Abstract—Wireless multicasting suffers from the problem that the transmit rate is usually determined by the receiver with the worst channel conditions. Composite or adaptive beamforming allows using beamforming patterns that trade off antenna gains between receivers, which can be used to overcome this problem. A common solution for wireless multicast with beamforming is to select the pattern that maximizes the minimum rate among all receivers. However, when using opportunistic multicast to transmit a finite number of packets to all receivers – the finite horizon problem – this is no longer optimal. Instead, the optimum beamforming pattern depends on instantaneous channel conditions as well as the number of received packets at each receiver. We formulate the finite horizon multicast beamforming problem as a dynamic programming problem to obtain an optimal solution. We further design a heuristic that has sufficiently low complexity to be implementable in practice. To deal with imperfect feedback, and in particular feedback delay, we extend the algorithm to work with estimated state and channel information. We show through extensive simulations that our algorithms significantly outperform prior solutions.

I. INTRODUCTION

Wireless multicast is an efficient technique to disseminate multimedia data to groups of users. Using the broadcast nature of the wireless medium allows data to be served to multiple users simultaneously, but at the same time this constrains the transmit rate to the rate that can be supported by the receiver with the worst channel conditions.

To address this issue, a range of different mechanisms have been proposed. With opportunistic multicast scheduling (OMS), a base station (BS) exploits multiuser diversity and may opportunistically transmit to a subset of receivers that experience good channel conditions. This trades off multicast gain (achieved by transmitting to as many receivers as possible) versus multiuser diversity gain (achieved by transmitting to a subset of receivers that currently experience good channel conditions) [1]–[3]. Multiuser diversity gain can be exploited if receivers with a bad channel are likely to experience better channels in the future, and vice versa. It is particularly suitable for homogeneous scenarios where receivers’ long-term average rates are similar, but instantaneous rates are highly variable. If, however, there are some receivers that on average have worse channels than the rest, exploiting multiuser diversity and transmitting preferentially to the users with better channels is detrimental to performance. Here, the overall rates are still limited by the worst receivers.

To overcome this problem, transmit beamforming can be used to adjust antenna gains to the different receivers. This allows improving the signal-to-noise ratios (SNR) of receivers with bad instantaneous channels at the expense of worsening those of receivers with better instantaneous channels. There are two main techniques for multi-user beamforming: (i) composite beamforming [4] and (ii) adaptive beamforming [5]. A composite beam is composed of multiple pre-determined single-lobe beam patterns. In contrast, adaptive beamforming calculates antenna weights directly based on the measured channels to the different receivers. While composite beamforming has lower complexity, adaptive beamforming may achieve better performance in multi-path rich environments. Similar to opportunistic multicast, the main challenge when designing multi-user beamforming mechanisms is the tradeoff between high gain beamforming to few receivers versus lower gain beamforming to a larger receiver set [4], [6], [7].

Most of the prior work in this area aims at maximizing the rates of the receivers, which is optimal for the infinite horizon problem (i.e., where an infinite amount of data is to be sent). In contrast, we analyze the more realistic finite horizon problem where the BS sends a block of data of a certain size to all receivers. In case there are multiple blocks to be sent, the BS starts transmitting packets for the next block only after all receivers have received the first data block. This is relevant for most practical scenarios where reliable delivery as well as delay constraints are a concern, as for example for multicast video streaming.

In the infinite horizon problem, it is possible to exploit opportunistic gain very aggressively, since lagging receivers have an infinite amount of time to catch up. In contrast, in the finite horizon case, the decision when to exploit opportunistic gain and when to favor lagging users very much depends on the state of the receivers (i.e., the amount of data received thus far and therefore how close the receivers are to finishing). This substantially changes the problem compared to the infinite horizon problem. In addition, the higher the number of users, the higher the multi-user diversity and therefore the potential for opportunistic gain. Exploiting opportunistic gain aggressively leads to higher average rates in the short term, but at the same time may lead to some users finishing early, thus reducing potential future opportunistic gain.

We first model the problem and obtain the optimum solution via dynamic programming. This allows us to study the impact of receiver state, instantaneous channel conditions, and average channels on the optimum receiver set to transmit to, and hence the optimum beamforming patterns. First, we study these tradeoffs in toy scenarios with two users. These insights allow us to design a low complexity heuristic algorithm that captures the main characteristics of the optimum solution and at the same time can run in real-time in practical wireless scenarios.

We then present a range of simulation results for larger scenarios with homogeneous and heterogeneous receiver sets and Rayleigh fading channels. While the complexity of the dynamic programming solution prevents us from solving those larger problem instances optimally, we see that our proposed heuristic provides significantly better performance than so-
lutions based on broadcasting or greedily maximizing rates. Note that both the broadcast and greedy mechanisms use beamforming and take the instantaneous channel conditions into account. The greedy mechanism is thus optimal for the infinite horizon case (or problem instances with very large block sizes) in homogeneous scenarios as shown in [8]. The broadcast mechanism makes use of beamforming to maximize the minimum rate and hence does not suffer from receivers with bad channel conditions as much as conventional OMS. It corresponds to the solution in [7] that is optimal for scenarios with fixed channels but may be too conservative in case of variable channels. It is also optimal for scenarios with variable channels where the receiver set is very heterogeneous and one receiver has a significantly worse average channel than the other receivers.

In practice, feedback arrives with a certain delay and the feedback frequency has to be set sufficiently low so as not to create excessive feedback overhead. We investigate the impact of imperfect feedback on the performance of the algorithms and extend our heuristic algorithm to make decisions based on partial information and estimated channel and receiver state.

Similar to prior work mentioned above, we assume erasure coding of transmitted data, which is highly beneficial in wireless multicast scenarios and ensures that each packet received by a receiver is useful (with high probability). Fixed rate LDPC [9] or rateless LT [10] or Raptor codes [11] are examples of such erasure codes that work well in practice. Also rateless channel codes [12]–[14] are an interesting alternative that integrate the above rateless coding properties directly into the channel code. In our design, this integration is less important since the multicast beamforming equals instantaneous SNRs at the receivers, while multicast rateless channel coding is particularly beneficial to allow receivers with different SNRs to decode. To make our approach amenable to currently used RF decoders (LTE, 802.11), we do not consider rateless channel coding in this paper.

The paper is structured as follows. A review of state-of-the-art for opportunistic multicast beamforming is given in Section II. In Section III, we model the finite horizon opportunistic multicast beamforming problem and provide an optimum solution based on dynamic programming. We design a low complexity heuristic, FH-OMB, in Section IV, and in Section V we compare its performance to the optimum solution and the greedy and broadcast schemes proposed in prior works. Section VI concludes the paper and provides an outlook on future work.

II. RELATED WORK

Opportunistic Multicasting: Opportunistic multicasting has been well studied for both the infinite horizon problem [1], [15]–[18] as well as the finite horizon problem [2], [3], [19]. Among the first ideas to address the infinite horizon problem for homogeneous scenarios was to split the receivers into two groups according to their instantaneous channels and serve the group with the better channel quality. As the composition of the group changes from slot to slot, all users have equal chances to be served [15], [20], [21]. This work was extended in [1] by optimizing the selection ratio, i.e., the size of the receiver set to transmit to. As a single pre-computed selection ratio is not always optimal, [17] and [2] propose a dynamic user selection mechanism that depends on the instantaneous channel at each transmission. The authors of [2] solve the user selection problem for the finite horizon case using extreme value theory to minimize completion time. However, the user selection is only based on the instantaneous channel but not on the user state (i.e., the amount of data received by users). In wireless systems with packet loss, this is suboptimal since users may have received a different number of packets. The problem is addressed in [3], [19], where it is shown that the optimal solution for the finite horizon problem needs to take receiver state into account. The main challenge of opportunistic multicasting is to cope efficiently with receivers with bad channel conditions. In this context, transmit beamforming can be used to balance the users’ SNRs.

Multicast Beamforming: Multicast beamforming provides a trade off between multicast gain and beamforming gain. Beamforming to receivers with poor channel conditions improves the SNR at these receivers (but at the same time lowers SNR at other receivers). The basic algorithm proposed in [6] first transmits omnidirectionally to the receivers that have a high SNR and then beamforms sequentially to the remaining weak receivers. Better performance can be achieved by selecting the beamforming vector that maximizes the minimum SNR among all multicast receivers [22], [23]. In [4], receivers are partitioned into groups that are scheduled sequentially, which may outperform mechanisms that always beamform to all receivers. The paper proposes two multicast beamforming mechanisms, one that splits power equally among all beams and one that allows for asymmetric power allocation. Both mechanisms use composite beamforming, where a multi-lobe beam pattern that serves multiple receivers is composed of multiple single-lobe beam patterns. In [7], the authors improve upon this work and provide an optimal solution for the equal power split and two different heuristics for the (NP-hard) asymmetric power allocation mechanism. Both [4] and [7] consider the finite horizon problem but do not take channel variations and opportunistic scheduling into account. The same problem is addressed in [5] using adaptive beamforming. While providing better antenna gains, determining optimum beamforming pattern with adaptive beamforming in multipath environments is more complex.

Opportunistic Multicast Beamforming: There is very little existing work that jointly takes opportunistic multicast scheduling and multicast beamforming into account. A theoretical analysis of the optimum user selection ratio for opportunistic multicast beamforming using extreme value theory is provided in [8]. Once the user group is determined, the optimal beamforming pattern is the one that maximizes the minimum SNR among the users that are served. The algorithm is designed for independent and identically distributed users (i.e., homogeneous scenarios) for the infinite horizon multicast problem. A similar work in [24] also focuses on the same infinite horizon problem and proposes an instantaneous throughput maximization algorithm. We show that these approaches are not suitable for the finite horizon multicast
problem, especially for heterogeneous user distributions. Other relevant works presented in [25]–[31] improve throughput by exploiting the MIMO capability. However, getting timely channel state information (CSI) for all users is hard, particularly for mobile users. Our paper differs from prior work in that it addresses finite horizon opportunistic multicast beamforming in homogeneous and heterogeneous scenarios and explicitly takes into account receiver state (i.e., the amount of data already received).

III. SYSTEM MODEL

We consider a wireless network with a single BS (or access point) and a set \( T \) of multicast receivers, with \(|T| = N\). We assume the channels between the BS and the receivers are independent discrete memoryless channels.\(^1\) Let \( G \) denote the set of all possible vector channels from the BS to the receivers. The probability that at a given time the channel vector \( C \) has channel gains \( g \in G \) is given by \( P(C = g) \). Let \( C_i, g_i, \) and \( G_i \) denote the corresponding channel instance, gain, and set of possible channels for receiver \( i \). As is common for opportunistic scheduling, we assume that the BS has perfect knowledge of the current channel instances, but for any future channel instances only the channel distribution is known.

The BS uses composite beamforming. The antenna array has \( K \) antenna patterns that are optimized to produce one strong single-lobe beam that covers a sector of approximately \( 360^\circ / K \) and that together cover the whole azimuth of \( 360^\circ \). A composite beam is a multi-lobe beam pattern composed of several single-lobe beams that are transmitted simultaneously [4]. Each single-lobe beam \( k \) has a certain beam weight \( \alpha_k \). This weight corresponds to the fraction of the total transmit power allocated to that beam, and thus determines the SNR at the receivers covered by the beam. To ensure that the total radiated power remains unchanged, we have the constraint \( \sum_k \alpha_k = 1 \). Let \( k_i^* \) be the strongest single-lobe beam that covers receiver \( i \) and let \( \gamma_{i}^{\text{SLB}} \) denote the SNR at that receiver when using that single-lobe beam when the channel gain is \( g_i \). Then the SNR of that receiver for a multi-lobe beam pattern is

\[
\gamma_{i}^{\text{SLB}} = \alpha_k \gamma_{i}^{\text{SLB}}.
\]

We consider a time-slotted model. In each time slot the BS transmits data to the receivers using a certain modulation and coding scheme (MCS) and beamforming pattern. For MCS \( m \in M \), the number of bits transmitted in a slot is \( R_m \) and the corresponding packet reception probability for an SNR of \( \gamma \) is \( p_{m}(\gamma) \). Note that we assume that receiver \( i \) will only be served when a multi-lobe beam is used with \( \alpha_{k_i^*} \neq 0 \).

A. Problem Formulation

The BS has a block of data of size \( B \) (in bits) to transmit to all receivers. An erasure code is applied to the data before transmission, so that each data packet is useful for each receiver that receives it, as long as that receiver has obtained less than \( B \) bits so far.

\(^1\)Note that our heuristic works for continuous channels and we provide simulation results for Rayleigh fading channels in Section V.

The optimization problem is thus for the BS to select at each time slot the multi-lobe beam pattern with corresponding weights and the MCS that minimizes the expected completion time. Optimal choice of beam pattern and MCS depend on the current instantaneous channel, the probability distributions of the channels, and the amount of data received by the receivers so far.

When beamforming to a subset of receivers \( T' \subseteq T \), the highest expected rate to those receivers is obtained by selecting beam weights \( \alpha_k^* \) that maximize the minimum SNR at the receivers. Let \( T'_k = \{i \in T' : k_i^* = k\} \subseteq T' \) be the subset of receivers served by beam \( k \). The minimum SNR of receivers in \( T'_k \) for a single-lobe beam pattern and a given channel \( g \) is

\[
\gamma_{i}^{\text{SLB}}(T'_k) = \min_{i \in T'_k} \gamma_{i}^{\text{SLB}}(g)
\]

and, as shown in [7], the optimum weights for the multi-lobe beam pattern are thus given by

\[
\alpha_k^* = \begin{cases} \frac{1}{\sum_{j=1}^{K} \gamma_{j}^{\text{SLB}}(T'_k)}^{-1} & \text{if } T'_k \neq \emptyset \\ 0 & \text{otherwise} \end{cases}
\]

This results in the same minimum SNRs for all lobes of the multi-lobe beam. Hence, all receivers in \( T' \) are served with the MCS that provides the highest expected rate

\[
m^* = \arg \max_{m} R_m p_{m}(\alpha_k^* \gamma_{i}^{\text{SLB}}(T'_k))
\]

Thus, rather than optimizing over all possible beam weights, it is sufficient to optimize over all possible subsets of receivers.

Note that the algorithm in [7] always serves all receivers associated with a given beam, while this is no longer optimal for opportunistic multicast. Consider a scenario where all receivers are located in the same beam. This is the conventional OMS scenario for which it is well known that broadcasting to all users is not always optimal [1].

B. Dynamic Programming Solution

With this we can formulate the problem as a stochastic shortest path problem and solve it through dynamic programming [32]. The state is given by the amount of data received by the receivers so far \( s = [s_1 \ldots s_N], 0 \leq s_i \leq B \) and we denote the state space by \( S \).\(^2\) As all time slots have the same duration, the cost per slot is 1.

When multicasting to a subset \( T' \) of receivers with an instantaneous channel of \( g \), the transition probability from state \( s \) to state \( s' \) is

\[
\rho_g^{T'}(s,s') = \sum_{K} \min_{e \in E} \left( \prod_{i=1}^{N} p_{e_i}(\gamma_{i}^{s_i}) (1 - p_{e_i}(\gamma_{i}^{s_i}))^{1-e_i} \right),
\]

where the vector minimization above is element-wise. \( E = \{e \in \{0,1\}^N\} \) is the set of binary vectors of size \( N \) and \( e_i \) is the \( i \)th element of \( e \), indicating whether receiver \( i \) received the

\(^2\)Given that there is a discrete set of rates \( R_m \), many states cannot be reached and we remove these states from the state space to speed up the computation.
the next section we design a lower complexity heuristic. A policy \( \mu_* : G \mapsto \bigcup_{T' \subseteq T} T' \) specifies the best subset of receivers to transmit to for any instantaneous channel \( g \) when in state \( s \). Let \( G \) be the set of all possible mappings. Since the probability of terminating after a finite number of steps is positive, we can use Bellman’s equation to find the optimal policy

\[
\mu_* = \arg \min_{\mu \in M} \left( \sum_{g \in G} P(C = g) \sum_{s' \in S} \rho^{\mu(g)}(s, s') D^*(s') \right).
\]

The corresponding optimal expected completion time is

\[
D^*(s) = \min_{\mu \in M} \left( \sum_{g \in G} P(C = g) \sum_{s' \in S} \rho^{\mu(g)}(s, s') D^*(s') \right).
\]

IV. HEURISTIC ALGORITHM

Our Finite-Horizon Opportunistic Multicast Beamforming (FH-OMB) heuristic has two main parts: 1) given the current instantaneous channel, computing the next states the system could move to using the different multi-beam lobes that correspond to multicasting to the different subsets of receivers, and 2) estimating the expected completion times from those new states. The decision taken by the heuristic is then to beamform to the subset of receivers that results in moving to the state with the lowest expected completion time.\(^3\)

A. Instantaneous Beamforming Decision

Let the current state be \( s \) and the current instantaneous channel be \( g \). Assume the estimated completion times \( D(s') \) for all future states are known. When beamforming to \( T' \subseteq T \) we can calculate \( \gamma_{g}(T'_i) \), \( \alpha_{g}^{*} \), and the resulting optimum MCS \( m^{*} \) using Equations (1)–(3). The expected future state \( s'_i(T') \) is given by

\[
s'_i(T') = \min \left( s_i + R_{int} p_{m^{*}}(\gamma_{g}^{*}), B \right) \quad \forall i
\]

and the optimum subset of receivers \( T^* \) to beamform to is thus

\[
T^* = \arg \min_{T' \subseteq T} D(s'(T')).
\]

In contrast to the dynamic programming formulation we compute expected average future state rather than looking at all combinations of possible future states based on packet loss events. Note that this still requires minimization over a number of completion times that is exponential in the number of receivers, which can be done exhaustively for small receivers sets.

For larger receiver sets, we cluster receivers according to their state \( s_i \) and relative quality of the instantaneous channel. The rate receiver \( i \) would obtain with the current channel \( g_i \) for a single-lobe pattern is

\[
R(i) = \max_{m} R_m p_m(\gamma_{g_i}^{SLB}),
\]

and the average rate that is obtained under all possible channels is

\[
\bar{R}(i) = \sum_{g_i \in G_i} P(C_i = g_i) \max_{m} R_m p_m(\gamma_{g_i}^{SLB}).
\]

The relative channel quality is \( R(i)/\bar{R}(i) \). Let \( 0 = \xi_1 < \xi_2 < ... < \xi_U = B \) be a set of state thresholds and \( 0 = \theta_1 < \theta_2 < ... < \theta_V = \infty \) be a set of relative channel quality thresholds. We then group all receivers with \( T_{uv} = \{ i \in T : \xi_u \leq s_i < \xi_{u+1}, \theta_v \leq R(i)/\bar{R}(i) < \theta_{v+1} \} \) where the total number of groups is \( U \). In Equation (8), we now optimize over subsets \( T' \subseteq T \) that include whole receiver groups (i.e., if one of the receivers in a group is included, the whole group must be included). We set the thresholds so that the receivers are distributed relatively evenly among the groups. In order to further reduce the number of combinations, a group can only be scheduled if all groups that have better relative channel quality and at the same time have lower or equal receiver state are also scheduled. Fig. 1 shows an example of a 3-by-3 grouping of receivers, with the x and y-axis showing receiver state and relative channel quality, respectively. Scheduling a darker color block marked “Scheduled” requires that all blocks with lighter color marked “Required” also have to be scheduled. With this, the maximum number of combinations and thus the complexity of FH-OMB is \( O \left( \frac{(V+U-1)!}{U!(V-1)!} \right) \). The number of beamforming patterns is fixed for fixed values \( U \) and \( V \).\(^4\)

B. Estimating the Expected Completion Time

The main complexity of the dynamic programming solution lies in the calculation of the expected completion time. Hence, this is what the heuristic primarily addresses. As only the instantaneous channel is known at the BS, we base the expected completion time of a future state on the average channel of the receivers. Due to the shape of the rate function\(^5\), simply averaging the channel would overestimate the receive rate. Hence we first calculate the average single-lobe rate of

\(^3\)Note that our proposed algorithm could be extended so that in case the best beam changes (due to received feedback), the algorithm could directly take this into account. This also takes care of user mobility.

\(^4\)From the simulations we find that a reasonably low value for \( U \) and \( V \) (i.e., \( U = V = 4 \)) suffices in practice, leading to a fixed number of subsets to consider for the optimization.

\(^5\)The rate function depends on the receive power, which is logarithmic based on the Shannon-Hartley theorem.
receiver $i$, $\hat{R}(i)$, as given by Equation (16) and then set the 
receiver’s average SNR $\bar{\gamma}_g^{SLB}$ such that 
\[
\max_m \left( R_m p_m(\bar{\gamma}_g^{SLB}) \right) = \hat{R}(i).
\]

For fixed SNRs and a continuous rate function, according to [7] the maximum rate when multicasting to a receiver set is obtained for a multi-lobe beam pattern that encompasses the whole receiver set. Analogous to Equations (1) and (2), for a receiver subset $T'$ we can derive $\bar{\gamma}_g^{SLB}(T'_k)$ as well as $\hat{\alpha}_k$ based on the average SNRs $\bar{\gamma}_g^{SLB}$ calculated above. The corresponding hypothetical average rate is given by
\[
\hat{R}(T') = \max_m R_m p_m(\hat{\alpha}_k^{SLB}(T'_k)).
\]

With this, we can now approximate the expected completion time as follows. For a given state $s$, let $T_1 = \{i \in T : s_i < B\}$ be the set of receivers that still require further packets and let $s_{\max}^{(1)} = \max_{i \in T_1} s_i$ be the state of the receiver(s) closest to completing. When multicasting to this receiver set at rate $\hat{R}(T'_1)$ given by Equation (11), one or more of the receivers would complete after a time $\tau_1 = (B - s_{\max}^{(1)}) / \hat{R}(T'_1)$. Determine the set of remaining receivers $T_2 = \{i \in T : s_i < s_{\max}^{(1)}\}$ and set $s_{\max}^{(2)} = \max_{i \in T_2} s_i$ to calculate $\tau_2$, etc. In general,
\[
\tau_j = (B - s_{\max}^{(j)}) / \hat{R}(T'_j).
\]

In other words, the estimation algorithm proceeds diagonally through the state space until hitting a boundary with $s_i = B$ for one of the dimensions, then proceeds diagonally along that boundary until hitting the next one, and so on, until reaching the final state. The algorithm terminates after at most $N$ steps. The expected completion time is given by
\[
D(s') = \sum_j \tau_j.
\]

**Accounting for opportunistic gain:** When determining $\tau_j$ above, we assume that receivers in $T'_1$ are served first, then receivers in $T'_2$, etc. This ignores that receiver sets will be selected based (also) on their instantaneous channels. As a consequence, $\hat{R}(T')$ is a conservative estimate of the actual rate at which this receiver group is served, since they are more likely to be served when their channel is good. We refine Equation (12) to take into account opportunistic gain as follows. We assume that receivers in groups $T'_1$ and $T'_2$ are served during $\tau_1 + \tau_2$. If the channels of the receivers in $T'_1 \setminus T'_2$ are good, group $T'_1$ will be served, otherwise group $T'_2$ will be served. Hence, receivers in $T'_1$ see better average channels (since some of the beam weight $\alpha$ that was required for receivers in $T'_1 \setminus T'_2$ can now be used for other beams) whereas there is no change for receivers in $T'_2$. We remove the worst fraction $\tau_2 / (\tau_1 + \tau_2)$ of channel combinations of the receivers in $T'_1 \setminus T'_2$ and update their average channels accordingly. We then recompute Equations (11) and (12) and obtain a new $\tau'_1$. Similarly, the calculation of $\tau'_2$ is based on receiver groups $T''_2$ and $T''_3$, and so on. The completion time is then calculated as $D(s') = \sum_j \tau'_j$.

**Example and discussion:**

To provide an intuition for the completion time estimation, we discuss an example for a two-receiver case in Fig. 2. In a two-user scenario, there are only three possible beamforming patterns serving receiver sets $\{1\}$, $\{2\}$, or both $\{1,2\}$. For $\{1\}$ and $\{2\}$, single-lobe beam patterns with maximum array gain to the respective receiver are used, whereas for $\{1,2\}$ the multi-lobe beam that equalizes the SNRs of the receivers is chosen. In the latter case, both receivers are served at the same rate and have the same packet loss probability. For each of the average future states $s'((1))$, $s'((2))$, and $s'((1,2))$ we compute the expected completion time. Consider, for example, $s'((2))$. Since both users have not yet finished, we calculate the number of time slots $\tau_2$ required for the first receiver to have $B$ bits. In the example this is receiver 2. We then compute $\tau_2$ required for the second user to complete, based on the single-lobe pattern to that user only. Using $\tau_2$, we can recompute the first segment to obtain $\tau'_1$ that partially accounts for opportunistic gain. $\tau'_1 = \tau_1$ since there is no opportunistic gain for a single receiver. The actual path that is taken through the state space (shown with a dotted line) depends on the instantaneous channel conditions at future states and is generally shorter than the sum of the estimated path.

![Fig. 2. Completion time estimation with the FH-OMB heuristic.](https://example.com/fig2.png)
of better receivers and vice versa. This impacts the system’s performance and the problem escalates when the erroneous beam pattern causes a wrong selection of the receiver subset. In what follows, we describe an algorithm to estimate the channel in the absence of instantaneous channel information.

At a time slot $t$, assume that the outdated channel gain of receiver $i$ which was reported $\lambda$ slots ago (delayed by $\lambda$ ms) is $g_i[t-\lambda] \in G_i$. The probability that a channel $C_i$ has a gain $g_i$ given the outdated channel is expressed as follows:

$$P(g_i) = P(C_i = g_i, g_i \in G_i | g_i[t-\lambda]).$$

(14)

Obtaining $P(g_i)$ for all $g_i \in G_i$ gives the probability distribution of the current channel when the feedback frame is delayed by $\lambda$. The resulting estimated channel gain is

$$\hat{g}_i = \sum_{g_i \in G_i} P(g_i) g_i \quad \forall \lambda = [0, \lambda_{\text{max}}].$$

(15)

The corresponding estimated channel instance, channel gain vector and SNR when using a single lobe beam are $\hat{C}_i$, $\hat{g}_i$, and $\gamma_{\hat{g}_i}^{\text{SLB}}$, respectively. Note that after feedback is received, it ages in subsequent slots, until new feedback becomes available. It is therefore important to determine the channel distributions for all possible delays $\lambda$. Fig. 3 shows that a larger $\lambda$ contributes to a wider distribution of the expected channel because a longer delay causes higher channel uncertainty.

Once the $\gamma_{\hat{g}_i}^{\text{SLB}}$ for all receivers are obtained, we compute the optimum weights for the multi-lobe beam pattern in Equation (2) by replacing $\gamma_{\hat{g}_i}^{\text{SLB}}$ in Equation (1) with $\gamma_{\hat{g}_i}^{\text{SLB}}$.

3) Completion time: Section IV-B explains the computation of expected completion time taking into account opportunistic gain when perfect feedback is available. If, however, channel information is outdated, exploiting opportunistic gain becomes more difficult (i.e., the higher the channel uncertainty, the more likely it is that the perceived ‘opportunity’ in fact no longer exists and exploiting it would be detrimental to performance). We take this effect into account when calculating estimated completion time.

As mentioned earlier, obtaining the beam pattern to compute the expected completion time is different than that to obtain the instantaneous beam pattern. Since the distribution of the future channels is determined by the estimated channels (for different $\lambda$), the future beam pattern (which determines the progress of the receivers) is also influenced by the distribution of the received rate given by the distribution of the estimated channels. Consequently, the computation of the average future rate and the beam patterns associated with it is more complex
than that in Equation (11), where instantaneous feedback is available.

Computing the expected completion time requires the average rate of the receivers. To obtain the average rate, we first estimate the average channel of the receivers and then compute the beam weight that gives the average rate. However, as explained in Section IV-B, due to the shape of the rate function, the average estimated channel should be derived from the average rate. This is because simply averaging the estimated channels would result in an overestimation of the average channel thus gives an inaccurate average rate.

On that account, we first compute the average estimated rate based on the distribution of the estimated channel $\hat{g}_i$ from Equation (15) and its corresponding MCS which maximizes the rates of each estimated channel as follows:

$$\bar{R}(i) = \frac{1}{X_{\max}} \sum_{\lambda=0}^{X_{\max}} \left( \sum_{\hat{g}_i \in \hat{g}_i} P(C_i = \hat{g}_i) \max_m \left( R_m p_m(\gamma_{i,\hat{g}_i}^{SLB}) \right) \right)$$

To take delayed feedback into account, Equation (16) includes equally distributed $\lambda$ since the BS experiences an increasing delay of up to $\lambda = \lambda_{\max}$ before it receives the feedback where $\lambda = 0$ and this is repeated until the receiver has received all the intended data.

The corresponding average estimated channel $\gamma_{i,\hat{g}_i}^{SLB}$ based on $\bar{R}(i)$ that is used to determine the beam weight is then derived as follows:

$$\max_m \left( R_m p_m(\gamma_{i,\hat{g}_i}^{SLB}) \right) = \bar{R}(i).$$

The average estimated rate for a receiver subset $T'$ is obtained by replacing $\gamma_{g}^{SLB}$ in Equation (11) with $\gamma_{i,\hat{g}_i}^{SLB}$ from Equation (17). Lastly, the expected completion time is computed based on the average estimated rate in Equation (17) instead of $\bar{R}(T')$ in Equation (11).

V. SIMULATION RESULTS

In this section, we present simulation results to analyze the performance of the algorithms. We first investigate a simple scenario with two receivers and a two-state channel to compare the optimal dynamic programming solution (Dyn-Prog) and the finite horizon opportunistic multicast beamforming heuristic (FH-OMB) and gain insights into the optimum strategy and fundamental tradeoffs. We then investigate more realistic scenarios with multi-path Rayleigh fading channels, larger number of receivers, and larger block sizes. For these, we do not provide dynamic programming results as the run time is prohibitive due to the algorithm’s complexity. The multi-path Rayleigh fading channel corresponds to the ITU Pedestrian B path loss model in [33]. For all the scenarios, we use a subset of 13 MCSs given in the LTE specification for the 20MHz LTE downlink model (with modulation schemes QPSK, 16-QAM, and 64-QAM, and code rates from 0.1885 to 0.9258). The corresponding transmit rates range from 5Mbps to 95Mbps. A time slot has a duration of 1ms. The main performance metric is completion time, i.e., the number of time slots needed for all receivers to receive $B$ kbits. We compare the performance of Dyn-Prog and the FH-OMB heuristic with three alternative mechanisms:

1) Broadcast Algorithm: Broadcast uses a multi-lobe beam pattern that covers all receivers $i$ with $s_i < B$, maximizes the minimum SNR across all lobes, and serves the receivers with the optimum MCS $m^*$ for that SNR as given in Equations (1)–(3). This scheme is presented in [7] (i.e., using the ASP model developed therein) and it is shown to be optimal for constant channels with fixed SNR.

2) Greedy Algorithm: For Greedy, we sort the receivers with $s_i < B$ according to their instantaneous channel quality, given by the single-lobe SNR $\gamma_{i,\hat{g}_i}^{SLB}$. Let $T_1$ be the receiver set that includes the receiver with the best channel (that hasn’t finished yet), $T_2$ be the set of the two receivers with the two best channels, etc. The algorithm then determines the receiver set to beamform to as $T^* = \arg \max \sum_{i \in T_j} R_m p_m(\gamma_{i,\hat{g}_i}^{SLB})$. The optimum receiver set is the one with the highest overall sum rate for all receivers that have not yet finished. This algorithm corresponds to the one proposed in [8] and works well for homogeneous receiver sets.

3) MEBC-ASP Algorithm in [7]: Obtaining the optimal asymmetric power split (ASP) between beams is an NP-hard problem. To address this issue, [7] proposes a simple most efficient bin created (MEBC) algorithm, which first maps the beam multicast problem to a generalized cost variable size bin-packing (GCVS-BP) problem and then solves it using a generalized version of the bin-packing algorithm. The bin packing algorithm partitions the beams into groups such that the aggregated transmission delay is minimized. For common rate functions, and in particular the ones of LTE we use in this paper, only has one most effective bin size, i.e., the lowest MCSs (corresponding to a very large bin size). Thus, the algorithm will serve all receivers with broadcast. The only case where multiple groups are used is when even the lowest MCS is not enough to serve all users at the same time, and it is thus necessary to have more than one group to reduce the number of simultaneous beams (thus the power split). When the algorithm partitions receivers into multiple groups, the aggregated transmission delay is obtained by summing the completion time of each group that is served sequentially by the BS. As the beam grouping is pre-computed using the average channel instead of the instantaneous channel of the receivers, MEBC does not opportunistically exploit channel variations and may suffer from a high delay.

A. Simple Scenario

In this section, we present the results for a simple scenario with $N = 2$ receivers and block size $B = 1000$ kbits. Each receiver $i$ has two possible instantaneous channels ($g_i = \{H_i, L_i\}$ where $H$ and $L$ represents the channel with a higher and lower SNR, respectively), such that $G = \{H_1 H_2, H_1 L_2, L_1 H_2, L_1 L_2\}$ with $P(C_i = H_i) = P(C_i = L_i) = 0.5 \forall i$. We analyze a homogeneous scenario and a heterogeneous scenario.

1) Homogeneous Scenario: In this scenario receivers have the same set of channels ($H = H_1 = H_2, L = L_1 = L_2$). We investigate the impact of channel variability, $\sigma = \gamma_{SLB}^{H} - \gamma_{SLB}^{L}$, on $\gamma_{SLB}^{H}$ and $\gamma_{SLB}^{L}$.
i.e., the difference between the high gain channel and the low gain one. (For example, the left most point of Fig. 4 has $\gamma^{\text{SLB}} = 10\text{dB}$, $\gamma^{\text{L}} = 9\text{dB}$, $\sigma = 1\text{dB}$ and the right most point has $\gamma^{\text{SLB}} = 18\text{dB}$, $\gamma^{\text{L}} = -4.7\text{dB}$, $\sigma = 22.7\text{dB}$). $\gamma^{\text{SLB}}$ and $\gamma^{\text{L}}$ values are chosen such that with single-lobe beamforming the receivers would achieve the same average rate and hence we can compare relative rate changes as the variability increases.

In Fig. 4, both \textit{Greedy} and the \textit{FH-OMB} heuristic perform almost as good as the optimal \textit{Dyn-Prog}. As both receivers have the same channel distribution, differences in receiver state are likely to cancel out over time and maximizing the instantaneous sum rate as \textit{Greedy} does is a good strategy. Only when one receiver is close to finishing and the other receiver is lagging further behind may it be beneficial to favor the lagging receiver instead. Note that the graph also shows 95\% confidence intervals but due to the large number of simulation runs they are tiny.

For small channel variability ($\sigma < 3\text{dB}$), the maximum sum rate is achieved by serving both receivers for any of the channel combinations, hence \textit{Broadcast} and \textit{Greedy} have the same performance. Once the channel variability is beyond this point, beamforming only to the receiver with a good channel when the other receiver has a bad channel ($H_1 L_2$, $L_1 H_2$) provides higher throughput than beamforming to both receivers. Hence, \textit{Broadcast} is unnecessarily conservative by always serving both receivers and its completion time increases substantially as the channels become more variable. The largest most-effective bin size used in our simulation setting is 2.0045 (i.e., the lowest MCS). As receiver with the lowest average SNR (i.e., 12.53\text{dB}) requires a bin of size 0.0559, which is significantly smaller than the largest most-effective bin size, the receivers are always grouped into one bin. Therefore, \textit{MEBC} performs similarly to \textit{Broadcast}.

Since in such a homogeneous scenario maximizing sum throughput is almost always the right strategy, \textit{Greedy} even slightly outperforms \textit{FH-OMB} for higher channel variability. Due to this variability, receiver states may differ enough so that \textit{FH-OMB}‘s conservative completion time estimate prevents it from opportunistically exploiting good channels as aggressively as \textit{Greedy}. This can be seen in more detail in Fig. 5, which shows average system throughput per time slot (averaged over all simulation runs and over both receivers, where receivers that finished have 0 throughput) for the scenario with channel variability $\sigma = 11.5\text{dB}$. Throughput of \textit{FH-OMB} starts out the same as that of \textit{Dyn-Prog} and \textit{Greedy}, but drops off slightly once receiver states becomes more heterogeneous and one receiver is close to finishing.

Fig. 6 shows the completion time estimates for the dynamic programming algorithm (left) and the \textit{FH-OMB} heuristic (right) for the same scenario (i.e., $\sigma = 11.5\text{dB}$). \textit{FH-OMB}‘s completion time estimate based on average channels underestimates completion time when the channel is more variable, but the relative differences in estimated completion time for the different states for the two algorithms are very similar. \textit{FH-OMB}‘s completion time estimation algorithm thus leads to the right beam-forming decisions in most cases. The performance gap is due to the fact that \textit{FH-OMB}‘s completion time estimate is slightly less “round” than the true estimate, making it appear more beneficial to stay close to the diagonal where both receivers have the same state.

It is interesting to note that the completion time increases for $1\text{dB} \leq \sigma \leq 11.5\text{dB}$ and then decreases again. When channel variation is low, both receivers are likely to finish at approximately the same time. The higher $\sigma$, the more likely it becomes that one receiver finishes earlier than the other, which increases completion time given by the maximum of the individual completion times. When increasing $\sigma$ even further, completion times reduce since with a good channel, only very few time slots are needed to complete. There is a significant probability that one of the receivers will finish very early, and the system can then serve the remaining receiver at a higher rate with the corresponding single-lobe beam.

2) \textbf{Heterogeneous Scenario:} For the heterogeneous scenario, we fix the $\gamma^{\text{SLB}}_{H_1} = 11\text{dB}$ and $\gamma^{\text{SLB}}_{L_1} = -1.4\text{dB}$ of the first receiver. For the second receiver, we vary $\gamma^{\text{SLB}}_{H_2}$ between 11\text{dB} and 31\text{dB} and $\gamma^{\text{SLB}}_{L_2}$ between $-1.4\text{dB}$ and 18.6\text{dB}, so that the two receivers become more and more heterogeneous as the channel values for the second receiver increase.

As the $\gamma^{\text{SLB}}_{H_2}$ and $\gamma^{\text{SLB}}_{L_2}$ increase, completion time decreases for all algorithms. \textit{Greedy} performs close to optimal for the first three data points where receivers are sufficiently homogeneous and the optimum strategy is to beamform to the receiver with high channel gain when one receiver has high channel gain and the other receiver has low channel gain. Here, beamforming to both receivers as \textit{Broadcast} and \textit{MEBC} (which groups receivers into a single bin) do, is again too
conservative.

The jump in Greedy’s completion for the next data point is because from this point on the good channel of the better receiver is so good that Greedy favors the receiver exclusively in that case and only serves both receivers when the good receiver finishes. In contrast, Broadcast’s strategy to balance the rates and forego opportunistic gain becomes closer and closer to optimal as the scenario becomes more heterogeneous and from an average SNR of $\gamma_{SLB} = 17\text{dB}$ on is the optimal strategy. The weak performance of Greedy can be explained from Fig. 8, where Greedy achieves high throughput until the first receiver finishes at less than approximately 18 time slots. The second receiver is still far from finishing as evidenced by the throughput curve flattening out around 30 time slots. FH-OMB performs close but is sub-optimal compared to Dyn-Prog, since the expected completion time is slightly inaccurate. The comparison in Fig. 9 shows that the expected completion time of FH-OMB algorithm (right) is less “round” than that of Dyn-Prog (left). Thus FH-OMB is more conservative and it sacrifices higher instantaneous rates to ensure that the relative difference in receiver state does not diverge too much.

To provide further insights into the behavior of the algorithms we show the state space visits in Fig. 10–13. As expected Broadcast keeps the two receivers very close to the diagonal where both receivers have the same amount of data, and slight deviations from the diagonal are only due to packet loss. Greedy in contrast makes quick progress until the second receiver finishes and for the remaining time only has the first receiver to serve. In fact, the steps with which the good receiver makes progress with Greedy can clearly be seen in Fig. 11. FH-OMB serves receivers similar to Dyn-Prog early on but then becomes too conservative as the good receiver progresses and beamforms more to the lagging receiver to balance receiver states.

B. Multipath Rayleigh Fading

In this section, we show simulation results for a flat multipath Rayleigh fading channel, where the channel does not change within a time slot. The Doppler shift for the Rayleigh channel is set to 10Hz, corresponding to a slow fading channel for receivers moving at walking speed. Receivers are randomly distributed within the coverage area. The BS transmit power is set to 43dBm. With this, a cell edge receiver that is 250m from the BS is able to receive a packet with the lowest MCS with an average probability of 30%. The block size $B$ is set to 6400bits.

We study the impact of increasing the number of receivers $N$ from 2 to 64 with different number of beamforming lobes (i.e., $K = \{2, 4, 8, 16\}$), again for a heterogeneous and a homogeneous scenario. Note that due to the high complexity of Dyn-Prog, we only compare the performance of the FH-OMB heuristic with that of Broadcast, Greedy and MEBC.

1) Random Receiver Distribution: We first discuss a heterogeneous scenario, where $N = \{2, 4, 8, 16, 32, 64\}$ receivers are randomly distributed within the cell area of radius 250m and for $K = 8$ beamforming lobes. The performance depends significantly on the specific receiver distribution, in particular for smaller numbers of receivers. For up to 16 receivers, Broadcast performs almost as good as FH-OMB since there is a high probability that there is one receiver with a significantly worse channel than the others (see Fig. 14). As the number of receivers increases, a higher number of receivers see similar channel conditions and as in the previous two-channel scenario, the performance of Broadcast degrades since it does not exploit opportunistic gain. However, in this heterogeneous scenario this effect occurs mainly for $N > 32$ receivers,
where Broadcast’s performance is significantly worse than that of Greedy and FH-OMB. Similar to the cases evaluated in Section V-A, the largest effective bin size is significantly larger than that required by the receivers. Therefore, the receivers are partitioned into the same group, which leads to a similar performance of MEBC to Broadcast. Greedy performs worse than Broadcast for small \( N \) for the same reason as above. The scenario is so small that the receivers are all very heterogeneous. As homogeneity increases for higher network densities, exploiting opportunistic gain becomes more important and Greedy outperforms Broadcast. FH-OMB performs well for all sizes of the receiver set. Its state-based completion time estimation results in the right tradeoff between opportunistic gain and multicasting gain and provides the lowest completion times of all approaches. It consistently outperforms Greedy by 9% to 29%. The performance gain over Broadcast and MEBC ranges from 1% to 76%.

Next, we look at the impact of varying \( K \) for a fixed \( N = 32 \). When increasing the number of beamforming lobes \( K \), the array gain of the single lobe beam increases as well. In the specific antenna configuration that is chosen for our simulation, the array gains for \( K = \{2, 4, 8, 16\} \) are 1.9, 3.4, 6.6 and 11.4, respectively. (Note that the array gain is not linear in \( K \).) Therefore, as observed from Fig. 15, completion time decreases with increasing \( K \) with respect to the achievable gain. FH-OMB outperforms both Greedy and Broadcast for all \( K \). However, increasing \( K \) has a more significant impact on the completion time of Broadcast than on Greedy and FH-OMB. For low \( K \) and a wider beamwidth, Broadcast is limited by the receiver with the lowest SNR in each beam. (Also, a significant amount of the radiated energy does not cover any receiver.) As \( K \) increases, fewer and fewer receivers are covered by a beam and in the extreme case of a single beam per receiver, Broadcast manages to perfectly balance the SNRs at the receivers (i.e., no energy is wasted by having a higher than necessary SNR at any receiver). Hence, Broadcast’s performance becomes closer and closer to FH-OMB. In contrast, Greedy may still beamform to a few receivers with high SNRs so that those finish first, before serving receivers with lower SNRs. In short, in heterogeneous scenarios with sufficient \( K \), Broadcast that favors the weaker receivers by multicasting to all the receivers performs better than Greedy that capitalizes in maximizing opportunistic gain.

To shed more light on the behavior of the algorithm, we show the CDF of completion time for the simulation runs for \( K = 8 \) and with \( N = 16 \) and \( N = 64 \) receivers in Fig. 16 and Fig. 17, respectively. In Fig. 16, Broadcast and Greedy have relatively similar completion time as FH-OMB in 10% of the simulation runs. This happens in scenarios where all receivers are distributed quite close to the BS and thus all receivers have a relatively homogeneous good average channel quality. When receivers are distributed sparsely within the cell radius, with high probability they have different average channel qualities. Under this scenario, Greedy performs badly since it opportunistically serves the better receivers first and therefore results in higher completion times than both FH-OMB and Broadcast. Here, FH-OMB receivers finish at 465 slots for the worst scenarios, whereas Greedy and Broadcast both require 570 and 840 slots, respectively. When the number of receivers increases, Fig. 17 shows that Broadcast no longer has most of its completion time close to FH-OMB in most of the simulation runs. In fact, around 20% of Broadcast’s completion time is similar to Greedy due to the limited number of beamforming lobes (\( K = 8 \), which leads to low multi-lobe beam’s SNR. Broadcast is particularly bad in scenarios where many of the receivers are relatively far from the BS (and thus more homogeneous). The worst case completion time of FH-OMB is at 675 slots, while Greedy and Broadcast require 810 and 2400 slots, respectively.

2) Cell Edge Receiver Distribution: In this scenario, receivers are all distributed close to the cell edge in the range of 190m to 220m and thus form a relatively homogeneous group. While such a scenario is less realistic than the one presented in Section V-B1, it is included to illustrate the performance degradation of the Broadcast and MEBC algorithms in more homogeneous scenarios. Note that this performance is also indicative of the performance in heterogeneous scenarios with very high user densities, where many receivers are at the cell edge (see Fig. 14).

Here, the performance differences are much more drastic and Broadcast performs worse than the other schemes already for \( N > 8 \) (see Fig. 18). For 32 receivers, FH-OMB outperforms Broadcast by 59%. Although maximizing instantaneous throughput is the right strategy for homogeneous scenario, FH-OMB still manages to slightly outperform the Greedy algorithm by about 1 − 10%. Despite the homogeneity of the scenario, the slight differences among the receivers require a more sophisticated mechanism that does take states into account. Similar to the scenario with heterogeneous receiver distribution in Section V-B1, completion time improves with increasing \( K \) (see Fig. 19) due to the higher effective SNR for each beam.

<table>
<thead>
<tr>
<th>Number of Receiver (N)</th>
<th>Completion Time (slots)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 4 8 16 32 64</td>
<td></td>
</tr>
<tr>
<td>Completion Time (slot)</td>
<td></td>
</tr>
<tr>
<td>200 400 600 800</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 14. Random receiver distribution, \( K = 8 \).

<table>
<thead>
<tr>
<th>Number of Beamforming Lobe (K)</th>
<th>Completion Time (slots)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 4 8 16</td>
<td></td>
</tr>
<tr>
<td>Completion Time (slot)</td>
<td></td>
</tr>
<tr>
<td>200 400 600</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 15. Random receiver distribution, \( N = 32 \).

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Completion Time (slots)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast</td>
<td></td>
</tr>
<tr>
<td>Greedy</td>
<td></td>
</tr>
<tr>
<td>FH-OMB</td>
<td></td>
</tr>
<tr>
<td>MEBC [7]</td>
<td></td>
</tr>
<tr>
<td>CDF</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 16. CDF of the completion time for random receiver distribution. \( N = 16 \).

Fig. 17. CDF of the completion time for random receiver distribution. \( N = 64 \).
3) Random Receiver Distribution in Large Cell: The performance of MEBC in Section V-B1 and Section V-B2 is similar to that of Broadcast due to the small required bin size of the worst receiver at each beam. To show an improved performance of MEBC against Broadcast, we simulate a scenario similar to that presented in [7] by increasing the cell radius to 260m. By doing so, the required bin size of the worst user increases and even the lowest MCS is not enough to serve all receivers at the same time. As a result, receivers are partitioned into multiple groups. By doing this, the receive power at the receivers is sufficiently high to successfully received packets transmitted using the lowest MCS. Indeed, we observe MEBC performs better than Broadcast by being less conservative. Broadcast always simultaneously beamforms to all receivers, resulting in extremely low gain at the edge receivers. However, FH-OMB consistently outperforms MEBC. Since the performance of MEBC is similar to Broadcast in all the scenarios used to evaluate fairness and feedback delay, the performance description of MEBC is excluded from the following sections.

C. Fairness of the Algorithms.

The fairness criteria that we use is the completion time of the receivers and it is computed based on Jain’s fairness index [34]. Given the completion time of each receiver $D_i$, fairness $\beta$ is

$$\beta = \frac{\left(\sum_i^N D_i\right)^2}{N \sum_i^N (D_i)^2}.$$  \hspace{1cm} \text{(18)}$$

Fig. 21 and Fig. 22 depict the fairness of the schemes when increasing the number of receivers $N$ and beamforming lobes $K$, respectively. Broadcast schedules all receivers and serves them with equal rate and thus it achieves the highest fairness in both settings. Unlike Broadcast, Greedy maximizes instantaneous rate, allowing receivers with better channels to complete receiving the data block before those with worse channels. This introduces a large difference in the individual completion times between the receivers and yields the lowest fairness among all schemes. FH-OMB achieves a much higher fairness than Greedy and limits the completion time differences between the receivers. However, FH-OMB yields slightly lower fairness than Broadcast since it jointly optimizes for instantaneous rate and completion time by serving a smaller subset of receivers than that of Broadcast which may cause a greater difference in the completion times.

For a fixed number of beamforming lobes, $K = 8$ (see Fig. 21), increasing $N$ increases the diversity of the receivers. For a throughput maximization scheme like Greedy, this increases the number of receiver groups to be served. By serving them sequentially (from the best to the worst channel group), Greedy creates a difference in completion time between the groups, particularly between the first and the last groups and thus yields a low fairness. FH-OMB trades off between multiuser diversity and multicast gain, thus it achieves better fairness than Greedy, with more receivers, the larger variation of completion time among the receivers also reduces fairness.

Increasing the number of beamforming lobes $K$ increases the single lobe array gain and receivers achieve a higher channel gain (see Fig. 22). For instance, two receivers (one better and one worse) that are located in the same beam may now be served with two individual beams when $K$ increases. This then increases the chance of both receivers to be served simultaneously without causing throughput loss. Both Greedy and FH-OMB benefit from this and they thus achieve a significant fairness improvement as the number of beamforming lobes increases.

In summary, although FH-OMB sacrifices up to 15% in terms of fairness, but in turn needs half the completion time of Broadcast as shown in Fig. 19.

D. Impact of Imperfect Feedback

To examine the impact of imperfect feedback, we introduce different intervals at which the feedback frames are sent: $\lambda = \{5, 10, 20, 40, 80, 160\}$ ms. Since the coherence time of the multipath Rayleigh fading channel is $\tau = 40$ ms, the actual channel state information and the one reported in the last feedback frame (for simplification, we call it last state
information) are correlated for up to $\tau$ ms and uncorrelated otherwise. Note that, although all receivers send the feedback frame every $\lambda$ ms, the slot at which the feedback frame is sent is asynchronous among the receivers.

In this section, we also include the corresponding schemes for Broadcast, Greedy, and FH-OMB in which the estimation algorithms explained in Section IV-C are applied. We call these schemes eBroadcast, eGreedy, and eFH-OMB, respectively. While eFH-OMB operates as described in Section IV-C, eBroadcast and eGreedy inherit their own mechanism for their decisions but use the estimated channels as explained in Section IV-C instead of the last state information.

**Impact of $\lambda$ on completion time:** Fig. 23 depicts the impact of increasing $\lambda$ on the completion time of the abovementioned schemes (with and without estimation). As $\lambda$ increases, the correlation between the current and last state information decreases, which causes a scheme to select an inaccurate weight for the beamforming pattern. A too high beam weight wastes resources and a too low one reduces reception probability.

The impact of increasing $\lambda$ on Broadcast is less severe compared to that of Greedy and FH-OMB because Broadcast is conservative. It always beamforms towards all receivers and transmits at a lower rate, thus reducing the magnitude of the error made. While Broadcast always schedules the receivers that have not received a complete data block (the subset of the scheduled receivers is fixed), Greedy and FH-OMB have a higher diversity of receiver subsets. Therefore, inaccurate state information causes Greedy and FH-OMB to not only make an error in the beam weights (like Broadcast) but also in the subset of scheduled receivers. These errors cause a more pronounced impact of $\lambda$ on the completion time of Greedy and FH-OMB than on Broadcast. Although FH-OMB performs similarly to Broadcast at $\lambda = 160$ms, it happens for different reasons. When feedback is largely outdated (i.e., the correlation between the actual and last state information is very low) but FH-OMB assumes feedback is perfect, this causes errors in the computation of the expectation completion time (see Section IV-C for further explanation). Consequently, FH-OMB chooses the wrong subset of receivers to be served and they are served with a wrong MCS, which leads to high completion time for larger $\lambda$. Greedy performs worst since it highly depends on the instantaneous channel knowledge to exploit the opportunistic gain. Without instantaneous channel knowledge, a beamforming pattern selected based on the last state information is no longer throughput-optimal.

As shown in Fig. 23, with the estimation algorithms detailed in Section IV-C, eFH-OMB, eGreedy, and eBroadcast achieve a substantial improvement over their corresponding schemes where estimation is not applied. For $\lambda < 20$ ms, eFH-OMB and eGreedy outperform eBroadcast since they can still exploit the opportunism because the actual and the last state information are correlated. When $\lambda$ is large, the variation of the actual channel is large and thus the estimated channel approaches the average channel (see Section IV-C and Fig. 3 for more details). In this case, opportunistic gain can no longer be exploited and the best strategy is to equally serve all the receivers. Therefore, we observe good performance of eBroadcast at $\lambda = 160$ms in Fig. 23 since serving at a wrong rate (low MCS) has minimal to no impact on the better receivers within the same beam lobe. eFH-OMB performs close to eBroadcast since the expected completion time restricts the opportunistic gain (as detailed in Section IV-C) and thus eFH-OMB is more conservative and schedules more receivers as eBroadcast does.

**Impact of $\lambda$ on the distribution of MCS:** As the chosen rate for the receiver served at each slot determines the performance of the algorithm, we take a further look into the distribution of the MCSs used by each scheme for increasing $\lambda$ as shown in Fig. 24. Note that there exists a correlation between the average number of successful receivers and the MCS distribution: when more of the higher MCSs are used (i.e., serving a smaller subset of better receivers), the average number of successful receivers is lower and vice versa. As depicted in Fig. 24, Broadcast and eBroadcast that serve more receivers mostly use the lower MCS and eGreedy uses higher MCSs which serve fewer receivers. In general, the distribution of MCSs is impacted by the remaining receivers in the system as well as the distribution of the estimated channel (for schemes with estimation algorithm).

For Broadcast, we observe a slight increase in the number of the higher MCSs when increasing $\lambda$. A higher $\lambda$ causes an error in the computation of the beam weight, thus some receivers lose the data packet and are served at a later time. At this later time, fewer receivers remain in the system and the BS has to beamform in fewer directions. This leads to higher gains for the beams and thus on average higher MCSs are used. With estimation, eBroadcast serves the receiver at the rate close to the average channel rate of the worst receivers at each beam lobe as $\lambda$ increases. Therefore, we observe an increasing number of intermediate MCSs (that are suitable for the estimated channels) as $\lambda$ increases. For Greedy, better receivers leave the system earlier, even for higher $\lambda$. Here, a wrong MCS choice has minimal impact on these receivers since they progress quickly once they are successfully served. However, the remaining (worse) receivers can only be served using the lower MCSs and wrong usage of MCSs causes more tries to successfully serve the subset of receivers which (based on the last received feedback) maximizes throughput. Since the number of attempts to serve the remaining receivers is higher than that to serve the receivers with excellent channels, more lower MCSs usage is seen from Fig. 24 for Greedy as $\lambda$ increases. Although based on the same distribution of

![Fig. 23. Impact of delay on completion time for random receiver distribution, $K = 8$, $N = 32$.](image-url)
estimated channels, eGreedy has a different MCS distribution (more higher MCSs) than eBroadcast due to the opportunistic nature of the algorithm. The frequency of higher MCSs reduces with increasing $\lambda$ since the opportunism is limited by the estimated channels.

We also observe an increasing number of higher MCSs for FH-OMB as $\lambda$ increases. For higher $\lambda$, the correlation between the actual and the last state information is reduced and FH-OMB can no longer compute its tradeoff correctly. The impact is twofold: (i) it requires more tries to serve a chosen receiver group, (ii) some receivers receive the data block at a much later time. Since FH-OMB is neither biased towards the worse nor the better receivers, it uses MCSs that are suitable for the receiver group, which in this case are the MCSs greater than MCS5. Therefore, when more tries are required, an increase in these MCSs is observed. With estimation, the number of intermediate MCSs of $eFH$-OMB increases with $\lambda$. This is however not usually due to the increased number of tries, but due to the distribution of the estimated channels (refer Section IV-C for more details) which limits the opportunistic gain. As a result, $eFH$-OMB conservatively serves the receiver as $eBroadcast$ does, therefore they have a similar MCS distribution, particularly for $\lambda = 160$ms.

VI. CONCLUSION

In this paper, we studied opportunistic multicast beamforming for the finite horizon problem, where a base station has a fixed amount of erasure-coded data to transmit to multiple receivers. We modeled the problem as a dynamic programming problem to obtain the optimal solution. Due the high complexity of this approach, we designed a heuristic algorithm, FH-OMB, that captures the characteristics of the optimal solution and provides a performance that is very close to it. We evaluated FH-OMB’s performance both for a discrete channel model as well as multipath Rayleigh fading.

We observed that in the more realistic Rayleigh fading scenario, the performance gains of our FH-OMB heuristic are much more pronounced than that in the simple scenarios with two receivers and two channel states. It outperforms other schemes based on maximizing the minimum SNR and broadcasting to all receivers (Broadcast), optimizing grouping of receivers (MEBC), as well as greedily maximizing sum rate (Greedy). Compare to Broadcast and MEBC, FH-OMB’s gain increases as the number of receivers increases since Broadcast does not exploit opportunistic gain. Also, FH-OMB is particularly beneficial over Greedy in heterogenous receiver scenarios because it trades off between multi-user diversity and multicast gain results in lower completion times. It improves performance by up to 76% over Broadcast and MEBC, and up to 29% over Greedy for heterogeneous scenarios with Rayleigh fading. For homogeneous scenarios, these gains are up to 122% and 10%, respectively. Similar (though slightly lower) gains are obtained for the simpler scenarios with a discrete channel model. This paper additionally addressed the impact of imperfect feedback and estimation algorithms are designed to counter the performance degradation due to this impact. With estimation, our proposed scheme ($eFH$-OMB) outperforms eGreedy and eBroadcast with estimation by up to 46.14% and 34.62%, respectively.

ACKNOWLEDGEMENT

This paper is supported by the LOEWE initiative (Hessen, Germany) within the NICER project, the German BMBF Ministry within CRISP, the European Research Council grant ERC CoG 617721, the Spanish Ministry of Economy and Competitiveness through grants Ramon y Cajal RFC-2012-10788 and HyperAdapt TEC2014-55713-R, and the Madrid Regional Government through the TIGRE5-CM program S2013/ICE-2919.

REFERENCES


