.; Eldar, Y.C.; Shamai, S. Zero-forcing precoding and generalized inverses. *IEEE Trans. Signal Process.* **2008**, *56*, 4409–4418.
12. Yoo, T.; Goldsmith, A. Optimality of zero-forcing beamforming with multiuser diversity. In Proceedings of the IEEE International Conference on Communications ICC 2005, Seoul, South Korea, 16–20 May 2005; Volume 1, pp. 542–546.
13. Yoo, T.; Goldsmith, A. On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 528–541.
14. Dimic, G.; Sidiropoulos, N.D. On downlink beamforming with greedy user selection: Performance analysis and a simple new algorithm. *IEEE Trans. Signal Process.* **2005**, *53*, 3857–3868.
15. Kountouris, M.; de Francisco, R.; Gesbert, D.; Slock, D.T.; Salzer, T. Low complexity scheduling and beamforming for multiuser MIMO systems. In Proceedings of the 2006 IEEE 7th Workshop on Signal Processing Advances in Wireless Communications, Cannes, France, 2–5 July 2006; pp. 1–5.
16. Karipidis, E.; Sidiropoulos, N.D.; Luo, Z.Q. Quality of service and max-min fair transmit beamforming to multiple cochannel multicast groups. *IEEE Trans. Signal Process.* **2008**, *56*, 1268–1279.
17. Huang, K.; Andrews, J.G.; Heath, R.W., Jr. Performance of orthogonal beamforming for SDMA with limited feedback. *IEEE Trans. Veh. Technol.* **2008**, *58*, 152–164
18. Castaneda, E.; Silva, A.; Gameiro, A.; Kountouris, M. An overview on resource allocation techniques for multi-user MIMO systems. *IEEE Commun. Surv. Tutorials* **2016**, *19*, 239–284. .
19. Zhang, C.; Huang, Y.; Jing, Y.; Jin, S.; Yang, L. Sum-rate analysis for massive MIMO downlink with joint statistical beamforming and user scheduling. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 2181–2194.
20. Pun, M.O.; Koivunen, V.; Poor, H.V. Performance analysis of joint opportunistic scheduling and receiver design for MIMO-SDMA downlink systems. *IEEE Trans. Commun.* **2010**, *59*, 268–280.
21. Katsinis, G.; Tsiropoulou, E.; Papavassiliou, S. Multicell interference management in device to device underlay cellular networks. *Future Internet* **2017**, *9*, 44.

22. Zafaruddin, S.M.; Bistritz, I.; Leshem, A.; Niyato, D. Multiagent Autonomous Learning for Distributed Channel Allocation in Wireless Networks. Available online: http://www.eng.biu.ac.il/leshema/files/2019/05/Multiagent-Autonomous-Learning-for-Distributed.pdf (accessed on 5 August 2019).
23. Bertsekas, D.P.; Gallager, R.G.; Humblet, P. *Data networks*; Prentice-Hall International New Jersey: Upper Saddle River, NJ, USA, 1992; Volumr 2.
24. Stankovic, J.A.; Spuri, M.; Di Natale, M.; Buttazzo, G.C. Implications of classical scheduling results for real-time systems. *Computer* **1995**, *28*, 16–25.
25. Stankovic, J.A.; Spuri, M.; Ramamritham, K.; Buttazzo, G.C. *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 460.
26. Srikant, R.; Ying, L. *Communication Networks: An Optimization, Control, and Stochastic Networks Perspective*; Cambridge University Press: Cambridge, UK, 2013.
27. Panwar, S.S.; Towsley, D.; Wolf, J.K. Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service. *J. ACM (JACM)* **1988**, *35*, 832–844.
28. Bhattacharya, P.P.; Ephremides, A. Optimal scheduling with strict deadlines. *IEEE Trans. Autom. Control* **1989**, *34*, 721–728.
29. Towsley, D.; Panwar, S. Optimality of the Stochastic Earliest Deadline Policy for the G/M/c Queue Serving Customers with Deadlines. Available online: https://pdfs.semanticscholar.org/cb01/08dca4ded7afd18e1bbbc950617749473c11.pdf (accessed on 5 August 2019).
30. Cohen, R.; Katzir, L. Scheduling of voice packets in a low-bandwidth shared medium access network. *IEEE/ACM Trans. Netw. (TON)* **2007**, *15*, 932–943.
31. Brucker, P.; Brucker, P. *Scheduling Algorithms*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 3.
32. Raviv, L.; Leshem, A. Maximizing service reward for queues with deadlines. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2296–2308, doi:10.1109/TNET.2018.2867815.
33. Grimmett, G.; Stirzaker, D. *Probability and Random Processes*; Oxford University Press: Oxford, UK, 2001.
34. Peha, J.M.; Tobagi, F.A. Cost-based scheduling and dropping algorithms to support integrated services. *IEEE Trans. Commun.* **1996**, *44*, 192–202.
35. Atar, R.; Giat, C.; Shimkin, N. The $c\mu/\theta$ rule for many-server queues with abandonment. *Oper. Res.* **2010**, *58*, 1427–1439.
36. Ayesta, U.; Jacko, P.; Novak, V. A nearly-optimal index rule for scheduling of users with abandonment. In Proceedings of the 2011 IEEE INFOCOM, Shanghai, China, 10–15 April 2011; pp. 2849–2857.
37. Yu, Z.; Xu, Y.; Tong, L. Deadline scheduling as restless bandits. 2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 27–30 September 2016; pp. 733–737.
38. Mangold, S.; Choi, S.; May, P.; Klein, O.; Hiertz, G.; Stibor, L. IEEE 802.11 e wireless LAN for quality of service. *Proc. Eur. Wirel.* **2002**, *2*, 32–39.
39. Hadar, I.; Raviv, L.; Leshem, A. Scheduling For 5G Cellular Networks With Priority And Deadline Constraints. In Proceedings of the 2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE), Eilat, Israel, 12–14 December 2018; pp. 1–5.
40. Bejarano, O.; Knightly, E.W.; Park, M. IEEE 802.11 ac: From channelization to multi-user MIMO. *IEEE Commun. Mag.* **2013**, *51*, 84–90.
41. Perahia, E.; Gong, M.X. Gigabit wireless LANs: An overview of IEEE 802.11 ac and 802.11 ad. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2011**, *15*, 23–33.
42. Bellalta, B.; Barcelo, J.; Staehle, D.; Vinel, A.; Oliver, M. On the performance of packet aggregation in IEEE 802.11 ac MU-MIMO WLANs. *IEEE Commun. Lett.* **2012**, *16*, 1588–1591.
43. Perahia, E.; Cordeiro, C.; Park, M.; Yang, L.L. IEEE 802.11 ad: Defining the next generation multi-Gbps Wi-Fi. In Proceedings of the 2010 7th IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2010; pp. 1–5.
44. Nitsche, T.; Cordeiro, C.; Flores, A.B.; Knightly, E.W.; Perahia, E.; Widmer, J.C. IEEE 802.11 ad: Directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi. *IEEE Commun. Mag.* **2014**, *52*, 132–141.
45. Ghasempour, Y.; Knightly, E.W. Decoupling beam steering and user selection for scaling multi-user 60 ghz wlans. Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Chennai, India, 10–14 July 2017; p. 10.

46. Kendall, D.G. Stochastic processes occurring in the theory of queues and their analysis by the method of the embedded Markov chain. *Ann. Math. Stat.* **1953**, *24*, 338–354.
47. Horn, R.A.; Johnson, C.R. *Matrix analysis;* Cambridge University Press: Cambridge, UK, 2012,
48. Viswanathan, H.; Venkatesan, S.; Huang, H. Downlink capacity evaluation of cellular networks with known-interference cancellation. *IEEE J. Sel. Areas Commun.* **2003**, *21*, 802–811.
49. Tse, D.; Viswanath, P. *Fundamentals of Wireless Communication*; Cambridge University Press: Cambridge, UK, 2005.
50. Gill, P.E.; Golub, G.H.; Murray, W.; Saunders, M.A. Methods for modifying matrix factorizations. *Math. Comput.* **1974**, *28*, 505–535.
51. Bojanczyk, A.; Brent, R.; De Hoog, F. QR factorization of Toeplitz matrices. *Numer. Math.* **1986**, *49*, 81–94.
52. Yoo, K.; Park, H. Accurate downdating of a modified Gram-Schmidt QR decomposition. *BIT Numer. Math.* **1996**, *36*, 166–181.
53. Sinha, R.; Papadopoulos, C.; Heidemann, J. *Internet Packet Size Distributions: Some Observations*; Tech. Rep. ISI-TR-2007-643; USC/Information Sciences Institute: Marina Del Rey, CA, USA, 2007.