# Concurrent Channel Probing and Data Transmission in Full-duplex MIMO Systems

Zhenzhi Qian
The Ohio State University
2015 Neil Ave.
Columbus, Ohio 43210
qian.209@osu.edu

Fei Wu
The Ohio State University
2015 Neil Ave.
Columbus, Ohio 43210
wu.1973@osu.edu

Zizhan Zheng
Tulane University
6823 St. Charles Ave.
New Orleans, Louisiana 70118
zzheng3@tulane.edu

Kannan Srinivasan
The Ohio State University
2015 Neil Ave.
Columbus, Ohio 43210
kannan@cse.ohio-state.edu

Ness B. Shroff
The Ohio State University
2015 Neil Ave.
Columbus, Ohio 43210
shroff.11@osu.edu

## ABSTRACT

An essential step for achieving multiplexing gain in MIMO downlink systems is to collect accurate channel state information (CSI) from the users. Traditionally, CSIs have to be collected before any data can be transmitted. Such a sequential scheme incurs a large feedback overhead, which substantially limits the multiplexing gain especially in a network with a large number of users. In this paper, we propose a novel approach to mitigate the feedback overhead by leveraging the recently developed Full-duplex radios. Our approach is based on the key observation that using Full-duplex radios, when the base-station (BS) is collecting CSI of one user through the uplink channel, it can use the downlink channel to simultaneously transmit data to other (non-interfering) users for which CSIs are already known. By allowing concurrent channel probing and data transmission, our scheme can potentially achieve a higher throughput compared to traditional schemes using Half-duplex radios. The new flexibility introduced by our scheme, however, also leads to fundamental challenges in achieving throughout optimal scheduling. In this paper, we make an initial effort to this important problem by considering a simplified group interference model. We develop a throughput optimal scheduling policy with complexity $O((N/I)^I)$, where $N$ is the number of users and $I$ is the number of user groups. To further reduce the complexity, we propose a greedy policy with complexity $O(N \log N)$ that not only achieves at least 2/3 of the optimal throughput region, but also outperforms any feasible Half-duplex solutions. We derive the throughput gain offered by Full-duplex under different system parameters and show the advantage of our algorithms through numerical studies.


*Mobihoc '17, July 10-14, 2017, Chennai, India*

ACM ISBN 978-1-4503-4912-3/17/07…$15.00
https://doi.org/10.1145/3084041.3084046

## CCS CONCEPTS

• **Networks** → *Network protocol design*; *Network control algorithms*;

## KEYWORDS

Scheduling, Full-duplex, Near-optimal throughput

**ACM Reference format:**
Zhenzhi Qian, Fei Wu, Zizhan Zheng, Kannan Srinivasan, and Ness B. Shroff. 2017. Concurrent Channel Probing and Data Transmission in Full-duplex MIMO Systems. In *Proceedings of Mobihoc '17, Chennai, India, July 10-14, 2017,* 10 pages.
https://doi.org/10.1145/3084041.3084046

## 1 INTRODUCTION

Mobile data traffic is expected to increase at rate of 53% per year by 2020 [1]. Multi-user MIMO (MU-MIMO), which can potentially increase the network capacity linearly with the number of users, has been considered as an important technique to confront this data traffic challenge. Theoretically, in a system with $M$ transmit and receive antennas, the throughput using MU-MIMO can be $M$ times of the throughput using a single transmit and receive antenna pair [20], where $M$ is commonly referred as the spatial multiplexing gain.

In this paper, we consider one important application of MU-MIMO, i.e., the downlink wireless cellular network consisting of one Base Station (BS) equipped with many antennas and many users each equipped with one antenna. In such systems, the BS could utilize MU-MIMO to transmit multiple data streams to multiple users simultaneously. Nevertheless, to take the advantage of MU-MIMO in practice, it is prerequisite for the transmitter to learn the accurate channel state information (CSI) of the users [14]. Note that in traditional wireless networks, radios can only operate in Half-duplex (HD) mode, i.e., a radio cannot transmit and receive packets on the same frequency at the same time. As a result, traditional schemes to harness the multiplexing gain of MU-MIMO, e.g., [18, 24], require that the channel state information (CSI) of the users have to be learned first before any data can be transmitted. Such a sequential channel learning scheme incurs a large overhead when there are a large number of users, which would in turn substantially limit the multiplexing gains of MU-MIMO, especially if the channel

coherence time is relatively short [18, 24], *the large channel learning overhead has been a long-standing open problem which limits the achievable throughput of MU-MIMO in practice.*

Recently, Full-duplex (FD) radios [5, 6, 9] have been developed, which allow simultaneous transmission and reception on the same frequency. The availability of Full-duplex provides significant flexibility in designing wireless resource allocation algorithms. For example, it has been shown that in some cases [22], Full-duplex can almost double the throughput and effectively improve spectrum efficiency. This leads to the following natural and important question: *Is it possible to leverage Full-duplex to address the feedback overhead challenge in Multi-user MIMO downlink systems?*

In this paper, we answer this question in the affirmative. By using a Full-duplex BS, we are able to break the boundary between the channel learning phase and the data transmission phase. As shown in Fig. 1, the BS receives the channel probing signal from Alice in round 1 and measures the downlink channel to Alice assuming channel is reciprocal[1]. Then in round 2, the BS uses Full-duplex capability to send data to Alice and receive the probing signal from Bob simultaneously, assuming Bob does not interfere with Alice. After the BS measures all downlink channels, the BS operates in MU-MIMO mode in round 3. Compared to Half-duplex systems, once the BS knows the downlink channel to Alice, it can start transmission immediately rather than waiting until the end of the channel learning phase. Henceforth, we will refer to this concept as *concurrent channel probing and data transmission.*

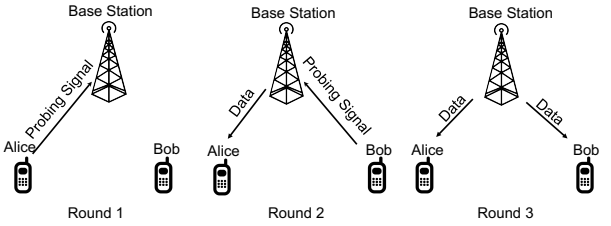



**Figure 1: Concurrent channel probing and data transmission.**

Due to the interference between users, the performance of concurrent channel probing and data transmission scheme depends highly on the set of users selected to send probing signals and the ordering of these users. Therefore, the following important question remains: *How do we design a low-complexity scheduling policy that achieves provably good throughput performance under the concurrent channel probing and data transmission?*

While the design of high performance scheduling policies have been extensively studied in traditional wireless systems [13], relatively few efforts [23] have focused on the scheduling problem in Full-duplex systems. In particular, it is much more challenging to consider this problem under concurrent channel probing and data transmission. The reason is that: 1) The ordering of users sending probing signal matters. A user that sends a probing signal earlier also starts transmission earlier. 2) Within one channel coherence time, the scheduling decisions are coupled in terms of time and interference relations. The rate received by a certain user depends on what time it transmits the probing signal as well as the interference relations with the users scheduled to send probing signals later. These two facts make the scheduling problem more complicated and classical scheduling policies do not apply here. In this paper, we aim to develop a throughput near-optimal scheduling policy and investigate the Full-duplex gain for a various of network settings.

The key contributions of this paper are summarized as follows:

- We develop a scheduling policy that achieves the optimal throughput region under concurrent channel probing and data transmission. Compared to Brute-Force search, the complexity has been decreased from $O(N!)$ to $O((N/I)^I)$.
- To further reduce the scheduling complexity in large systems, we design a low-complexity greedy policy with complexity $O(N \log N)$ that not only achieves at least 2/3 of the optimal throughput region but also outperforms any feasible Half-duplex solutions. We conjecture that the real performance of the greedy policy is very close to the optimal, which is confirmed by simulations.
- We derive the throughput gain offered by Full-duplex under different system parameters and use simulations to validate our theoretical results.

The rest of the paper is organized as follows. We discuss related works in Section 2. In Section 3, we describe the system model and problem formulation. In Section 4, we develop a throughput optimal policy which stabilizes the system under any feasible arrival rates. In Section 5, we design a low-complexity greedy policy and provide provable performance guarantees. In Section 6, we derive the Full-duplex gain under different network settings and system parameters. We conduct simulations to validate our theoretical results in Section 7 and make concluding remarks in Section 8.

## 2 RELATED WORK

In-band Full-duplex, as an emerging technology in wireless communication, was implemented by combining RF and baseband interference cancellation [5, 6, 9], enabling simultaneous bi-directional transmission between a pair of nodes. Full-duplex has now been widely studied in a number of wireless communication scenarios. Full-duplex WiFi-PHY based MIMO radios was first implemented in [4], and experiments showed that the theoretical doubling of throughput is practically achieved. While it is hard to make Full-duplex MIMO radios fit in small personal devices, it is feasible to build a Full-duplex MIMO Base Station due to bigger size and more powerful computational ability [11]. In [7, 8], the authors proposed the continuous feedback channel, which enables sequential beamforming that update weights while also performing downlink transmission. The authors showed that the system outperforms its Half-duplex counterpart and reduced the control overhead at the same time. This work can be viewed as an preliminary attempt of the idea of concurrent channel probing and data transmission. However, the authors assumed that users are symmetric and did not consider the scheduling problem, which is the focus of our study here.

In addition to the research efforts focused on implementation and experiments, there have also been several theoretical works on Full-duplex systems. Although Full-duplex is expected to double the capacity in single pair of nodes, [21] showed that the inter-link

[1]Measuring downlink channel to a user through channel probing from the user is standard in a time division duplex (TDD) system [18, 24].

interference and spatial reuse substantially reduces network-level Full-duplex gain, making it less than 2 in typical cases. In order to deal with the increasing inter-link interference, [17] presented a new interference management strategy to achieve a larger rate gain over Half-duplex systems. The capacity region of multi-channel Full-duplex links was characterized in [15] and rate gain is illustrated for various channel and cancellation scenarios. The authors in [22] also investigated the achievable throughput performance of MIMO, Full-duplex and their variants that allow simultaneous activation of two RF chains. The scheduling problem in Full-duplex cut-through transmission was considered in [23], where the authors characterized the interference relationship between links in the network with cut-through transmission and designed a Q-CSMA type of scheduling algorithm to leverage the flexibility of Full-duplex cut-through transmission. In contrast to the aforementioned works, this is the first work that considers the scheduling problem under concurrent channel probing and data transmission and provides analytical framework to characterize the network-level Full-duplex gain.

## 3 SYSTEM MODEL

We consider the downlink phase of a single-cell Full-duplex MIMO system. There are $N$ users in this system and each of them is equipped with only one antenna. The Base Station (BS) has multiple antennas and Full-duplex capability. In addition, we assume time is slotted and we consider a discrete-time system. We use $\mathcal{N}$ denote the set of all users in the system.

### 3.1 Channel Model

We consider a block fading channel, where the channel state remains the same within each time-slot, but may vary from time-slot to time-slot. We assume channel state information (CSI) is only available at the user side at the beginning of each time-slot. In order to fully achieve the multiplexing gain of MU-MIMO, the BS needs to collect CSI via feedback through the uplink channel. We assume that channels are reciprocal, in which case a user could send a probing signal on its single antenna and the BS, by measuring on its antennas, learns the downlink CSI. Any CSI expires by the end of the current time-slot, and it has to be learned again in the next time-slot. In practice, collecting CSI from multiple users takes time and its overhead is linear with respect to the number of the corresponding users. We assume that in one time-slot, the transmitter can collect CSI from at most $K$ users. Therefore, each time-slot can be further divided into $K$ mini-slots and it takes one mini-slot to learn each CSI. The BS can only transmit one packet per mini-slot to each user whose channel information is already known.

In traditional Half-duplex systems, CSI collection and data transmission must be separated in time to avoid interference. Data transmission phase starts only if all desired CSIs are collected. Full-duplex systems, on the other hand, allows data transmission immediately after each CSI is collected.

### 3.2 User Groups

Full-duplex capability does not always offer "free lunch", its performance suffers from complex interference patterns. One way to characterize interference is using user groups which guarantee no inter-group interference. Thus, we can break the scheduling problem into two steps: 1) Given $N$ users, how to divide them into different user groups. 2) Given group information, how to find a scheduling policy that achieves good throughput performance. Dividing users into groups is not easy due to the conflict between interference constraints and the desire to have more groups and less users in each group. We focus on the second step in this work and leave the joint problem as the future work. The problem is still challenging even when the group information is already given.

Assume $N$ users are split into $I$ user groups, which guarantees no inter-group interference. For example, suppose user $u_i$ and $u_j$ are from different groups, the uplink stream of user $u_i$ does not interfere with the downlink stream of user $u_j$. Based on each user's geographical statistics, the group information will be determined once over a much larger time scale. The group information is assumed to be static and remains the same in a time-slot. Fig. 2 is an illustration of a downlink system with 2 user groups. We use $g(u)$ to denote the group index of user $u$, and let $\mathcal{G}_{g(u)}$ denote the set of users in group $g(u)$.

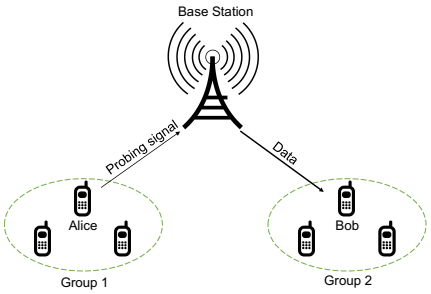



**Figure 2: A downlink system with 2 user groups, the BS receives probing signal from Alice and transmits data packets to Bob (channel is already known) simultaneously.**

### 3.3 Traffic Model

The BS maintains a queue $Q_u$ to store packets requested by each user $u$. The arrival process to each queue is assumed to be stationary and ergodic. We assume packet arrival and departure both occur at the beginning of each time-slot. Let $A_u[t]$ denote the number of packet arrivals to queue $Q_u$ in time-slot $t$. Let $R_u[t]$ denote the downlink rate to queue $Q_u$ in time-slot $t$. The queue-length $Q_u[t]$ evolves as:

$$Q_u[t+1] = \max\{Q_u[t] + A_u[t] - R_u[t], 0\}. \quad (1)$$

### 3.4 Scheduling Policy

In each time-slot $t$, a scheduling policy $P$ determines the schedule based on the system state, e.g., queue-length and delay. Such schedule can be described as a scheduling vector $\mathbf{f} = (u_1, \cdots, u_K)$, which indicates that user $u_i$ sends a probing signal in the $i^{th}$ mini-slot. $u_i = 0$ implies that the BS is only transmitting, not learning any channel in the $i^{th}$ mini-slot. "0" element is also considered as a dummy user from a dummy group with zero queue-length. Due to interference constraints, once the BS chooses to learn user $u$'s channel during the $i^{th}$ mini-slot, it will block all other users in $\mathcal{G}_{g(u)}$ from receiving any packet. However, the BS can transmit data packets to users from other groups since there is no interference

between these groups. We use $R^{\mathbf{f}}_{u_i}$ to denote the downlink rate to user $u_i$ under scheduling vector $\mathbf{f}$. For all $i = 1, \ldots, K$, $R_{u_i}[t] = R^{\mathbf{f}}_{u_i}$ if scheduling vector $\mathbf{f}$ is adopted in time-slot $t$. From now on, we omit the subscript $[t]$ when looking into the schedule made in a certain time-slot $t$. Note that $R^{\mathbf{f}}_{u_i}$ is the number of mini-slots from $i+1$ to $K$ such that the group of the scheduled user is different from group $g(u_i)$, i.e., $R^{\mathbf{f}}_{u_i} = \sum_{j=i+1}^{K} \mathbf{1}_{\{g(u_i) \neq g(u_j)\}}$. For example, if $\mathbf{f} = (u_a, u_b, u_c, 0, \cdots, 0)$ and $g(u_a) = g(u_b) \neq g(u_c)$. From the second mini-slot to the $K^{th}$ mini-slot, there are $K-2$ users in $\mathbf{f}$ such that its group is other than $g(u_a)$. Thus, $R^{\mathbf{f}}_{u_a} = K-2$. Similarly, we have $R^{\mathbf{f}}_{u_b} = K-2$ and $R^{\mathbf{f}}_{u_c} = K-3$. Denote the set of feasible scheduling policies as $\Pi$.

In this paper, we mainly focus on the throughput performance of the system. First we define the *optimal throughput region* for any given system parameters $N$ and $K$. As in [3, 12], a stochastic queueing network is said to be stable if it behaves as a discrete-time countable Markov chain and the Markov chain is *stable* in the following sense: 1) The set of positive recurrent states is non-empty. 2) It contains a finite subset such that with probability one, this subset is reached within finite time from any initial state. When all the states communicate, stability is equivalent to the Markov chain being positive recurrent [16]. The *throughput region* $\Lambda^P$ of a scheduling policy $P$ is defined as the set of arrival rate vectors for which the network remains stable under this policy.

*Definition 3.1.* (*Optimal throughput region*) The optimal throughput region is defined as the union of the throughput regions of all possible scheduling policies, which is denoted by $\Lambda^*$, i.e.,

$$\Lambda^* = \bigcup_{P \in \Pi} \Lambda^P. \tag{2}$$

*Definition 3.2.* (*Throughput optimal policy*) A scheduling policy is throughput-optimal if it can stabilize any arrival rate vector strictly inside $\Lambda^*$.

## 4 OPTIMAL SCHEDULING POLICY

In this section, we propose a throughput-optimal scheduling policy to the concurrent probing and transmission problem. We first observe that the following classic result applies to our setting as well.

Theorem 4.1. *Any policy that maximizes the weight* $w(\mathbf{f}) = \sum_{u \in \mathcal{N}} Q_u R^{\mathbf{f}}_u$ *in each time-slot, a.k.a., the MaxWeight scheduling policy, is throughput-optimal.*

Proof. Please refer to the proof in [19]. □

From the theorem, it suffices to find a scheduling vector $\mathbf{f}^*$ such that the weight $w(\mathbf{f})$ is maximized in each time-slot, i.e.,

$$\mathbf{f}^* = \arg\max_{\mathbf{f}} \sum_{u \in \mathcal{N}} Q_u R^{\mathbf{f}}_u. \tag{3}$$

However, it is not trivial to find a MaxWeight schedule with low complexity. We note that for traditional wireless scheduling under 1-hop interference, MaxWeight scheduling boils down to finding a maximum weighted matching in each time-slot, which can be done in $O(N^3)$ where $N$ is the number of nodes. This result does not apply to our setting, however, since the ordering of users sending probing signal matters. A Brute-Force search enumerates all possible permutations of users, leading to a high complexity of $O(N!)$, which is infeasible when $N$ is large. Thus, an interesting question is how to find a MaxWeight schedule in our setting in a more efficient way. To this end, we propose the following algorithm with complexity $O((N/I)^I)$ (polynomial when $I$ is a constant regardless of $N$). In the algorithm, $m_i$ indicates the number of users to be chosen from group $i$, $1 \leq i \leq I$, and $\mathbf{m} = (m_1, \cdots, m_I)$ is the user-selection vector. Algorithm 1 will be applied to each time-slot to generate the MaxWeight schedule.

**Algorithm 1** Search algorithm for MaxWeight Schedule

**Input:** For all $u \in \mathcal{N}$, group $g(u)$ and queue-length $Q_u$.
**Output:** Scheduling vector $\widehat{\mathbf{f}}$

```
Initialization: User-selection vector m = (0, 0, ··· , 0),
  ŵ = 0, f̂ = (0, 0, ··· , 0).
for all m such that Σ_i m_i ≤ K do
    Set scheduling vector f = (0, 0, ··· , 0).
    Set scheduled user set U = ∅
    for i=1, 2, ··· , I do
        Add m_i users with longest queue-length from group i to
        U.
    Fill in scheduling vector f with users in U,
    following the Longest Queue-length First order.
    if w(f) > ŵ then
        ŵ = w(f)
        f̂ = f
return f̂
```

For a given user-selection vector $\mathbf{m}$, Algorithm 1 picks $m_i$ users from group $i$ with longest queue-length, for all $i = 1, 2, \cdots, I$. It then generates a candidate scheduling vector $\mathbf{f}$ by filling in users following the Longest Queue-length First (LQF) order. The weight $w(\mathbf{f})$ is evaluated for all possible user-selection vectors $\mathbf{m}$ and its resulting scheduling vector, Algorithm 1 returns the scheduling vector $\widehat{\mathbf{f}}$ that has the maximum weight.

Theorem 4.2. *The schedule* $\widehat{\mathbf{f}}$ *returned by Algorithm 1 maximizes weight* $w(\mathbf{f})$.

Proof. We divided the proof into two steps. For the first step, we show that the LQF order maximizes the weight for a given scheduled user set. Then for the user-selection part, we show that it is sufficient to evaluate all possible user-selection vectors $\mathbf{m}$ and its resulting scheduled user set by adding $m_i$ users with longest queue-length from each group $i$. The proof details are provided in our technical report [2].

Given $\mathbf{m}$, the schedule yields maximum weight is determined by: (1) For each group $i$, add $m_i$ users with longest queue-length into the scheduled user set $U(\mathbf{m})$. (2) Schedule the users from $U(\mathbf{m})$ following the LQF order. Thus, traversing all possible $\mathbf{m}$ will return the MaxWeight schedule. And this proves the optimality of Algorithm 1. □

Remark 4.2.1. *Applying LQF to the set of all users does not guarantee the maximum. Since LQF is a myopic rule, it always gives higher priority to users with longer queue-length regardless of their interference relations. In fact, queue-length and interference relations both*

*play a key role in this problem, and we need to do user-selection to get a good balance between these two factors.*

## 5 A LOW-COMPLEXITY GREEDY POLICY

Although Algorithm 1 returns throughput optimal policy in polynomial time, the complexity $O((N/I)^I)$ grows very high when the number of groups $I$ is large. It is interesting to see whether there is any low-complexity policy that achieves provably good throughput. In this section, we propose a greedy algorithm which incrementally adds users to the schedule and prove that it achieves at least 2/3 of the optimal throughput region. In addition, our proposed greedy policy always achieves a larger throughput region than any scheduling policies under Half-duplex.

### 5.1 Greedy Algorithm Description

*Definition 5.1.* (*Marginal Gain*) Given a schedule $\mathbf{f} = (u_1, \cdots, u_Q, 0, \cdots, 0)$ and a user $u$ that is a candidate user to be considered in $j^{th}$ mini-slot (when evaluating user $u$, the first $j-1$ scheduled users have already been determined in $\mathbf{f}$), the marginal gain $\Delta_u^{\mathbf{f},j}$ is defined to be the weight difference caused by adding user $u$ as the $j^{th}$ element of $\mathbf{f}$, assuming there are no future scheduled users, i.e., $\Delta_u^{\mathbf{f},j} = w\left((u_1, \cdots, u_{j-1}, u, 0, \cdots, 0)\right) - w\left((u_1, \cdots, u_{j-1}, 0, \cdots, 0)\right)$.

To evaluate the marginal gain of adding user $u$ to the schedule $\mathbf{f}$, we must consider the benefit as well as the cost. The benefit is obvious, we have one more user and it keeps transmitting packets until the end of the current time-slot, i.e., receives a rate of $K-j$. Hence its weight contribution is $Q_u(K-j)$. On the other hand, if we schedule user $u$ in $j^{th}$ mini-slot, it will block the transmission of the previously scheduled users that are from the same group $g(u)$. Thus, the weight loss is $\sum_{i=1}^{j-1} Q_{u_i} \mathbf{1}_{\{g(u_i)=g(u)\}}$. Therefore, we have:

$$\Delta_u^{\mathbf{f},j} = Q_u(K-j) - \sum_{i=1}^{j-1} Q_{u_i} \mathbf{1}_{\{g(u_i)=g(u)\}}. \tag{4}$$

A positive marginal gain means that by adding a new user, the weight will not be decreased. Marginal gain considers queue-length as well as the group information and is able to discriminate different cases (e.g., long queue-length & strong interference v.s. short queue-length & weak interference). Although the marginal gain is not the actual gain of user $u_j$ since we do not know the future scheduled users, it is still a good metric to evaluate the potential gain of adding one candidate user to the current schedule. Moreover, as we will soon see, the Marginal Gain-based Greedy (MGG) Algorithm achieves good throughput performance.

The MGG Algorithm, inspired by Section 4, we first sort users according to their queue-lengths, and then start from the user that has the longest queue-length in the system, the MGG Algorithm iteratively evaluates the user $u$ with next longest queue-length. The MGG Algorithm will add user $u$ if its marginal gain is positive, otherwise skip user $u$ and continue to evaluate the user with the next longest queue-length until $K$ users have been scheduled or all $N$ users are all evaluated.

The complexity of Algorithm 2 is at most $O(N \log N)$ (comes from the sorting operation), regardless of the value $I$ takes. Compared to Algorithm 1, Algorithm 2 uses LQF and marginal gain to efficiently select valuable users. Again, applying LQF only would work poorly, since it only gives higher priority to those users with longer queue-length rather than large marginal gain. In fact, the inter-user interference is very important and should not be ignored.

**Algorithm 2** Marginal Gain-based Greedy Algorithm

**Input:** $\forall$ user $u \in \mathcal{N}$, group $g(u)$ and queue-length $Q_u$.
**Output:** Scheduling vector $\mathbf{f}^G$

```
Initialization: f^G = (0, ··· , 0)
Initialization: index = 1
Sort queue-length, assume Q_{u_1} ≥ Q_{u_2} ≥ ··· Q_{u_N}
for all i from 1 to N do
    if index ≤ K then
        if Δ^{f^G, index}_{u_i} ≥ 0 then
            Add user u_i to f^G as the index^{th} element
            index = index + 1
return f^G
```

### 5.2 Performance Analysis

The MGG Algorithm is simple, however it sacrifices some throughput performance. In this section, we aim to provide a theoretical worst-case lower bound on its throughput performance.

Theorem 5.2. *The Greedy Algorithm 2 stabilizes at least 2/3-fraction of the arrival vector on the optimal throughput region, i.e., achieves 2/3 of the optimal throughput region.*

Proof. From [10], we know that it suffices to show that $w(\mathbf{f}^G) \geq 2/3 w(\mathbf{f}^*)$, where $\mathbf{f}^*$ is the MaxWeight schedule. Consider the users selected by $\mathbf{f}^G$ and $\mathbf{f}^*$. Let $\mathcal{A}$ denote the set of users shared by both schedules, let $\mathcal{B}$ denote the set of users only scheduled in $\mathbf{f}^*$ and let $C$ denote the set of users only scheduled in $\mathbf{f}^G$.

Remark 5.2.1. *The MaxWeight schedule is not necessarily unique, but these schedules have the same weight. We can choose any of these schedules to be schedule $\mathbf{f}^*$ here.*

Remark 5.2.2. *In practice, users in $\mathcal{B}$ could interfere with users in $\mathcal{A}$. Here in the proof, we aim to show a stronger claim which assumes that in the MaxWeight schedule, users from $\mathcal{B}$ do not interfere with users in $\mathcal{A}$ and $\mathcal{B}$ itself.*

*Definition 5.3.* (*Extra weight*) Extra weight $\epsilon$ is defined to be the weight loss in the MGG schedule caused by interference from users in $C$. That is to say, the total weight $w(\mathbf{f}^G) + \epsilon$ is calculated as if there is no interference caused by users in $C$, adding each user in $C$ does not block the downlink transmission of all the scheduled users which are from the same group.

We divide the proof into two parts, for the first part, we show that $w(\mathbf{f}^G) + \epsilon \geq w(\mathbf{f}^*)$. Then we show that $\epsilon \leq 1/2 w(\mathbf{f}^G)$. Combining both parts, we know $w(\mathbf{f}^G) \geq 2/3 w(\mathbf{f}^*)$, which concludes the proof.

**Part 1** In this part, we want to show that $w(\mathbf{f}^G) + \epsilon \geq w(\mathbf{f}^*)$, which means the weight of the MGG schedule by ignoring the interference caused by users in $C$ is greater than the weight of the MaxWeight schedule. The following lemmas illustrate the relationship between the MGG schedule and MaxWeight schedule, and these results will be used later.

Lemma 5.4. *Consider the MaxWeight schedule* $\mathbf{f}^* = (u_1^*, \cdots, u_\Omega^*, 0, \cdots, 0)$. *For each* $1 \le i \le \Omega$, *the marginal gain* $\Delta_{u_i^*}^{\mathbf{f}^*, i} \ge 0$.

Proof. Please see our technical report [2]. □

Remark 5.4.1. *Similar to the MGG schedule generated by Algorithm 2, the MaxWeight schedule adds a user only if the marginal gain is non-negative. The only difference is that the MGG schedule will give higher priority to users with longer queue-length, whereas the MaxWeight schedule may skip some users with long queue lengths and choose other users with large marginal gain.*

In the MaxWeight schedule, for each user $u \in \mathcal{A} \cup \mathcal{B}$, we use $t_1(u)$ to denote the mini-slot that user $u$ is scheduled. In the MGG schedule, for each user $u \in \mathcal{N}$ we define $t_2(u)$ to be the mini-slot that its marginal gain is evaluated (either schedule $u$ or skip $u$ in $t_2(u)^{th}$ mini-slot), if $u$ has never been considered as a candidate, $t_2(u) = K$.

Lemma 5.5. *In the MaxWeight schedule, for each* $b \in \mathcal{B}$, *consider user* $d$ *which has the longest queue-length among all users in group* $g(b)$ *that are not scheduled in the MGG schedule. We have:* $t_1(b) < t_2(d)$, *i.e.,* $b$ *is scheduled earlier in the MaxWeight schedule than the time that* $d$ *is skipped in the MGG schedule.*

Proof. Please see our technical report [2]. □

Define $N_{\mathcal{B}}(t)$ and $N_C(t)$ to be the number of users in $\mathcal{B}$ and $C$ scheduled in the MaxWeight and MGG schedule from the first mini-slot to $t^{th}$ mini-slot. We have the following lemma:

Lemma 5.6. *For each* $b \in \mathcal{B}$, *which is scheduled in* $t_1(b)^{th}$ *mini-slot, we have* $N_{\mathcal{B}}(t_1(b)) \le N_C(t_1(b))$.

Proof. Please see our technical report [2]. □

From Lemma 5.6, we can find a ***mapping*** $h : \mathcal{B} \to C$, $i^{th}$ user $b_i$ in $\mathcal{B}$ corresponds to $i^{th}$ user $c_i$ in $C$, such that $c_i$ is always scheduled earlier than $b_i$, i.e., $t_1(b_i) \ge t_2(c_i)$. For each user $b_i$, consider user $d_i$ which has the longest queue-length among all users in group $g(b_i)$ that are not scheduled in the MGG schedule. Note that users from group $g(b_i)$ only belongs to $\mathcal{A}$ or $\mathcal{B}$, user $d_i$ has the longest queue-length among all users in $\mathcal{B} \cap \mathcal{G}_{g(b_i)}$, thus $Q_{d_i} \ge Q_{b_i}$. From Lemma 5.5, we know $t_1(b_i) < t_2(d_i)$ and thus $t_2(c_i) < t_2(d_i)$. Then $Q_{c_i} \ge Q_{d_i}$ due to the LQF order of evaluating users in the MGG policy. Therefore, $Q_{c_i} \ge Q_{b_i}$.

Lemma 5.7. *The MGG schedule will schedule more users than the MaxWeight schedule, i.e.,* $|\mathcal{B}| \le |C|$.

Proof. Please see our technical report [2]. □

Now we are ready to prove the result of part 1. Compare $w(\mathbf{f}^G)+\epsilon$ with $w(\mathbf{f}^*)$, we have two kinds of losses.

$\mathcal{A}$ **loss:** For each user $a \in \mathcal{A}$, $a$ will be scheduled no earlier in the MGG schedule than that in the MaxWeight schedule, i.e., $t_1(a) \le t_2(a)$ (corollary of Lemma 5.6). Each user $a$ in the MGG schedule will receive lower or equal rate than that in the MaxWeight schedule.

$\mathcal{B}$ **loss:** In the MGG schedule, there is no weight contributed by users in $\mathcal{B}$.

If the total weight of the users in $C$ can be used to cover $\mathcal{A}$ and $\mathcal{B}$ losses, then $w(\mathbf{f}^G) + \epsilon \ge w(\mathbf{f}^*)$ holds. First, we consider $\mathcal{A}$ loss: let $Loss_{a_i}$ denote the weight loss on user $a_i$.

$$\begin{aligned} Loss_{a_i} &= Q_{a_i}(K - t_1(a_i) \\ &\quad - |\{a \in \mathcal{A} | a \text{ is scheduled after } a_i \text{ in } \mathbf{f}^*\}|) \\ &\quad - Q_{a_i}(K - t_2(a_i) \\ &\quad - |\{a \in \mathcal{A} | a \text{ is scheduled after } a_i \text{ in } \mathbf{f}^G\}|) \\ &= Q_{a_i}(t_2(a_i) - t_1(a_i)) \ge 0. \end{aligned} \tag{5}$$

Similarly, we use $Loss_{b_i}$ to denote the weight loss on user $b_i$:

$$Loss_{b_i} = Q_{b_i}(K - t_1(b_i)) \ge 0. \tag{6}$$

The weight difference $w(\mathbf{f}^G) + \epsilon - w(\mathbf{f}^*)$ is the total weight of $C$ minus $\mathcal{A}$ loss and $\mathcal{B}$ loss:

$$\begin{aligned} & w(\mathbf{f}^G) + \epsilon - w(\mathbf{f}^*) \\ &= \sum_{i=1}^{|C|} Q_{c_i}(K - t_2(c_i)) - \sum_{i=1}^{|\mathcal{A}|} Loss_{a_i} - \sum_{i=1}^{|\mathcal{B}|} Loss_{b_i} \\ &= \sum_{i=1}^{|C|} Q_{c_i}(K - t_2(c_i)) - \sum_{i=1}^{|\mathcal{A}|} Q_{a_i}(t_2(a_i) - t_1(a_i)) \\ &\quad - \sum_{i=1}^{|\mathcal{B}|} Q_{b_i}(K - t_1(b_i)) \\ &\overset{(d)}{\ge} \sum_{i=1}^{|\mathcal{B}|} Q_{c_i}(t_1(b_i) - t_2(c_i)) + \sum_{i=|\mathcal{B}|+1}^{|C|} Q_{c_i}(K - t_2(c_i)) \\ &\quad - \sum_{i=1}^{|\mathcal{A}|} Q_{a_i}(t_2(a_i) - t_1(a_i)) \\ &\overset{(e)}{=} \sum_{i=1}^{|C|} Q_{c_i}(t_1(b_i) - t_2(c_i)) - \sum_{i=1}^{|\mathcal{A}|} Q_{a_i}(t_2(a_i) - t_1(a_i)). \end{aligned} \tag{7}$$

where inequality (d) comes from the property of mapping $h$ and equation (e) is derived by setting $t_1(b_i) = K$ for any dummy user $b_i$, $|\mathcal{B}| < i \le |C|$. Note that for each $i$, $t_1(b_i) - t_2(c_i) \ge 0$ and $t_2(a_i) - t_1(a_i) \ge 0$.

Lemma 5.8. *The R. H. S. of (7) is non-negative.*

Proof. Please see APPENDIX A. □

The result of Lemma 5.8 concludes the proof of part 1.

**Part 2** In this part, we want to show that $\epsilon \le 1/2 w(\mathbf{f}^G)$, i.e., the extra weight is upper bounded by one half of the weight of the MGG schedule. We use $\epsilon_i$ and $w_i(\mathbf{f}^G)$ to denote the extra and actual weight from group $i$. It suffices to show a stronger (per-group) claim: For each group $i$, we have $\epsilon_i \le 1/2 w_i(\mathbf{f}^G)$.

For each group $i$, note that we only need to consider the worst case where all the users from group $i$ are in $C$. Otherwise, assume there are some users in $\mathcal{A}$, then $w_i(\mathbf{f}^G)$ remains the same while $\epsilon_i$ is smaller.

Lemma 5.9. *Assume in the MGG schedule, we have* $m$ *users* $(u_1, \cdots, u_m$, *with queue-length* $Q_{u_1} \ge \cdots \ge Q_{u_m})$ *from group* $i$, *define* $T_m$ *to be the smallest rate of the last scheduled user such that the MGG schedule is feasible (marginal gain is always non-negative). Consider*

*the case $K = K_m \triangleq T_m + t_2(u_m)$, we have $\epsilon_i^{K_m} \leq 1/2 w_i\left(\mathbf{f}^G_{K_m}\right)$, where $\epsilon_i^{K_m}$ and $w_i\left(\mathbf{f}^G_{K_m}\right)$ are extra weight and actual weight of $\mathbf{f}^G$ from group $i$ under $K_m$.*

Proof. Please see APPENDIX B. □

Note that $K_m$ is the smallest value of $K$ such that the MGG schedule is feasible, for any $K \geq K_m$, extra weight $\epsilon_i$ will be the same since it is only related to $u_1, \cdots, u_m$, however, $w_i(\mathbf{f}^G)$ will increase with $K$.

$$\frac{\epsilon_i^K}{w_i(\mathbf{f}^G_K)} \leq \frac{\epsilon_i^{K_m}}{w_i(\mathbf{f}^G_{K_m})} \leq 1/2. \tag{8}$$

Therefore, we know for every feasible MGG schedule, $\epsilon_i / w_i(\mathbf{f}^G)$ is less than one half for any group $i = 1, \cdots, I$. We finish the proof of part 2 and now we are able to show $w(\mathbf{f}^G) \geq 2/3 w(\mathbf{f}^*)$. □

Proposition 5.10. *The 2/3 worst-case lower bound is tight in terms of weight.*

Proof. Assume $K = 2^r$ for some positive integer $r > 0$. All the users have the same queue-length, and there are $K - 1$ groups where each group has sufficiently many users. Then the MaxWeight schedule will serve $K - 1$ users, one for each group, which gives a total rate of $K(K-1)/2$, while the MGG Algorithm serves $K/2$ users from group 1, $K/4$ users from group 2, $\cdots$ and 1 user from group $r$, which gives a total rate of $(K^2 - 1)/3$. As $K \to \infty$, the efficiency ratio becomes arbitrarily close to 2/3. □

Theorem 5.11. *The throughput region of the proposed MGG policy is no smaller than the optimal throughput region under Half-duplex.*

Proof. We first prove the following lemma, which shows that the weight of MGG policy dominates the weight of any Half-duplex policy.

Lemma 5.12. *The weight of the MGG policy is no smaller than the maximum weight under Half-duplex, i.e., $w(\mathbf{f}^G) \geq w^*_{HD}$, where $w_{HD}(\cdot)$ is the total weight calculated under Half-duplex.*

Proof. Please see our technical report [2]. □

Now we need to show that the MGG policy stabilizes any arrival vector $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_n)$ within the optimal throughput region under Half-duplex $\Lambda^*_{HD}$. The following lemma can be used to prove this claim.

Lemma 5.13. *Consider the capacity region $\Lambda_{HD}$ under Half-duplex, $w^*_{HD}$ is the maximum weight among all feasible scheduling policies under Half-duplex. If there exists a Full-duplex scheduling policy $\mathbf{f}^G$, such that $w(\mathbf{f}^G) \geq w^*_{HD}(\mathbf{f})$ for any queue-length vector, then policy $\mathbf{f}^G$ can stabilize any arrival vector within $\Lambda^*_{HD}$.*

Proof. Please see our technical report [2]. □

Applying Lemma 5.12 and 5.13, Theorem 5.11 follows. □

Remark 5.13.1. *Other promising low-complexity algorithms, such as greedily select users with the largest marginal gain or simply adopt certain amount of users from each group cannot work well either in the comparison with traditional Half-duplex schemes or under heterogeneous traffic arrivals.*

# 6 CAPACITY GAIN OF FULL-DUPLEX OVER HALF-DUPLEX

In this section, we will discuss the capacity gain of Full-duplex over Half-duplex. Let $\Lambda_{FD}$ and $\Lambda_{HD}$ denote the capacity region under Full-duplex and Half-duplex mode, respectively. To simplify, we only evaluate the capacity magnitude $\nu_{FD}$ and $\nu_{HD}$ along the $(1, \cdots, 1)$ vector (e.g., $(\nu_{FD}, \cdots, \nu_{FD})$ is the largest arrival vector such that all users have the same arrival rate and the queuing system can be stabilized under Full-duplex mode). In addition, we assume all groups have the same size, i.e., $N_1 = \cdots = N_I = N/I$.

For Half-duplex, if the sum-rate is upper bounded by $B_{HD}$, then the lowest service rate is upper bounded by $B_{HD}/N$. According to the basic queuing theory, $\nu_{HD} \leq B_{HD}/N$. The sum-rate is calculated by:

$$\sum_{i=1}^{N} R_i^{HD} = \left(K - \sum_{j=1}^{I} m_j\right) \sum_{j=1}^{I} m_j. \tag{9}$$

where $m_j$ is the $j^{th}$ element in the user-selection vector. If $N \geq K/2$, the maximum of the sum-rate is achieved by taking $\sum_{j=1}^{I} m_j = K/2$, thus the upper bound $B_{HD} = \frac{K^2}{4}$. Otherwise, if $K$ is larger, the maximum is achieved by scheduling all users in the system, $B_{HD} = (K - N)\, N$. To sum up,

$$\nu_{HD} = \begin{cases} \frac{K^2}{4N}, & N \geq K/2, \\ K - N, & \text{otherwise.} \end{cases} \tag{10}$$

Next, we will look at the Full-duplex case, consider a randomized policy $P$ which uses random schedules from time-slot to time-slot, denote its sum-rate as $B_{FD}$. Since the optimal throughput region is the union of the throuutghput regions of all possible scheduling policies, we have $\nu_{FD} \geq B_{FD}/N$. The sum-rate under $\mathbf{f}$ is calculated by:

$$\sum_{i=1}^{N} R_i^{\mathbf{f}} = \sum_{j=1}^{I} \sum_{k<j} m_j m_k + \left(K - \sum_{j=1}^{I} m_j\right) \sum_{j=1}^{I} m_j. \tag{11}$$

where $m_j$ is the $j^{th}$ element in the user-selection vector $\mathbf{m}$.

The first term of the R. H. S. of (11) calculates the total rate from the first mini-slot to $\sum_{j=1}^{I} m_j^{th}$ mini-slot, we only need to count the number of user pairs $(u_i, u_j)$ such that $g(u_i) \neq g(u_j)$ and $u_i$ is scheduled before $u_j$. After $\sum_{j=1}^{I} m_j^{th}$ mini-slot, all scheduled user will have $K - \sum_{j=1}^{I} m_j$ additional rate. The total rate from the remaining mini-slot is just $\left(K - \sum_{j=1}^{I} m_j\right) \sum_{j=1}^{I} m_j$. To get the upper bound of the sum-rate, we need to solve the following maximization problem.

$$\begin{aligned} \underset{m}{\text{maximize}} \quad & \sum_{j=1}^{I} \sum_{k<j} m_j m_k + \left(K - \sum_{j=1}^{I} m_j\right) \sum_{j=1}^{I} m_j \\ \text{subject to} \quad & m_i \leq N/I, m_i \in \mathbb{N}, \\ & \text{for all } i = 1, 2, \cdots, I. \end{aligned}$$

If $N/I \geq \frac{K}{I+1}$ for all $i = 1, 2, \cdots, I$, then the maximum is achieved by taking $m_i = \frac{K}{I+1}$ for all $i = 1, 2, \cdots, I$. In this case, $B_{FD} = \frac{IK^2}{2(I+1)}$. Otherwise, the maximum is achieved by taking $m_i = N/I$ for all $i$.

$B_{FD} = \frac{N(2IK-N-IN)}{2I}$. In a word,

$$\nu_{FD} = \begin{cases} \frac{IK^2}{2N(I+1)}, & N \geq \frac{IK}{I+1}, \\ \frac{2IK-N-IN}{2I}, & \text{otherwise.} \end{cases} \quad (12)$$

Define Full-duplex gain $G_{FD} = \frac{\nu_{FD}}{\nu_{HD}}$, $\alpha = K/N$. We have:

$$G_{FD} = \begin{cases} \frac{2I}{I+1}, & \alpha < \frac{I+1}{I}, \\ \frac{2(2I\alpha-1-I)}{I\alpha^2}, & \frac{I+1}{I} \leq \alpha < 2, \\ 1 + \frac{I-1}{2I(\alpha-1)}, & \alpha \geq 2. \end{cases} \quad (13)$$

Fix group number $I = 10$, Fig. 3 shows the Full-duplex gain $G_{FD}$ for different $\alpha$. As we can see in the figure, if $\alpha$ is smaller than 1.1,

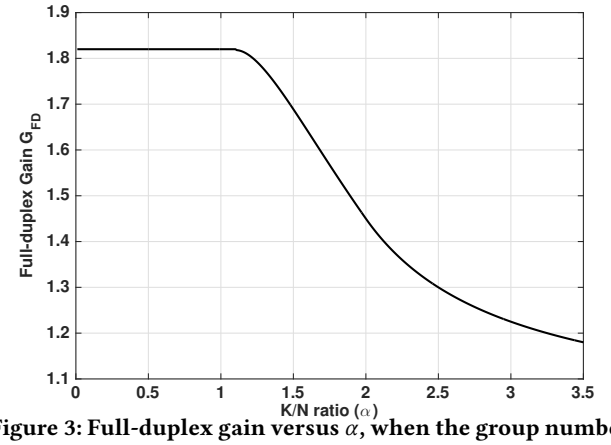



**Figure 3: Full-duplex gain versus $\alpha$, when the group number $I = 10$.**

Full-duplex gain $G_{FD}$ remains larger than 1.8. In this regime, the number of users $N$ is larger than (or comparable to) $K$, which means the learning phase takes as long as nearly $K/2$ mini-slots. Note that the Full-duplex gain comes from concurrent channel probing and data transmission, the longer learning phase takes, the larger $G_{FD}$ will be observed. On the other hand, when $\alpha$ becomes larger, $G_{FD}$ decreases from 1.82 to 1.18. This is because the learning phase is negligible compared to $K$, thus we don't have much gain compared to the traditional schemes. In general, when $I$ becomes larger, the upper bound of the $G_{FD}$ becomes closer to 2, which matches the expected potential of the Full-duplex gain.

Fix $\alpha$ to be 1.0, 1.5 and 3, Fig. 4 shows how does the Full-duplex gain $G_{FD}$ change with different group number $I$. From Fig. 4, we can observe that the Full-duplex gain $G_{FD}$ keeps increasing as $I$ becomes larger. The scheduler has more flexibility when given more groups, thus a larger Full-duplex gain should be expected. Moreover, in many user regime (green and blue curve), $G_{FD}$ has improved by 40% and 30% when $I$ increases from 2 to 15. However, $G_{FD}$ does not improve much in small user regime (red curve). The learning phase only takes a small fraction of time, thus $G_{FD}$ is always a little larger than 1.1, regardless of what value $I$ takes.

# 7 NUMERICAL RESULTS

In this section, we use simulations to evaluate our proposed greedy policy and compare its performance with traditional Half-duplex and Full-duplex MaxWeght Scheduling (MWS) schemes.

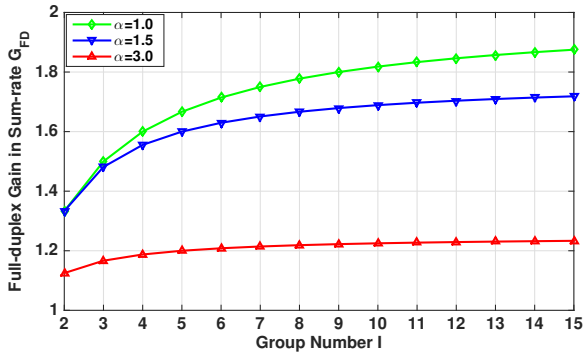



**Figure 4: Full-duplex gain versus group number $I$, when the $K/N$ ratio ($\alpha$) is fixed.**

## 7.1 Simulation Settings

We consider the downlink system of a single-cell Full-duplex MIMO system. There are $N$ users in this system and each user is equipped with only one antenna. The BS is assumed to have sufficiently large number of antennas. Suppose all users are divided into $I$ user groups such that users from different group does not interfere with each other. Unlike the assumption we make in Section 6, each user group now could have different group size. In addition, we assume that each time-slot has 15 mini-slots, i.e., $K = 15$. We consider i.i.d. arrival, i.e.,

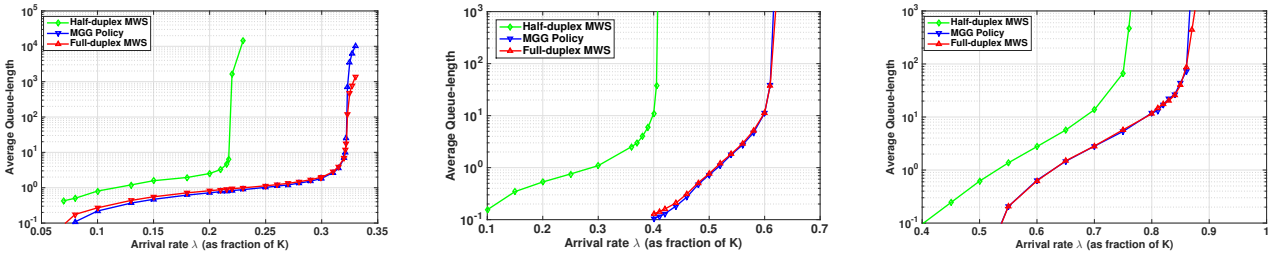
$$A_u[t] = \begin{cases} K, & \text{with probability } \lambda, \\ 0, & \text{otherwise.} \end{cases}$$

where $\lambda$ is the scaled arrival rate of queue $u$, $u \in \mathcal{N}$.

## 7.2 Performance of Greedy Policy under Different Regimes

Fix group number $I = 4$, we then evaluate the performance of the proposed greedy policy in three regimes which represent three conditions of (13). Define regime 1 as the many-user regime such that $\alpha \leq 1.25$. In regime 1, we take $N_1 = 8, N_2 = 5, N_3 = 6, N_4 = 1$, with sum $N = 20$ and $\alpha = 0.75$. Regime 2 denotes the moderate regime, where $N$ is comparable with $K$ such that $1.25 \leq \alpha \leq 2$. In regime 2, $N_1 = 3, N_2 = 2, N_3 = 2, N_4 = 3$, with sum $N = 10$ and $\alpha = 1.5$. Regime 3 represents the small-user regime such that $\alpha \geq 2$. In regime 3, we take $N_1 = 1, N_2 = 1, N_3 = 1, N_4 = 1$, with sum $N = 4$ and $\alpha = 3.75$. For all these three scenarios, we plot the average queue-length under different arrival rate $\lambda$ in Fig. 5.

In all three regimes, the performance of the MGG policy is very close to the Full-duplex MaxWeight policy. Thus, the throughput performance of the MGG policy is also very close to optimal. The Full-duplex gain is larger if $\alpha$ is small, meaning $K$ is smaller compared to $N$. In this case, the control overhead of sending probing signals becomes the system bottleneck. Introducing Full-duplex reduces the control overhead and thus the throughput is improved substantially. As $\alpha$ becomes larger, the control overhead no longer limits the throughput, since it only takes a small fraction of time to send probing signals. As a result, Full-duplex gain decreases from 1.5 to 1.13 from as $\alpha$ increases from 0.75 to 3.75.




(a) Regime 1 (b) Regime 2 (c) Regime 3

**Figure 5: Average queue-length under different arrival rate.**

### 7.3 Performance of Greedy Policy under Random Group Assignments

Given $N$ users, the way of assigning users to different groups affects the Full-duplex gain. In this section, we would like to evaluate throughput performance under random group assignments. Fix group number $I = 4$, number of users $N = 10$ and $K = 15$. Assume that each user has equal probability to be assigned to each group, the following figure shows the empirical CDF of the Full-duplex gain for 10000 samples of random group assignments.

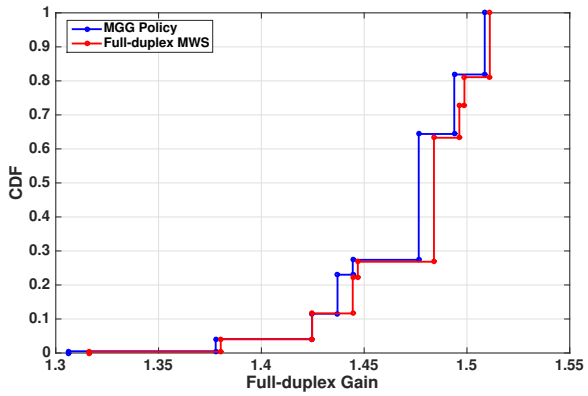



**Figure 6: The empirical CDF for Full-duplex gain compared to Half-duplex throughput optimal policy**

From Fig. 6, we can observe that the Full-duplex gain of the MGG policy and MaxWeight policy have similar distributions. Although in theory there may exist scenarios in which the MGG policy is sub-optimal, in typical scenarios it achieves near-optimal throughput performance. The median Full-duplex gain under the MaxWeight scheduling and the MGG policy is around 1.48. Although the lowest Full-duplex gain is around 1.3, in typical scenarios (90% of all samples), the Full-duplex gain is larger than 1.44 (44% improvement).

## 8 CONCLUSION

In this paper, we develop a throughput optimal scheduling policy for concurrent channel probing and data transmission scheme. To further reduce the complexity when there are a large number of groups, we propose a greedy policy with complexity $O(N \log N)$ that not only achieves at least 2/3 of the optimal throughput region but also outperforms any feasible Half-duplex solutions. Furthermore, we derive the Full-duplex gain for different system parameters. Finally, we use numerical simulations to validate our theoretical results.

## ACKNOWLEDGMENTS

This work has been supported in part by NSF grants CNS-1254032, CNS-1302620, CNS-1314538, CNS-1518829 and CNS-1547306, Office of Naval Research grant N00014-17-1-2417, and from the Army Research Office grant W911NF-14-1-0368.

## REFERENCES

[1] 2016. White paper: Cisco VNI Forecast and Methodology, 2015-2020. http://www.cisco.com. (2016).
[2] . 2017. Concurrent Channel Probing and Data Transmission in Full-duplex MIMO Systems. https://arxiv.org/abs/1705.08000. (May. 2017).
[3] Matthew Andrews, Krishnan Kumaran, Kavita Ramanan, Alexander Stolyar, Rajiv Vijayakumar, and Phil Whiting. 2004. Scheduling in a queuing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences* 18, 02 (2004), 191–217.
[4] Dinesh Bharadia and Sachin Katti. 2014. Full duplex MIMO radios. In *USENIX NSDI*. 359–372.
[5] Dinesh Bharadia, Emily McMilin, and Sachin Katti. 2013. Full duplex radios. *ACM SIGCOMM Computer Communication Review* 43, 4 (2013), 375–386.
[6] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. 2010. Achieving single channel, full duplex wireless communication. In *ACM MOBICOM*. ACM, 1–12.
[7] Xu Du, John Tadrous, Chris Dick, and Ashutosh Sabharwal. 2014. MIMO broadcast channel with continuous feedback using full-duplex radios. In *Asilomar Conference on Signals, Systems and Computers*. IEEE, 1701–1705.
[8] Xu Du, John Tadrous, Chris Dick, and Ashutosh Sabharwal. 2015. MU-MIMO beamforming with full-duplex open-loop training. In *International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 301–305.
[9] Melissa Duarte, Chris Dick, and Ashutosh Sabharwal. 2012. Experiment-driven characterization of full-duplex wireless systems. *IEEE Transactions on Wireless Communications* 11, 12 (2012), 4296–4307.
[10] Atilla Eryilmaz, Rayadurgam Srikant, and James R Perkins. 2005. Stable scheduling policies for fading wireless channels. *IEEE/ACM Transactions on Networking* 13, 2 (2005), 411–424.
[11] Evan Everett and Ashutosh Sabharwal. 2016. Spatial degrees-of-freedom in large-array full-duplex: the impact of backscattering. *EURASIP Journal on Wireless Communications and Networking* 2016, 1 (2016), 286.
[12] Bo Ji, Gagan R Gupta, Manu Sharma, Xiaojun Lin, and Ness B Shroff. 2015. Achieving optimal throughput and near-optimal asymptotic delay performance in multichannel wireless networks with low complexity: a practical greedy scheduling policy. *IEEE/ACM Transactions on Networking* 23, 3 (2015), 880–893.
[13] Xiaojun Lin, Ness B Shroff, and Rayadurgam Srikant. 2006. A tutorial on cross-layer optimization in wireless networks. *IEEE Journal on Selected areas in Communications* 24, 8 (2006), 1452–1463.
[14] Jia Liu, Atilla Eryilmaz, Ness B Shroff, and Elizabeth S Bentley. 2016. Understanding the impact of limited channel state information on massive MIMO network performances. In *ACM MOBIHOC*. 251–260.
[15] Jelena Marašević and Gil Zussman. 2016. On the Capacity Regions of Single-Channel and Multi-Channel Full-Duplex Links. *arXiv preprint arXiv:1605.07559* (2016).
[16] Michael J Neely. 2013. Delay-based network utility maximization. *IEEE/ACM Transactions on Networking* 21, 1 (2013), 41–54.

[17] Achaleshwar Sahai, Suhas Diggavi, and Ashutosh Sabharwal. 2013. On uplink/downlink full-duplex networks. In *Asilomar Conference on Signals, Systems and Computers*. IEEE, 14–18.
[18] Quentin H Spencer, Christian B Peel, A Lee Swindlehurst, and Martin Haardt. 2004. An introduction to the multi-user MIMO downlink. *IEEE Communications Magazine* 42, 10 (2004), 60–67.
[19] Leandros Tassiulas and Anthony Ephremides. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Automat. Control* 37, 12 (1992), 1936–1948.
[20] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication*. Cambridge university press.
[21] Xiufeng Xie and Xinyu Zhang. 2014. Does full-duplex double the capacity of wireless networks?. In *IEEE INFOCOM*. 253–261.
[22] Yang Yang, Bo Chen, Kannan Srinivasan, and Ness B Shroff. 2014. Characterizing the achievable throughput in wireless networks with two active RF chains. In *IEEE INFOCOM*. 262–270.
[23] Yang Yang and Ness B Shroff. 2015. Scheduling in wireless networks with full-duplex cut-through transmission. In *IEEE INFOCOM*. 2164–2172.
[24] Anfu Zhou, Teng Wei, Xinyu Zhang, Min Liu, and Zhongcheng Li. 2015. Signpost: Scalable MU-MIMO signaling with zero CSI feedback. In *ACM MOBIHOC*. ACM, 327–336.

# A PROOF OF LEMMA 5.8

*Definition A.1.* (*Available rate*) Let $S(t)$ denote the "available rate" in $t^{th}$ mini-slot:

$$S(t+1) = S(t) + \begin{cases} t_1(b_j) - t_2(c_j) & \text{if } u_t^G = c_j, \\ -(t_2(a_j) - t_1(a_j)) & \text{if } u_t^G = a_j. \end{cases}$$

where the initial value $S(0) = 0$ and $u_t^G$ is the $t^{th}$ element in the MGG schedule $\mathbf{f}^G$.

Start with the first scheduled user in $\mathbf{f}^G$, if we encounter with a user from $C$, then the "available rate" will be added $t_1(b_j) - t_2(c_j)$ more rates offered by users in $C$. Otherwise, the "available rate" will be deducted by "$\mathcal{A}$ loss rate" $t_2(a_j) - t_1(a_j)$. In general, $S(t+1)$ is the sum of available rate of queue-length no smaller than $Q_{u_t^G}$. The definition of $S(t)$ allows us to decouple the queue-length from its rate, and to evaluate (7), we only need to compare the "available rate" and "$\mathcal{A}$ loss rate". If for any $1 \le t \le K$, $S(t)$ is always non-negative, then the R. H. S. of (7) is also non-negative. Consider each $t$ such that $u_t^G \in \mathcal{A}$, $S(t+1) \ge 0$ means the sum of available rate received by users with queue-length higher than $Q_{u_t^G}$ is larger than the "$\mathcal{A}$ loss rate" on user $u_t^G$. That is to say, for each $a_i$, there will be sufficiently many rate offered by users in $C$ which have longer queue-length than $Q_{a_i}$. It is sufficient to show that the R. H. S. of (7) is non-negative.

On the other hand, we can rewrite the recursion formula of $S(t)$ as:

$$S(t+1) = S(t) + \begin{cases} t_1(b_j) - t_2(c_j) & \text{if } u_t^G = c_j, \\ t_1(a_j) - t_2(a_j) & \text{if } u_t^G = a_j. \end{cases}$$

Start from $t = 1$, for each user $u_t^G$, $S(t)$ increments by the time difference of scheduling the same user or the corresponding user under mapping $h$. Thus, $S(t)$ is actually the difference between the sum of $t$ different timestamps in MaxWeight schedule and the sum of $t$ consecutive timestamps from $1, 2, \cdots$ up to $t$. The later sum is the minimum of the sum of $t$ different timestamps, hence $S(t) \ge 0$ holds for any $1 \le t \le K$.

# B PROOF OF LEMMA 5.9

We use mathematical induction to prove this lemma.

***Base Case:*** If $m = 1$, it is the trivial case, since $\epsilon_i^{K_1} = 0$.

If $m = 2$, we have:

$$\frac{\epsilon_i^{K_2}}{w_i(\mathbf{f}_{K_2}^G)} = \frac{Q_{u_1}}{Q_{u_1}(T_2 + t_2(u_2) - t_2(u_1) - 1) + Q_{u_2}T_2}. \tag{14}$$

We know that $Q_{u_2}T_2 \ge Q_{u_1}$ and $T_2 \ge 1$, $t_2(u_2) - t_2(u_1) \ge 1$. Hence $Q_{u_1}(T_2 + t_2(u_2) - t_2(u_1) - 1) + Q_{u_2}T_2 \ge 2Q_{u_1}$ and $\frac{\epsilon_i^{K_2}}{w_i(\mathbf{f}_{K_2}^G)} \le 1/2$.

***Inductive hypothesis:*** Assume the lemma holds for $m$ users from group $i$, i.e., $\frac{\epsilon_i^{K_m}}{w_i(\mathbf{f}_{K_m}^G)} \le 1/2$.

***Inductive step:*** consider the case where we have $m+1$ users from group $i$ ($Q_{u_1} \ge Q_{u_2} \cdots \ge Q_{u_{m+1}}$). $T_{m+1}$ must satisfy:

$$\begin{cases} Q_{u_{m+1}}T_{m+1} \ge \sum_{j=1}^{m} Q_{u_j}. & (15) \\ Q_{u_{m+1}}(T_{m+1} - 1) < \sum_{j=1}^{m} Q_{u_j}. & (16) \end{cases}$$

User $u_1, u_2, \cdots, u_m$ will determine $T_m$:

$$\begin{cases} Q_{u_m}T_m \ge \sum_{j=1}^{m-1} Q_{u_j}. & (17) \\ Q_{u_m}(T_m - 1) < \sum_{j=1}^{m-1} Q_{u_j}. & (18) \end{cases}$$

We then evaluate $\epsilon_i^{K_{m+1}} / w_i(\mathbf{f}_{K_{m+1}}^G)$:

$$\frac{\epsilon_i^{K_{m+1}}}{w_i\left(\mathbf{f}_{K_{m+1}}^G\right)} = \frac{\epsilon_i^{K_m} + \sum_{j=1}^{m} Q_{u_j}}{w_i\left(\mathbf{f}_{K_m}^G\right) + Q_{u_{m+1}}T_{m+1} + \sum_{j=1}^{m} Q_{u_j}(K_{m+1} - K_m - 1)}. \tag{19}$$

Given the inductive hypothesis, it suffices to show

$$2\sum_{j=1}^{m} Q_{u_j} \le Q_{u_{m+1}}T_{m+1} + \sum_{j=1}^{m} Q_{u_j}(K_{m+1} - K_m - 1). \tag{20}$$

From (15), we already know $\sum_{j=1}^{m} Q_{u_j} \le Q_{u_{m+1}}T_{m+1}$. We only need to show $\sum_{j=1}^{m} Q_{u_j} \le \sum_{j=1}^{m} Q_{u_j}(K_{m+1} - K_m - 1)$, or equivalently, $K_{m+1} - K_m - 1 \ge 1$. By definition, $K_{m+1} = t_2(u_{m+1}) + T_{m+1} \ge t_2(u_m) + 1 + T_{m+1}$, $K_m = t_2(u_m) + T_m$. The only thing left is to show $T_{m+1} - T_m \ge 1$ ($T_{m+1} > T_m$). Suppose $T_m \ge T_{m+1}$, from (18), we know:

$$Q_{u_m}T_m < \sum_{j=1}^{m-1} Q_{u_j} + Q_{u_m} = \sum_{j=1}^{m} Q_{u_j}. \tag{21}$$

Then,

$$Q_{u_{m+1}}T_{m+1} \le Q_{u_m}T_m < \sum_{j=1}^{m} Q_{u_j}. \tag{22}$$

(22) contradicts (15), therefore, $T_{m+1} > T_m$, $T_{m+1} - T_m \ge 1$, Lemma 5.9 holds.