RESEARCH ARTICLE OPEN ACCESS

# Cooperation Versus Multiplexing: Multicast Scheduling Algorithms for OFDMA Relay Networks

HIMA BINDU K, K.M.RAYUDU
M.Tech Student, CMR Engineering College
Associate Professor, Dept Of CSE,CMR Engineering College

**Abstract—**
With the next-generation cellular networks making a transition toward smaller cells, two-hop orthogonal frequency-di-vision multiple access (OFDMA) relay networks have become a dominant, mandatory component in the 4G standards (WiMAX 802.16j, 3GPP LTE-Adv). While unicast flows have received rea-sonable attention in two-hop OFDMA relay networks, not much light has been shed on the design of efficient scheduling algorithms for multicast flows. Given the growing importance of multimedia broadcast and multicast services (MBMS) in 4G networks, the latter forms the focus of this paper. We show that while relay cooperation is critical for improving multicast performance, it must be carefully balanced with the ability to multiplex multicast sessions and hence maximize aggregate multicast flow. To this end, we highlight strategies that carefully group relays for cooperation to achieve this balance. We then solve the multicast scheduling problem under two OFDMA subchannelization models. We es-tablish the NP-hardness of the scheduling problem even for the simpler model and provide efficient algorithms with approxima-tion guarantees under both models. Evaluation of the proposed solutions reveals the efficiency of the scheduling algorithms as well as the significant benefits obtained from the multicasting strategy.

**Index Terms—**Orthogonal frequency-division multiple access (OFDMA), relay cooperation, scheduling, session multiplexing, wireless multicast.

## I. INTRODUCTION

WITH the next-generation wireless networks moving toward smaller (micro, pico) cells for providing higher data rates, there is a revived interest in multihop wireless networks from the perspective of integrating them with cellular networks. With a decrease in cell size, relay stations (RS) are now needed to provide extended coverage. In this context, two-hop relay-enabled wireless networks [Fig. 1(a)] have become a dominant, mandatory component in the 4G standards (WiMAX 802.16m [1], 3GPP LTE-Adv [2]) due to the plethora of envisioned applications (hotspots, office buildings, under-ground tunnel access, etc.) they support.

Orthogonal frequency-division multiple access (OFDMA) has become the popular choice for air interface technology in 4G networks. The entire spectrum is divided into multiple

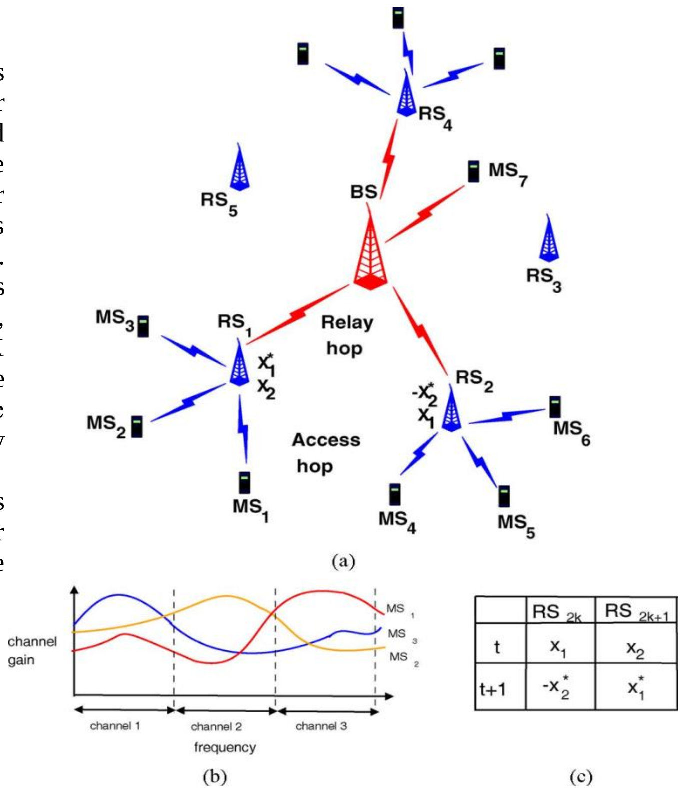


Fig. 1. System model and gains. (a) Network model. (b) User/channel diver-sity. (c) Relay cooperation.

carriers (subchannels), allowing for multiple users to operate in tandem. This leads to several physical-layer and scheduling benefits [3], [4]. The two-hop network model coupled with OFDMA provides several diversity (multiuser, channel, and cooperative) gains that can be leveraged through intelligent scheduling.

While several scheduling works [5]–[7] have focused on unicast traffic for two-hop OFDMA relay networks, multi-cast traffic has not been explored much in these networks. With 4G networks becoming a key component in the content delivery chain, multimedia broadcast and multicast services (MBMS [8]) are gaining importance as an efficient means to disseminate common information to subscribers. The design of efficient scheduling algorithms for multicast traffic forms a vital component of MBMS and in turn forms the focus of this paper. Multicasting in two-hop relay networks is significantly different from the conventional cellular multicast: The broad-cast advantage of multicast data is significantly diminished on the access (second) hop [Fig. 1(a)], where they become equivalent to multiple unicast transmissions from different RS to mobile stations (MS), thereby requiring more transmission resources. Relay cooperation mechanisms allow multiple RS to simultaneously transmit the multicast data on the same transmission resource. This helps retain the broadcast nature of the traffic on the access hop, making cooperation a critical component in improving multicast performance.

The key question, however, is the following: Is relay cooper-ation always beneficial? Interestingly, we show that there exists a subtle tradeoff between cooperation gains and the ability to multiplex multicast sessions effectively, both of which are es-sential for maximizing the aggregate multicast system perfor-mance. We highlight how strategies that carefully group relays for cooperation are needed to address this tradeoff effectively. We then solve the core multicast scheduling problem, which re-quires determining the allocation of subchannels to multicast sessions on both the relay and access hops such that both co-operation and multiplexing gains are leveraged to maximize the multicast system performance. In the process, motivated by re-cent relay standards [1], [2], [9], we consider two models for how subcarriers are grouped to form a subchannel in OFDMA: distributed (DP) and contiguous (CP) permutations. We estab-lish the NP-hardness of the scheduling problem even for the simplerDP modelandprovide efficientalgorithms with approx-imation guarantees for both models. Our contributions in this paper are multifold.

•We highlight and address the tradeoff between coopera-tion gain and effective multiplexing of multicast sessions through intelligent grouping of relays for cooperation.

• We provide LP-based algorithms with guarantees of for the DP model, and
for the harder CP model, where
is a small constant; , are the number of channels and relays. Evaluations reveal their close-to-optimal perfor-mance in practical scenarios.

• We also provide efficient, fast greedy algorithms for both the models, whose performance is very close to that of their LP-based algorithms.

We evaluate the proposed solutions in an event-drivensimulator that incorporates realistic physical-layer effects. Evaluations in-dicate the efficiency of the proposed scheduling algorithms as well as the significant benefits obtained from the overall multi-casting strategy that addresses the tradeoff between cooperation and session multiplexing.

The rest of the paper is organized as follows. The system de-scription is presented in Section II. The tradeoff between relay cooperation and session multiplexing is identified and addressed in Section III. Scheduling algorithms for the DP and CP models are presented in Sections IV and V, respectively. Practical con-siderations are presented in Section VI, followed by the evalu-ation of the solutions and concluding remarks in Sections VII
and VIII, respectively.

## II. SYSTEM DESCRIPTION

### *A. Related Work*

Relays: Several works [7], [10]–[13] have investigated the potential of relay-enabled wireless networks to provide improved coverage and capacity. Scheduling of unicast data has received higher emphasis [5]–[7], [10], [11], [13], [14] thus far in these networks. Most of the earlier works [10], [11] focused on TDMA variants where the scheduling decision re-duces mainly to deciding whether to employ a relay or not and for which particular user. They do not exploit multiple OFDM channels and the resulting diversity available across the relay and access hops. On the other hand, OFDM scheduling solu-tions for conventional cellular networks [3], [4], [15] cannot be directly extended to two-hop relay networks, where flow conservation across hops forms an important component. The more recent works [5]–[7] have looked at leveraging diversity and spatial reuse [16] gains in relays employing OFDMA. However, all these works are restricted to unicast data.

Multicasting: Unlike unicast works, the OFDMA scheduling works on multicast data have largely been restricted to one-hop cellular networks

[17], [18]. These solutions cannot be directly carried over to relay networks, where the nature of multicast traffic and its broadcast advantage is significantly altered on the access hop. Multicasting with relays has received increased attention recently. Information-theoretic works [19], [20] have looked at capacity bounds for a multicast system with relays. Use of network coding at relays to facilitate multicasting has also been studied in [21], [22]. Layered video, being a popular application for multicast, has been optimized for relays in [23] and [24]. While all these works have looked at various aspects of multicast transmission with relays, they do not incorporate OFDMA scheduling. In addition to making the problem significantly different, incorporation with OFDMA scheduling is also an important component in next-generation broadband access networks like LTE and WiMAX. In this direction, our prior work [25] considered the integration of multicast and unicast traffic in relay networks with OFDMA and provided some scheduling heuristics for the coexistence of heterogeneous traffic. However, it did not consider session multiplexing or its tradeoff with relay cooperation that arises within multicast scheduling and, hence, did not address the multicasting problem with relays rigorously.

Identifying and addressing this trade off by designing efficient multicast scheduling algorithms with performance guarantees for OFDMA relay networks is in turn the focus of this work.

*B. Network Model*

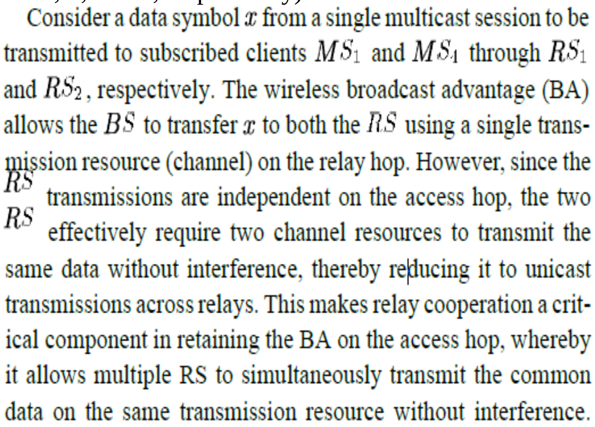
We consider a downlink OFDMA-based, relay-enabled, two-hop wireless network as shown in Fig. 1(a). A set of M MS are uniformly located within the macro cell. A small set of R RS are added to the midway belt of the network (R<M ). MS farther from the base station (BS) connect with the RS that is closest to them based on highest signal-to-noise ratio (SNR). The one-hop links between BS and RS are referred to as relay links, between RS and MS as access links, and between BS and MS as direct links (equivalent to relay links for scheduling purposes).

Downlink data flows are considered and assumed to origi-nate in the Internet and destined toward the MS. All stations are assumed to be half-duplex. Let $P_{B,}$ $P_R$ denote the maximum power used by the BS, RS for their transmission $(P_{\mathrm{R}} \leq P_{\mathrm{B}})$, which is split equally across all subchannels, and no power adaptation across channels is assumed, given the marginal gains resulting from it [26]. A set of total OFDM subchannels is considered, with two models for grouping of subcarriers to form a subchannel [1]: distributed permutation (DP) and contiguous permutation (CP). As the name suggests, the subcarriers con-stituting a subchannel are chosen randomly from the entire fre-quency spectrum in DP, while adjacent subcarriers are chosen in CP. In DP, a single channel quality value (averaged over en-tire spectrum), which is common to all its subchannels, is fed back by an RS/MS. This allows an RS/MS to employ a common rate on all subchannels. While the random choice of subcarriers in a subchannel eliminates channel diversity, it helps average out interference and reduce feedback. On the other hand, in CP, the high correlation in channel gains across adjacent subcarriers helps leverage subchannel diversity, whereby an RS/MS can employ different rates to suit different subchannel gains through scheduling. However, this requires feedback on all subchannels from RS/MS. Note that the measurement, feedback, and choice of rate levels (modulation and coding levels, MCS) are stan-dardized [1] for the two modes and directly provided by the MS (through RS) and RS to the BS in uplink frames, which the BS then directly uses for scheduling its transmissions to the RS and MS. Hence, for scheduling purposes, it suffices to model the rates being same (DP) or different (CP) on different subchan-nels for a user.

*C. Potential Gains*

Relay networks provide three forms of diversity gains. Con-sider the frequency response of three channels for three MS in

Fig. 1(b). Multipath fading and user mobility result in inde-pendent fading across users for a given channel, contributing to multiuser diversity. Furthermore, the presence of multiple channels and the corresponding frequency selective fading re-sults in different channels experiencing different gains for a given MS, contributing to channel diversity. These gains make it possible to schedule multiple users intandem, while providing good-quality channels to many of them (e.g., channels 3, 2, and 1 allocated to MS 1, 2, and 3, respectively).



Consider a data symbol $x$ from a single multicast session to be transmitted to subscribed clients $MS_1$ and $MS_4$ through $RS_1$ and $RS_2$, respectively. The wireless broadcast advantage (BA) allows the $BS$ to transfer $x$ to both the $RS$ using a single transmission resource (channel) on the relay hop. However, since the $RS$ transmissions are independent on the access hop, the two $RS$ effectively require two channel resources to transmit the same data without interference, thereby reducing it to unicast transmissions across relays. This makes relay cooperation a critical component in retaining the BA on the access hop, whereby it allows multiple RS to simultaneously transmit the common data on the same transmission resource without interference.

to enable relay cooperation given that it can be used in a dis-tributed manner [27]. The distributed nature eliminates the need for information exchange across relays. Of the two codewords

$$(\{x_1, -x_2^*\}, \{x_2, x_1^*\})$$

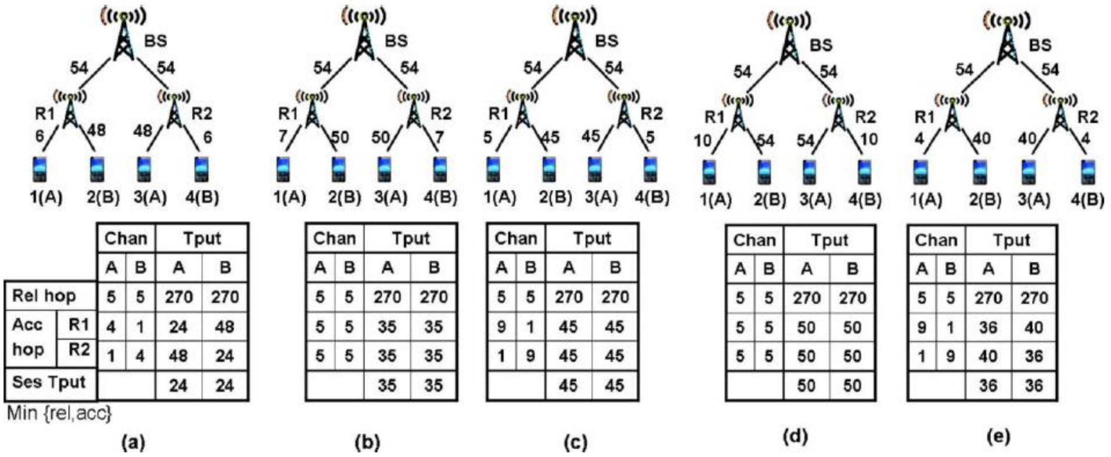
used by the scheme over two time-slots (for two symbols $x_1$,$x_2$) even-numbered relays transmit the first codeword, while odd-numbered relays transmit the second codeword during cooperation as shown in Fig. 1(c). This re-quires a single channel resource per data symbol while also increasing the received SNR at MS, a gain referred to as co-operative diversity (see [27] for details). While we consider the Alamouti scheme for cooperation, our scheduling solutions are equally applicable to other sophisticated cooperation strategies such as coordinated multipoint transmissions (CoMP) as well. CoMP is currently being standardized in LTE [2] (for release 11/12) and allows multiple transmitters (relays in our case) to cooperate and make a joint transmission to an MS, resulting in an SNR gain. Furthermore, precoded pilots (reference signals) are also made available for the MS to measure and report the rate in the presence of such cooperation.

*D. Scheduling Model*

Frame Structure: We consider a synchronized, time-slotted system (WiMAX, LTE) with BS and RS transmitting data in frames. Every frame consists of several time-slots and has to be populated with user assignments across channels for LTE (no channel sharing across slots) and user assignments across both time-slots and channels for WiMAX. To address both models generically, it is sufficient to consider the problem with one time-slot per frame since channels in other time-slots can be considered as additional channels available to the time-slot under consideration [6], [15]. Furthermore, the slotted frame structure allows us to decouple the scheduling of unicast and multicast traffic, with our focus being on the latter.

For multicast scheduling, assignments are made with respect to sessions, where multiple MS and corresponding RS can be subscribed to a session. K multicast sessions with backlogged buffers are considered (extensions to finite buffers is discussed in Section VI). As advocated in the relay standard [1], [9], each frame consists of a relay and an access zone, where the sched-uling of the half-duplex relays are time-divisioned with that of the BS, i.e., BS/RS-to-MS transmissions in the relay zone first followed by RS-to-MS transmissions in the access zone. Fur-thermore, simple receivers are considered at the MS and hence cooperation and combining of data transmission from the BS and RS to MS across frames is not leveraged. The BS is respon- sible for scheduling both the relay and the access hops in each frame, thereby resulting in per-frame schedules. While time di-visioning between the hops eliminates the reuse of channel re-sources across hops, it still allows for channel reuse to be lever-We consider the simple yet effective Alamouti space-time code

aged within the access hop through scheduling. The resulting session assignments to relay-hop channels for the current frame and the access-hop channels for the following frame are indi-cated by the BS to the RS and MS through a small control re-gion in the frame called the MAP. The MAP follows the pre-amble in the frame [1] and is transmitted at the lowest modula-tion and coding.



(a)

| | | Chan | | Tput | |
|---|---|---|---|---|---|
| | | A | B | A | B |
| Rel hop | | 5 | 5 | 270 | 270 |
| Acc hop | R1 | 4 | 1 | 24 | 48 |
| | R2 | 1 | 4 | 48 | 24 |
| Ses Tput | | | | 24 | 24 |

(b)

| Chan | | Tput | |
|---|---|---|---|
| A | B | A | B |
| 5 | 5 | 270 | 270 |
| 5 | 5 | 35 | 35 |
| 5 | 5 | 35 | 35 |
| | | 35 | 35 |

(c)

| Chan | | Tput | |
|---|---|---|---|
| A | B | A | B |
| 5 | 5 | 270 | 270 |
| 9 | 1 | 45 | 45 |
| 1 | 9 | 45 | 45 |
| | | 45 | 45 |

(d)

| Chan | | Tput | |
|---|---|---|---|
| A | B | A | B |
| 5 | 5 | 270 | 270 |
| 5 | 5 | 50 | 50 |
| 5 | 5 | 50 | 50 |
| | | 50 | 50 |

(e)

| Chan | | Tput | |
|---|---|---|---|
| A | B | A | B |
| 5 | 5 | 270 | 270 |
| 9 | 1 | 36 | 40 |
| 1 | 9 | 40 | 36 |
| | | 36 | 36 |

Fig. 2. Tradeoff illustration (all transmission rate values are in Mb/s). (a) No reuse. (b) Cooperation (1). (c) Reuse (1). (d) Cooperation (2). (e) Reuse (2).

our discussions with respect to only relay and access links. Direct links can be easily incorporated into the scheduling solutions by considering them as relay links without affecting performance guarantees.

*Objective:* The objective of our scheduling algorithms is to maximize the end-to-end multicast system throughput subject to the popular proportional fairness (PF) model. This can be captured as a utility maximization problem: max $\sum_k U_k$. The utility function of session $k$ corresponds to $U_k = \beta_k \log \bar{r}_k$ for PF, where $\beta_k$ captures the priority weight of the session's QoS class and $\bar{r}_k$ its average throughput. PF is widely adopted in the cellular domain as it strikes a good balance between throughput and fairness, while leveraging multiuser diversity. The system solution has been shown to converge to the optimum utility at long timescales if the scheduler's decisions at each epoch (frame interval) are made to maximize the aggregate *marginal* utility, $T_{\mathbf{max}} = \arg\max_T \left\{\sum_{k\in T} \Delta U_k\right\}$ [4], [28]. $\Delta U_k$ denotes the marginal flow (two-hop) utility received by session $k$ in a feasible schedule $T$. It is given by $\frac{\beta_k r_k^{\text{eff}}}{\bar{r}_k}$ for proportional fairness, thereby emphasizing users with good instantaneous channel conditions. $\bar{r}_k$ is updated as a moving average $\bar{r}_k(t+1) = (1-\frac{1}{\delta})\bar{r}_k(t) + (\frac{1}{\delta})r_k^{\text{eff}}(t)$, where $\delta$ is an exponential filtering coefficient.

$r_k^{\text{eff}}$ corresponds to the session's *two-hop flow* rate, which in turn is determined by the instantaneous *effective* rate on the relay and access hops combined. Let $r_k^{\text{rel}}$ and $r_k^{\text{acc}}$ be the net bit rates obtained for a session $k$ on the relay and access hops, respectively. The frame transmission results in $r_k^{\text{eff}} = \frac{\min\{r_k^{\text{rel}}, r_k^{\text{acc}}\}}{2}$ (assuming equal split of frame resources between relay and access hops), thereby accounting for flow conservation. We consider reliable multicast sessions, and hence the transmission rate for a session on a hop is assumed to be given by the minimum rate of its subscribed relays (users) on the relay (access) hop, respectively. If $\mathcal{M}, \mathcal{R}, \mathcal{K}$ denote the set of MS, RS, and sessions, respectively, then we have $r_k^{\text{rel}} = \min_{j\in R}\{r_{k,j}^{\text{rel}}\}$ and $r_k^{\text{acc}} = \min_{j\in R}\{r_{k,j}^{\text{acc}}\}$. Let $A_{k,j}^{\text{rel}}$ and $A_{k,j}^{\text{acc}}$ denote the set of channels assigned to relay $j$ for session $k$ on the relay and access hops, respectively, then we have $r_{k,j}^{\text{rel}} = \sum_{i\in A_{k,j}^{\text{rel}}} r_{k,j,i}^{\text{rel}}$ and $r_{k,j}^{\text{acc}} = \sum_{i\in A_{k,j}^{\text{acc}}} r_{k,j,i}^{\text{acc}}$, where $r_{k,j,i}^{\text{rel}}$ and $r_{k,j,i}^{\text{acc}}$ indicate relay and access-hop rates of session $k$ at relay $j$ on channel $i$, respectively. Since there could be multiple MS subscribed to the same session at a relay, we further have $r_{k,j,i}^{\text{acc}} = \min_{m\in\mathcal{M}:R(m)=j}\{r_{k,m,i}^{\text{acc}}\}$. The relay and access-hop rates of RS and MS, respectively, on a subchannel are assumed to be measured and fed back using approaches prescribed in the standard [1].

At each epoch $t$, weight $w_k(t) = \frac{\beta_k}{\bar{r}_k(t)}$ varies with $\bar{r}_k(t)$ (accounting for fairness). The core scheduling problem at the BS then reduces to determining the frame schedule that maximizes the following weighted sum rate:

$$T_{\mathbf{max}} = \arg\max_T \sum_{k\in T} w_k(t)\min\{r_k^{\text{rel}}, r_k^{\text{acc}}\}. \tag{1}$$

## III. MULTICASTING STRATEGY

### *A. Cooperation Versus Session Multiplexing*

While relay cooperation is critical for multicast, the key question, however, is the following: Is relay cooperation al-ways beneficial? Interestingly, there exists a subtle tradeoff between cooperation gains and the ability to multiplex multicast sessions effectively, both of which are essential for maximizing the aggregate multicast system performance. Consider the following example with two sessions and 10 channels on each hop (Fig. 2, $w_k = 1$). Users 1, 3 belong to session $a$, while 2, 4 belong to session $b$. The DP model is considered, where the transmission rate to a user (or relay) per channel does not vary across channels and are directly assumed as indicated in Fig. 2(a) on the relay ($r_{a,R1,i}^{\text{rel}} = r_{b,R1,i}^{\text{rel}}, r_{a,R2,i}^{\text{rel}} = r_{b,R2,i}^{\text{rel}}$) and access ($r_{a,R1,i}^{\text{acc}}, r_{a,R2,i}^{\text{acc}}, r_{b,R1,i}^{\text{acc}}, r_{b,R2,i}^{\text{acc}}$) hops for a single channel $i$. Note that the purpose of this example is to merely highlight the tradeoff—the actual magnitude of the gains resulting from addressing the tradeoff would in turn depend on various factors such as channel model, transmission power, etc. Furthermore, with relay-hop rates being significantly higher than the access-hop rates in our example, the access hop forms the bottleneck, whose performance consequently depends on the scheduling strategy employed.

In the basic no-reuse strategy (**NR**), multicast data reduces to unicast on the access hop, requiring the available channels to be split both across relays and across sessions within a relay. This results in a channel split of (4, 1, 1, 4) channels to users (1, 2, 3, 4), respectively, providing a per-session throughput of 24 Mb/s and a net throughput of 48 Mb/s as indicated in Fig. 2(a). When relay cooperation (**C**) is leveraged for a session, simultaneous cooperative transmission from both the relays occur on the same channel to increase the SNR gain at the MS, which allows for a higher rate to be used on a channel on the access hop (e.g., assume 6 Mb/s can be increased to 7 Mb/s for users 1, 4, and 48 Mb/s increased to 50 Mb/s for users 2, 3) as shown in Fig. 2(b). Although transmissions across relays carry the same data on the same channel for a given session, there will be mutual interference if the cooperative transmissions occur at different rates. Hence, the cooperative transmissions have to happen at the same rate (7 Mb/s), namely that of the bottleneck user in the session (user 1 in session 1 and user 4 in session 2). This results in an allocation of five channels for each session with users (1, 2, 3, 4) receiving an allocation of (5, 5, 5, 5) channels, where the five channels are reused across relays within a session (between users 1 and 3 in session A, and 2 and 4 in session B) through cooperation. This provides a per-session throughput of 35 Mb/s and hence a net throughput of 70 Mb/s, which is a gain of about 45% over the baseline.

Now, consider an alternate reuse strategy (**R**), where the available channels on the access hop are reused at each relay. However, instead of coupling themselves through

(e.g., assume 6 Mb/s reduces to 4 Mb/s for users 1, 4, and 48 to 40 Mb/s for users 2, 3), and translates to increased rates (e.g., assume 6 Mb/s increases to 10 Mb/s for users 1, 4, and 48 to 54 Mb/s for users 2, 3) when cooperation is leveraged with a correspondingly increased session bottleneck rate (10 Mb/s). Here, cooperation provides a higher per-session throughput of 50 Mb/s, delivering a net

### *B. Cooperating Relay Components*

To strike a good balance between cooperation and multi-plexing gains, we need an intelligent combination of coopera-tion and reuse strategies. This requires that we first partition the set of active relays into subsets, where: 1) there is negligible interference across relay subsets that promotes better session multiplexing through channel reuse across subsets; 2) the ap-preciable interference within subsets necessitates cooperation between the member relays serving the same session. We define a relay to be active if it has at least one user subscribed to a mul-ticast session. While relays with no subscribed clients can aid the transmissions in neighboring relays through cooperation, they also reduce the potential gain from session multiplexing by creating more interference and are hence not considered. However, the algorithms can be easily adapted to incorporate inactive relays as well.

The following simple mechanism (PART) helps achieve such a partition with the help of measurement and reporting cooperative transmissions (at the bottleneck user rate in the session), the relays operate independently at their respective rates subject to the interference that arises. The resulting channel rates are reduced on the access hop due to interference (e.g., assume 6 Mb/s reduces to 5 Mb/s, and 48 to 45 Mb/s) as indicated in Fig. 2(c). However, decoupling the relays' transmissions now allows us to efficiently leverage the high rates experi-enced by the session at different relays by allocating varying number of channels across relays unlike in cooperation. This in turn enables statistical multiplexing of sessions, which allows an asymmetric channel allocation to even users within a session, resulting in an allocation of (9, 1, 1, 9) channels to users (1, 2, 3, 4), respectively. Here, the 10 channels are reused at both the relays without any cooperation. This provides a per-session throughput of 45 Mb/s and a higher aggregate multicast flow of 90 Mb/s as shown in Fig. 2(c). This is a gain of about 30% over relay cooperation, which we refer to as the session multiplexing gain. Note that this statistical multiplexing gain comes at the cost of cooperation gain and interference. Hence, scenarios where users are closer to their associated RS than to the interfering RS (e.g., user clustering in hotspots) are appropriate for leveraging multiplexing gain, where the loss due to interference and consequently also the gain from cooperation tends to be low. On the other hand, when interference across relays is high, the benefits from cooperation outweigh multiplexing gains. This is evident from an alternate (higher interference) example in Fig. 2(d) and (e), where the high interference between relays reduces access-hop throughput of 100 Mb/s. This is a 35% gain over the 72-Mb/s throughput delivered by reuse strategy.

Thus, given a transmit power, every relay pair must deter-mine if the rate loss due to interference is significant enough to translate it to a rate gain through cooperation (C), or sustain the interference to leverage session multiplexing gain through channel reuse (R). function-alities provided in the relay standard [1], [9] [Fig. 1(a) is used as a running example].

Step1)BS instructs each active e RSj(RS1,RS2,RS4) to transmit training symbols (pilots) on a selected subset of channels in isolation. All associated (unas-sociated) MS $m$ measure the corresponding signal $\rho_m$ (interference $\rho^j_{m,c}$ ) and noise $\sigma_{m,c}$ power and report it to its RS.

Step 2) Let the RS represent vertices of a graph, with edges between vertices indicating the presence of appre-ciable interference between corresponding relays. The number of disjoint connected components in this graph [two in Fig. 1(a)] gives the number of relay subsets ({RS1,RS2},{RS4}) that cause neg-ligible interference to each other.

it to the RS, which can be achieved through standard measure-ment (from pilots) and reporting (in uplink frames) mechanisms available in the relay standard [1]. Also, the relay grouping mechanism runs at a much coarser timescale (several seconds) compared to scheduling, allowing its overhead to be amortized over several hundreds of frames. As a further optimization, the MS do not have to feed back all the interference information to the RS; each MS can make their local interference decisions themselves (based on thresholding), determine the set of neigh-boring RSs that cause interference, and report back only the in-terfering set of RS. From aggregated information, the RS can then determine which of its neighboring RS cause interference to at least a fraction of its MS. Thus, feedback overhead can be significantly reduced.

Our joint multicast strategy (JRC): 1) uses PART to first determine the relay subsets; and 2) solves the core multicast scheduling problem to enable cooperation between relays ($RS_1$, $RS_2$) serving the same session within each subset, and leverages low interference across subsets to reuse channels across subsets without any cooperation (coupling) to enable session multiplexing. Since our main focus is to address the challenging scheduling problem, we use simple mechanisms to determine the relay subsets (PART) as well as MS association (based on high SNR). However, more optimized approaches for relay grouping and MS association can also be used with our scheduling solutions, but are beyond the scope of this work.

### *C. Core Scheduling Problem*

Given the relay subsets, the scheduling objective of JRC can be made more specific as follows:

$$T_{\max} = \arg\max_{T} \sum_{k \in T} w_k(t) \min\{r_k^{\text{rel}}, r_k^{\text{acc}}(1), \ldots, r_k^{\text{acc}}(C-1)\} \quad (2)$$

where $C - 1$ is the total number of relay subsets (components) on the access hop ($C \leq R+1$ with $R \leq 4$ typically) and $r_k^{\text{acc}}(c)$ indicates the access-hop rate for session $k$ in relay component $c$. The relay hop contributes an additional component (subset), the difference being that all active relays are part of this component, where no cooperation is possible. Hence, the core scheduling problem in **JRC** now reduces to determining an allocation of $N$ channels to $K$ sessions on each of the $C$ components, such that the weighted sum of the minimum session rates (accounting for flow conservation) across components is maximized. We address this scheduling problem under the DP and CP models in the following sections.

Note that the baseline strategies, cooperation (**C**) and reuse (**R**), are obtained as special cases of our generic formulation: one component on the access hop with all $R$ relays ($C = 2$), and $R$ components on access hop with one relay each ($C = R + 1$), respectively. While session multiplexing is available even with $C = 2$ components, larger $C$ provides more room for multiplexing, but decreases the gain from cooperation.

## IV. MULTICAST SCHEDULING UNDER DP

With distributed permutation, all channels of a session experience the same rate in a component (due to cooperation), but vary across components. The scheduling problem (MDP) can be formulated as the following integer program (IP):

$$\begin{array}{lll} \text{MDP: Maximize} & \overset{K}{\underset{k=1}{}} \quad A_k & \\ \text{subject to} & \qquad \geq A_k \quad \forall k \in \qquad c \in & \\ & \sum_{k} X_{k,c} \leq N \quad \forall c; \quad X_{k,c} \in \{0, 1, \ldots, N\}. & \end{array}$$

$F_{k,c}$ represents the weighted effective rate of session $k$ in component $c$, i.e., $F_{k,c} = w_k r_k(c)$, where $r_k(c) = \min\{r_k^{\text{rel}}, r_k^{\text{acc}}(c)\}$ is the bottleneck rate of the session in component $c$ that takes into account cooperation and interference. $A_k$ captures the session's (weighted) effective bottleneck rate, which we also refer to as flow (since it is over two hops). Thus, the goal is to maximize the aggregate flow that can be delivered to multicast sessions. The first constraint captures flow conservation, where the flow received by a multicast session is restricted to the minimum flow ($A_k$) across all components. Furthermore, while multiple channels can be given to a session ($X_{k,c}$), the total across sessions is restricted to $N$ in each component (second constraint). The session's weight (  ) is folded into its modified flow rate in each component (  ).

### *A. Hardness of MDP*

*Theorem 1:* MDP is NP-hard even for two components.

*Proof:* Consider the decision version of the problem (optimization version being harder) 2MDP: *Given $N$ channels each in two components, is there a feasible schedule of value $S$?*

Consider the following version of unbounded knapsack problem, where elements with weight ($w_i$) have the same unit profit ($\frac{p_i}{} = q, \forall i$), and the knapsack has a capacity $N$ ($w_i, N \in \mathcal{Z}$). The corresponding decision problem (DKP) is the following: *Is there a subset ($L$) of elements (with repetition) such that $\sum_{i \in L} w_i = N$ ?* This decision problem is known to be NP-complete. We will provide a reduction from DKP to 2MDP. Given an instance of DKP, without loss of generality (wlog) assume $w_i > 2$, since otherwise $N$ and each $w_i$ can be scaled by a constant $a$ such that $aw_i > 2$. Construct an instance of 2MDP as follows: For every element $i$, create two sessions with rates on the two components (relay, access) as

and $\left(\frac{qw_i}{w_i - 1}, qw_i\right)$ for the two sessions, respectively. Now, with $N$ channels, our 2MDP scheduling problem achieves a value of $S = 2Nq$ only if there exists a subset of elements $L$ in DKP whose weight equals $N$ ($\sum_{i \in L} w_i = N$). ■

Furthermore, the inclusion of more components ($> 2$) and the CP model makes the problem harder.

### *B. LP-Based Algorithm: LSDP*

We propose the following linear program (LP)-based polyno-mial-time algorithm (LSDP) to solve MDP.

**Algorithm 1:** Multicast Scheduler under DP: LSDP

```
1:  Solve the LP relaxation of MDP with solution $X^*_{k,c}$, $A^*_k$.
2:  $\mathcal{C} = \{1, \ldots, C\}$
3:  while $\mathcal{C} \neq \emptyset$ do
4:      Loss due to integrality restoration.
5:      for $c \in \mathcal{C}$ do
6:          $Z_{k,c} = 0, \ \forall k, i$ and $B_k = A^*_k, \ \forall k$.
7:          for $i \in [1, N]$ do
8:              $Z_{k',c} = Z_{k',c} + 1$, where $k' =$ $\arg\max_k \{\min\{F_{k,c}, B_k\}\}$
9:              $B_{k'} = B_{k'} - \min\{F_{k',c}, B_{k'}\}$
10:         end for
11:         $L_c = \sum_k \{A^*_k - Z_{k,c}\} \cdot F_{k,c}$
12:     end for
13:     Integral allocation for component with smallest loss.
14:     $c' = \arg\min_{c \in \mathcal{C}} L_c$
15:     Update $\hat{X}_{k,c'} \leftarrow Z_{k,c'}, \forall k$
16:     Update $A^*_k = \min\{A^*_k, F_{k,c'} \cdot \hat{X}_{k,c'}\}, \forall k; \mathcal{C} \leftarrow \mathcal{C} \backslash c'$
17: end while
```

LSDP first solves the LP relaxation of MDP with $X_{k,c} \in [0, N]$ (step 1). Let the solution be $X^*_{k,c}$ and $A^*_k$ with net optimal flow being $\sum_k A^*_k$. $X^*_{k,c}$ gives the net fractional channel allocation to session $k$ on component $c$, with $\sum_k X^*_{k,c} \leq N$. However, some of the channels may be fractionally shared between sessions in each component, whose integrality needs to be restored for a feasible schedule. For each component $c$, we determine the loss due to integrality restoration (steps 4–12). This requires a new integral channel allocation ($Z_{k,c}$) for each component $c$. With $A^*_k$ as the maximum flow limit for session $k$, we assign each channel to the session yielding the largest flow in the component based on its remaining flow limit (steps 7–10). Alternately, $Z_{k,c}$ can be directly derived from $X^*_{k,c}$ by removing only the required number of channel allocations with the smallest flow, thereby eliminating the dependence on $N$. The loss resulting from this integral allocation is then determined with respect to the optimal fractional allocation (step 11). The component yielding the smallest loss is chosen ($c'$), and the corresponding integral allocation ($\hat{X}_{k,c'}$) is determined (steps 13–16). The procedure is repeated until integral channel allocation is restored to all components.

In characterizing the performance of LSDP, we first establish the following lemma.

*Lemma 1:* The optimal solution to the LP relaxation of MDP has at most $\min\{K, C\}$ sessions with fractional channel allocation in each component.

*Proof:* Assume $C < K$, otherwise the statement is trivial. Consider the equivalent formulation for the LP relaxation of MDP

$$\text{R-MDP: Maximize} \quad \sum_{k=1}^{K} N\hat{F}_k Y_k$$

$$\text{subject to} \quad \sum_{k=1}^{K} \frac{\hat{F}_k Y_k}{F_{k,c}} \le 1 \quad \forall c; \qquad Y_k \le 1 \quad \forall k$$

where $\hat{F}_k = \min_c\{F_{k,c}\}$. The output to R-MDP gives us the allocation ($X_{k,c_k} = NY_k$) to the bottleneck component ($c_k = \arg\min_c\{F_{k,c}\}$) of each session ($k$), with the allocation to its other components being scaled by their respective flow weights ($\frac{\hat{F}_k X_{k,c_k}}{F_{k,c}}$). R-MDP has $K + C$ constraints and $2K + C$ variables (including slack variables to convert inequalities to equalities). Interestingly, R-MDP can be viewed as the LP-relaxation of a multidimensional knapsack problem (R-MKP) with $C$ dimensions, the capacity limit of each dimension being 1, and each session having a $C$ dimensional weight ($\frac{\hat{F}_k}{F_{k,c}} \le 1$). The optimal solution for such a R-MKP formulation can be shown ([29, Lemma 9.2.1]) to admit at most $C$ (out of original $K$) variables with fractional values.

We still need to establish that when these fractional allocations from R-MDP are scaled to obtain the allocations to sessions on each component in the original LP relaxation of MDP, there are at most $C$ fractional allocations in each component. To see this, when $Y_k$ is not fractional, either $Y_k = 0$ (not allocated) or $Y_k = 1$. However, when $Y_k = 1$, the inequality on the bottleneck component for session $k$ becomes an equality, preventing any other session from receiving any allocation on any of the components. Hence, there can be at most $C$ sessions with nonzero allocation in the solution to the LP relaxation for MDP, which also limits the number of fractional allocations in each component to $C$. ■

*Theorem 2:* LSDP provides a performance guarantee of $\max\left\{\frac{1}{2}, \left(1 - \frac{C(C-1)}{2N}\right)\right\}$ in the worst case.

*Proof:* We will first bound the loss in performance due to rounding in each component $c$, followed by the loss across components.

*Loss Per Component:* Let the optimal (fractional) channel allocations ($X^*_{k,c}$) for component $c$ be represented in terms of their integer ($I_{k,c}$) and fractional ($q_{k,c}$) parts as $X^*_{k,c} = I_{k,c} + q_{k,c}$. We have $\sum_k X^*_{k,c} = N$. In translating the fractional solution to integral, while $I_{k,c}$ is achievable as is, some flow from the session will be lost in converting $q_{k,c}$ to integral allocations. Note that, by Lemma 1, there are at most $C$ sessions with fractional allocation in each component in the optimal solution. Hence, at most $\frac{C}{2}$ channels can be shared resulting in $\sum_k q_{k,c} \le \frac{C}{2}$. Wlog, consider $N - 1$ channels to be assigned integrally with only one channel being fractionally shared ($\sum_k q_{k,c} = 1$) by $m$ sessions, $m \le C$. The proof can be extended in a straightforward manner to multiple fractionally shared channels. Now, if $q_{j,c}F_{j,c} \ge q_{k,c}F_{k,c}, \quad \forall k \ne j$, LSDP would assign the fractional channel to session $j$. Hence, the following flow is achievable by LSDP in component $c$:

$$\hat{F}_c \ge \sum_k X^*_{k,c}F_{k,c} - \sum_k q_{k,c}F_{k,c} + q_{j,c}F_{j,c}. \tag{3}$$

Of the $m$ sessions sharing the channel, at most one session (say $i$) will receive an allocation of $q_{i,c} \ge \frac{1}{2}$. Since $q_{i,c}F_{i,c} \le q_{j,c}F_{j,c}$, we have

$$\begin{aligned}\hat{F}_c &\ge \sum_k X^*_{k,c}F_{k,c} - \sum_k q_{k,c}F_{k,c} + q_{i,c}F_{i,c}\\ &\ge \sum_k X^*_{k,c}F_{k,c} - \sum_k \frac{F_{k,c}}{2} + \frac{F_{i,c}}{2}.\end{aligned}$$

Let $\Phi, \Pi$ represent the ordering of $X^*_{k,c}F_{k,c}$ in decreasing value and $F_{k,c}$ in increasing value, respectively. Now, pairing the corresponding index elements from the two orderings, we have

$$\hat{F}_c \ge \sum_k X^*_{\Phi(k),c}F_{\Phi(k),c}\left(1 - \frac{F_{\Pi(k),c}}{2X^*_{\Phi(k),c}F_{\Phi(k),c}}\right) + \frac{F_{i,c}}{2}. \tag{4}$$

For all $k_1 < k_2$, we have $X^*_{\Phi(k_1),c}F_{\Phi(k_1),c} \ge X^*_{\Phi(k_2),c}F_{\Phi(k_2),c}$ and $F_{\Pi(k_1),c} \le F_{\Pi(k_2),c}$. Hence

$$\frac{F_{\Pi(k_1),c}}{2X^*_{\Phi(k_1),c}F_{\Phi(k_1),c}} \le \frac{F_{\Pi(k_2),c}}{2X^*_{\Phi(k_2),c}F_{\Phi(k_2),c}}.$$

Thus, with $\sum_k q_{k,c} = 1$, we find that the fractional allocation of the channel is such that more flow is retained from sessions with larger flow values than from those with smaller flows. Hence, maximum flow is lost during integrality restoration when $\frac{F_{\Pi(k),c}}{2X^*_{\Phi(k),c}F_{\Phi(k),c}}$ are the same for all $k$, which results in $F_{k,c} = F, \forall k$ and $X^*_{k,c} = \frac{N}{m}, \forall k$. Thus, we have

$$\hat{F}_c \ge NF - \frac{(m-1)F}{2}, \qquad \text{where } F^*_c = \sum_k \frac{NF}{m} = NF$$

$$\Rightarrow \hat{F}_c \ge \left(1 - \frac{(m-1)}{2N}\right)F^*_c.$$

Note that when more channels are fractionally shared, one session's fractional flow is retained for each channel. More importantly, the number of sessions sharing a channel $m$ decreases, making the guarantee better (best when $m = 2$). The worst case is achieved when one channel is fractionally shared by all $C$ sessions ($m = C$), resulting in a guarantee of $1 - \frac{(C-1)}{2N}$.

*Loss Across Components:* The integral flow resulting from the first component $c_1$ in LSDP is

$$\hat{F}_{c_1} \geq \left(1 - \frac{(C-1)}{2N}\right) \sum_k X^*_{k,c_1} F_{k,c_1} . \qquad (5)$$

Each session's integral flow from $c_1$ is used to limit the maximum flow ($A^*_k$) for the corresponding session in the next component $c_2$. Since the updated $A^*_k \leq X^*_{k,c_2} F_{k,c_2}$, the updated flow ($\hat{F}_{c_1}$) is also achievable with fractional allocations. Applying LSDP to $c_2$, we obtain

$$\hat{F}_{c_2} \geq \left(1 - \frac{(C-1)}{2N}\right) \hat{F}_{c_1} \geq \left(1 - \frac{(C-1)}{2N}\right)^2 \cdot \sum_k X^*_{k,c_1} F_{k,c_1} . \qquad (6)$$

Extending the argument to $C$ components, the net integral flow from LSDP is given by

$$\hat{F}_{\mathrm{LSDP}} \geq \left(1 - \frac{(C-1)}{2N}\right)^C \cdot \sum_k X^*_{k,c_1} F_{k,c_1} . \qquad (7)$$

This results in a net worst-case performance guarantee of at least $\left(1 - \frac{C(C-1)}{2N}\right)$, which gets tighter and better with increasing $N$.

However, the above guarantee is loose for smaller number of channels, where an alternate bound of $\frac{1}{2}$ can be established. If a channel is fractionally shared by $m$ sessions, then at most one out of $m$ sessions has a fractional allocation of $\geq \frac{1}{2}$. Thus, reducing each fractional allocation by $\frac{1}{2}$ and rounding to the nearest integer will automatically result in the channel being assigned to only one of the $m$ sessions. This simple rounding procedure retains an aggregate flow that is at least half the original value and can be used to bound LSDP's performance. When $N \leq C$, the above rounding mechanism coupled with the selection of at most $N$ sessions with the largest flow value out of (at most) $C$ will provide a guarantee of $\frac{N}{2C}$.

Thus, LSDP provides a performance guarantee of $\max\left\{\frac{1}{2}, \left(1 - \frac{C(C-1)}{2N}\right)\right\}$ when $N \geq C$, which is the case of practical interest, where the number of channels will be much more than the number of components and relays. ■

### C. *Greedy Algorithm: GSDP*

While the standard [1] allows for both DP and CP models, support for DP has been made mandatory due to its simplicity. Furthermore, recall that the scheduler has to run at the granularity of frames (5 ms in WiMAX, 1 ms in LTE). Hence, having a fast but efficient scheduling algorithm for the DP model that can operate in the absence of an LP solver (unlike LSDP) is useful from an implementation perspective. To this end, we provide a greedy algorithm (GSDP), whose average-case performance is very close to that of LSDP in practice (illustrated in Section VII).

GSDP leverages the following observation pertaining to the structure of the optimal LP fractional solution: When all $N$ channels are assigned to the session with the highest bottleneck flow ($k^* = \arg\max_k\{\min_c F_{k,c}\}$), the bottleneck component of $k^*$ uses up all $N$ channels, while the remaining components remain underutilized. To efficiently utilize (pack) $N$ channels on all $C$ components, more multicast sessions need to be multiplexed such that their respective bottlenecks occur in different components. This entails $k^*$ to sacrifice some channels on its bottleneck component, which can then be used by other sessions (with a bottleneck in other components) to deliver (pack) a higher flow per unit channel. GSDP uses this observation to greedily assign channels on a per-session basis across all components as follows.

**Algorithm 2:** Greedy Scheduler under DP: GSDP

```
1:  $A_{k,c} = 0, E_{k,c} = 0, \forall k, c;\ valid\_ses = 1,$
    $\mathcal{K} = \{1, \ldots, K\}$
2:  $U_{k,c} = \frac{F_k^{\min}}{F_{k,c}}, \forall k, c$, where $F_k^{\min} = \min_c F_{k,c}$
3:  Available channels, $M_c = N, \forall c$
4:  while $valid\_ses == 1$ do
5:      for $k \in [1, K]$ do
6:          $S_{k,c} = [U_{k,c} - E_{k,c}]^+, \forall c$, where $[x]^+ = \max\{x, 0\}$
7:          if $\Pi_c(M_c + E_{k,c}) == 0$ then $\mathcal{K} \leftarrow \mathcal{K} \backslash k$ end
8:      end for
9:      if $\mathcal{K} \neq \emptyset$ then
10:         $k' = \arg\max_{k \in \mathcal{K}} \frac{\min_c\{F_{k,c} M_c\}}{\sum_c M_c}$
11:         $A_{k',c} = A_{k',c} + 1$, if $S_{k',c} > 0, \forall c$
12:         $E_{k',c} = \frac{A_{k',c} F_{k',c} - \min_c\{A_{k',c} F_{k',c}\}}{F_{k',c}}, \forall c$
13:         $M_c = N - \sum_k A_{k,c}, \forall c$
14:     else
15:         $valid\_ses = 0$
16:     end if
17: end while
```

At every iteration, GSDP selects the session that delivers the maximum flow per unit channel when all the remaining channels in each component ($M_c$) are taken into account (step 10). Sessions that do not have any resource remaining on any component ($M_c + E_{k,c} = 0$) are not considered in the selection (step 7). Once a session is selected, a channel is allocated to the session ($A_{k,c}$) only on those components that do not already have excess channel resource ($S_{k,c}$) to accommodate a unit of the bottleneck flow $U_{k,c}$ (steps 6 and 11). Furthermore, after allocation, components that receive more flow than the bottleneck (due to integral allocation) update the excess resource available ($E_{k,c}$) for the session on the respective component (step 12). This excess flow is then used for subsequent allocations. The remaining channels for allocation are updated on each component after every iteration. GSDP terminates when no session has any remaining resource to accommodate a unit of its bottleneck flow ($\mathcal{K} = \emptyset$, step 15). It is easy to see that GSDP has a time complexity of $O(KNC^2)$, which is linear in $K$ and $N$, with $C$ being a small constant. The dependence on $N$ can be reduced to $\log N$ by increasing the channel allocation granularity to $\left\lceil \frac{N}{\log N} \right\rceil$ channels in every iteration.

## V. MULTICAST SCHEDULING UNDER CP

Unlike the distributed permutation model, in contiguous per-mutation, channels of a session experience different rates both within and across components. The corresponding scheduling problem (MCP) can be formulated as the following IP:

$$\text{MCP: Maximize} \quad \sum_{k=1}^{K} A_k$$

$$\text{subject to} \quad \sum_{i=1}^{N} F_{k,c,i} X_{k,c,i} \geq A_k \qquad \forall k, c$$

$$\sum_k \sum_c X_{k,c,i} \leq 1 \quad \forall i; \qquad X_{k,c,i} \in \{0, 1\}$$

where $F_{k,c,i} = w_k r_{k,i}(c)$. MCP is similar to MDP except that session flow rates ($F_{k,c,i}$) are now a function of the channel as well. This makes the problem and the design of efficient algorithms significantly harder under CP.

### *A. LP-Based Algorithms: LSCP1, LSCP2*

*then end*

**Algorithm 3:** Multicast Scheduler under CP: LSCP1

```
1:  Solve the LP relaxation of MCP ($X_{k,c,i} \in [0, 1]$) with
    solution $X^*_{k,c,i}$ and $A^*_k$.
2:  $\mathcal{C} = \{1, \dots, C\}$
3:  while $\mathcal{C} \neq \emptyset$ do
4:      Loss due to integrality restoration.
5:      for $c \in \mathcal{C}$ do
6:          $Z_{k,c,i} = 0, \ \forall k, i; B_k = A^*_k, \ \forall k; I = \{1, \dots, N\}$.
7:          while $I \neq \emptyset$ do
8:              $(k', i') = \arg\max_{k, i \in I} \{\min\{F_{k,c,i}, B_k\}\}$
9:              $Z_{k',c,i'} = 1$'
10:             $B_{k'} = B_{k'} - \min\{F_{k',c,i'}, B_{k'}\}; I \leftarrow I \backslash i'$
11:         end while
12:         $L_c = \sum_k \{A^*_k - \sum_i Z_{k,c,i}\} \cdot F_{k,c,i}$
13:     end for
14:     Integral allocation for component with smallest loss.
15:     $c' = \arg\min_{c \in \mathcal{C}} L_c$
16:     Update $\hat{X}_{k,c',i} \leftarrow Z_{k,c',i}, \forall k, i$
17:     Update $A^*_k = \min\{A^*_k, \sum_i F_{k,c',i} \hat{X}_{k,c',i}\}$, $\mathcal{C} \leftarrow \mathcal{C} \backslash c'$
18: end while
```

Algorithm LSCP1 follows an approach similar to LSDP. It uses the fractional solution from solving the LP relaxation of MCP as the starting point and restores integrality in each component sequentially. However, varying rates across channels ($F_{k,c,i}$) are now taken into account, which requires restoring integrality on a per-channel basis ($\hat{X}_{k,c,i}$) in each component. Also, the integrality restoration algorithm is different (steps 7–11): At each iteration, the session–channel ($k', i'$) pair that provides the maximum flow is jointly chosen and channel $i'$ is allocated to session $k'$ (steps 8 and 9) while taking into account the maximum flow limit for session $k'$ (determined by the flow returned after integral allocation from previous component—steps 6 and 17).

LSCP1 employs the same procedure (as in LSDP) of using the updated session flow (after integral allocation) from one component as the limiting flow for the session in the next component. Hence, if $\beta$ represents the performance guarantee of the single component problem, then the net guarantee reduces to $\beta^C$. We will now try to characterize $\beta$.

*Lemma 2:* Given the limiting flow for each session as input, the loss due to integrality restoration in each component can be bounded by half.

*Proof:* LSCP1 greedily allocates channels to sessions while ensuring that the maximum flow limit for a session is not violated in each component. We will show that this greedy algorithm maximizes a nondecreasing submodular function over a partition matroid. The suboptimality of such an approach has been shown to be bounded by $\frac{1}{2}$[30].

Consider the ground set to be session–channel pairs: $\Psi = \{(k, i) : k \in [1, K], i \in [1, N]\}$. Now, $\Psi$ can be partitioned into $\phi_i = \{(k, i) : k \in [1, K]\}, \forall i$. A partition matroid ($S$) can now be defined on $\Psi$ as a set of subsets of $\Psi$ such that for all subsets $P \in S$, we have: 1) if $Q \subseteq P$, then $Q \in S$; 2) if element $p \in P \backslash Q$, then $Q \cup \{p\} \in S$; and 3) $|P \cap \phi_i| \leq 1, \ \forall i$. This means that $P$ provides a feasible schedule (at most one session for each channel), allowing the partition matroid to capture our scheduling constraint. Our scheduling objective is given as

$$f(P) = \sum_k \mu_k(P),$$

$$\text{where } \mu_k(P) = \min\left\{A^*_k, \sum_{i:(k,i) \in P} F_{k,c,i}\right\}. \qquad (8)$$

It can be seen that if $Q \subseteq P$, then $\mu_k(Q) \leq \mu_k(P)$. Hence, for an element $(k, i)$ such that $A \cup \{(k, i)\}$ forms a valid schedule, it follows that $f(P \cup \{(k, i)\}) - f(P) \leq f(Q \cup \{(k, i)\}) - f(Q)$, resulting from the maximum flow limit for the session. This establishes that the function $f(P)$ is indeed submodular. Hence, our scheduling problem on each component aims to maximize this nondecreasing submodular function over a partition matroid. Furthermore, the greedy approach followed in step 8 of LSCP1 essentially determines

$(k', i') = \arg\max_{(k,i) \in \phi_i} \{f(P \cup \{(k, i)\}) - f(P)\}$. Now, the suboptimality of $\frac{1}{2}$ follows from the result in [30]. ■

The above lemma indicates that at least half of the optimal integral flow is achievable, given a flow limit for each session as input. However, the optimal integral flow itself varies with the input flow limits, which in turn is an output of the LP relaxation. Hence, it becomes necessary to bound the loss directly with respect to the fractional flow resulting from the LP relaxation. This loss can be bounded by the product of half from Lemma 2 and the integrality gap of the LP relaxation. With multiple components, we conjecture the integrality gap to be low (close to one; true for certain types of submodular functions), which bounds the net loss ($\beta$) to be close to half in each component. Given that $C$ is typically a small number ($C = 2$ being the dominant case), this provides a good guarantee ($\beta^2$) even for the harder CP model. Furthermore, the average-case performance is significantly better (illustrated by evaluations in Section VII).

*1) Improved* $\left(1 - \frac{1}{e} - \epsilon\right)$ *Algorithm: LSCP2:* LSCP2 improves the performance guarantee further, by replacing the greedy solution for integrality restoration inside each component (steps 4–16 in LSCP1), with a more sophisticated LP-based scheme that solves a variant of the maximum general assignment problem considered in [31]. Each single component ($c$) problem can be further formulated as

$$\text{IP}c: \quad \text{Maximize} \quad \sum_{k,s} F^s_{k,c} X_{k,c,s}$$

$$\text{subject to} \quad \sum_{k, s \in S_k : i \in s} X_{k,c,s} \leq 1 \qquad \forall i \in [1, N]$$

$$\sum_{s \in S_k} X_{k,c,s} \leq 1 \quad \forall k; \qquad X_{k,c,s} \in \{0, 1\}$$

where $F^s_{k,c} = \sum_{i\in s} F^s_{k,c,i}$. Here, channel allocation to a session is made in subsets of channels ($X_{k,c,s}$), where each subset $s \in S_k$ indicates a feasible set of channels that can be assigned to session $K$. Feasibility here refers to allocation with at most one subset per session (third constraint) and one session per channel (second constraint), along with the maximum flow constraint, where $\sum_{i\in s} F_{k,c,i} \leq A^*_k$. Hence, given a subset, one can modify the flow rates into $F^s_{k,c,i}$ such that $\sum_{i\in s} F^s_{k,c,i} = \min\{\sum_{i\in s} F_{k,c,i}, A^*_k\}$. Since $\mathrm{IP}c$ is a hard integer program, we solve its LP-relaxation ($\mathrm{LP}c$). However, there are an exponential number of variables due to $S_k$, which requires iterative primal-dual or Lagrangian-based LP techniques (see [31] for details), but can be solved to within $(1-\epsilon)$ of the optimum.

**Algorithm 4:** Multicast Scheduler under CP: LSCP2

```
1: Solve LP relaxation of MCP with output $X^*_{k,c,i}$, $A^*_k$.
2: for $c \in [1, C]$ do
3:     Formulate $\mathrm{IP}c$; solve its LP relaxation ($\mathrm{LP}c$) with output $X^*_{k,c,s}$
4:     Round $X_{k,c,s} = 1$ with probability $X^*_{k,c,s_k}$; $X_{k,c,s} = 0, \forall s \neq s_k, \forall k$
5:     Assign channel $i$ to $k' = \arg\max_{k:i\in s_k}\{F_{k,c,s_k}\}$; remove $i$ from $s_k \neq s_{k'}$; $\forall i \in [1, N]$
6:     Update $A^*_k = \min\{A^*_k, \sum_{i\in s_k} F^{s_k}_{k,c,i}\}$, $\forall k$
7: end for
```

However, the resulting LP relaxation solution in each component (step 3) may assign multiple subsets (fractionally) to a session, and a channel may be assigned to multiple sessions. This is addressed by first rounding the subset assignment variables for each session such that only a single subset $s_k$ is assigned to $k$, where $s_k = s$ occurs with probability $X^*_{k,c,s}$ (step 4). A channel ($i$) may still be assigned to multiple sessions, in which case, the channel is assigned to the session delivering the highest flow ($\max_k\{F^{s_k}_{k,c,i}\}$) and removed from the other sessions (step 5). This results in a feasible multicast schedule with integral channel allocations to sessions. Note that the randomized rounding procedure can be made deterministic by derandomizing using the method of conditional probabilities.

*Theorem 3:* LSCP2 has an approximation guarantee of $(1 - \frac{1}{e} - \epsilon)^C$.

*Proof:* The loss due to sequential flow updates across components has been shown to be bounded by $\beta^C$, where $\beta$ is the loss due to integral allocation in each component. Now, it can be shown that $\beta \geq 1 - \frac{1}{e} - \epsilon$.

Without loss of generality, let the ordering of (session, subset) with respect to channel $i$'s contribution ($F^s_{k,c,i}$) in decreasing value be $(k_1, s_1), (k_2, s_2), \ldots$. Now, the expected contribution of channel $i$ to session $k$ due to our rounding procedure can be given as $\Pi_{k'<k}(1 - X^*_{k',c,s_{k'}})X^*_{k,c,s_k}F^{s_k}_{k,c,i}$. Thus, the net expected contribution of channel $i$ in our integral solution is $\sum_{k:i\in s_k}\Pi_{k'<k}(1 - X^*_{k',c,s_{k'}})X^*_{k,c,s_k}F^{s_k}_{k,c,i}$, while that in the original LP relaxation solution is $\sum_{k,s:i\in s} X^*_{k,c,s}F^s_{k,c,i}$. Reference [31] uses arithmetic-geometric inequality to show that

$$\sum_{k:i\in s_k}\Pi_{k'<k}(1 - X^*_{k',c,s_{k'}})X^*_{k,c,s_k}F^{s_k}_{k,c,i} \geq \left(1 - \frac{1}{e}\right)\sum_{k,s:i\in s} X^*_{k,c,s}F^s_{k,c,i}.$$

Summing over the contribution of all $N$ channels and applying the $(1-\epsilon)$ suboptimality of the LP relaxation ($\mathrm{LP}c$) itself, we get $\beta \geq (1 - \frac{1}{e})(1-\epsilon) \geq (1 - \frac{1}{e} - \epsilon)$. ■

### *B. Greedy Algorithm: GSCP*

**Algorithm 5:** Greedy Scheduler under CP: GSCP

```
1: $A_{k,c,i} = 0, E_{k,c} = 0, \forall k, c, i; valid\_ses = 1$, $\mathcal{K} = \{1, \ldots, K\}$
2: Available channels, $\mathcal{I}_c = \{1, \cdots, N\}$ with $M_c = N$, $\forall c$
3: while $valid\_ses == 1$ do
4:     for $k \in [1, K]$ do
5:         $U_{k,c} = E_{k,c} + \max_{i\in\mathcal{I}_c} F_{k,c,i}$ $\forall c$
6:         $S_{k,c} = [\min_c\{U_{k,c}\} - E_{k,c}]^+$, $\forall c$, where $[x]^+ = \max\{x, 0\}$
7:         if $\Pi_c(M_c + E_{k,c}) == 0$ then $\mathcal{K} \leftarrow \mathcal{K}\backslash k$ end
8:     end for
9:     if $\mathcal{K} \neq \emptyset$ then
10:        $k' = \arg\max_{k\in\mathcal{K}} \frac{\min_c\{\sum_{i\in\mathcal{I}_c} F_{k,c,i}\}}{\sum_c M_c}$
11:        $A_{k',c,i'} = 1$, if $S_{k',c} > 0$, $\forall c$, where $i' = \arg\max_{i\in\mathcal{I}_c} F_{k',c,i}$
12:        $\mathcal{I}_c \leftarrow \mathcal{I}_c\backslash i'$, $M_c = N - \sum_{k,i} A_{k,c,i}$ $\forall c$
13:        $E_{k',c} = \sum_i A_{k',c,i}F_{k',c,i} - \min_c\{\sum_i A_{k',c,i}F_{k',c,i}\}, \forall c$
14:    else
15:        $valid\_ses = 0$
16:    end if
17: end while
```

We also provide a low-complexity, fast greedy algorithm (GSCP) for the CP model. GSCP is along the lines of its DP counterpart GSDP, although with two notable differences.

1) Each session experiences varying rates and, hence, varying flow across channels within each component ($F_{k,c,i}$). This makes the *component with the bottleneck flow* ($\min_c U_{k,c}$, step 5) *for a session vary from one iteration to another* depending on the remaining set of unallocated channels in the components. Hence, the excess flow available as well as the remaining flow needed in each component (to allocate a unit of the bottleneck flow to the session) is kept absolute and not normalized with respect to the bottleneck flow (steps 6 and 13) unlike in GSDP.
2) The session yielding the largest flow per unit channel for the set of remaining unallocated channels in the components is chosen in each iteration for channel allocation. Here, if a component does not have enough excess rate to accommodate a unit of the bottleneck flow for the session, the channel having the highest gain among the unallocated channels in that component is assigned to the ses- complexity of $O(KN^2C^2)$. While LSCP2 is of theoretical interest, we believe that LSCP1 and GSCP are of practical significance.

## VI. PRACTICAL CONSIDERATIONS

The multicast strategy **JRC** requires us to solve the general $C$ component problem. However, we must point out that $C = 2$ carries a lot of practical importance in the relay standard, mostly owing to its easier realization, where all relays on the access hop cooperate (strategy **C**) within a single component. For $C = 2$, our multicast scheduling algorithms provide good guarantees of $(1 - \frac{1}{N})$ for the DP model, and $(1 - \frac{1}{e} - \epsilon)^2 \approx 0.4$ for the CP model. Similarly, for $C = R + 1$ components, the algorithms and their corresponding guarantees can be used for the pure reuse (**R**) strategy. Thus, solving the generic $C$ component problem helps us obtain efficient scheduling algorithms for both cooperation and reuse strategies, either in isolation or combination.

We have considered backlogged buffers in our formulations. However, the LP formulations easily extend to incorporate finite data buffers for sessions by the addition of $K$ flow constraints (max flow limited to buffer size), while the algorithms and their guarantees would continue to apply.

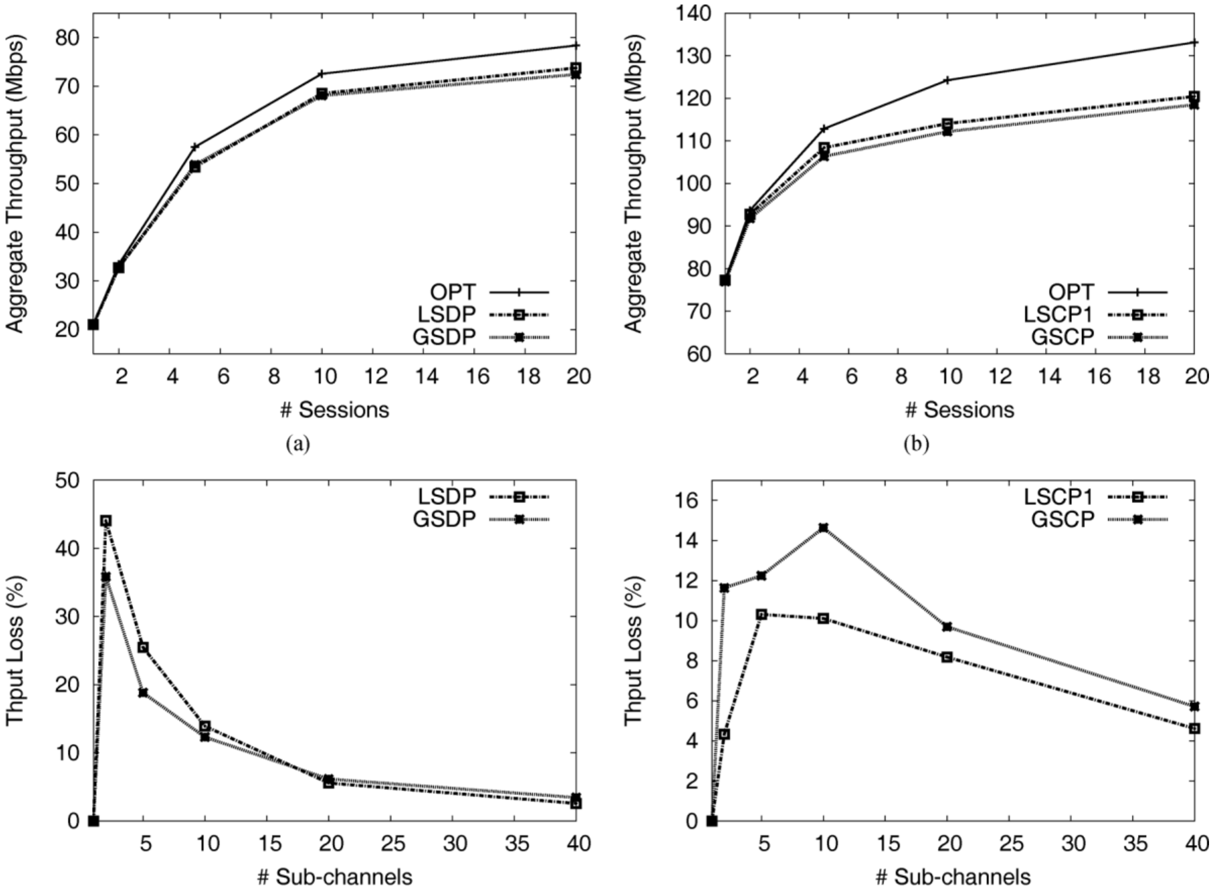
The conventional relay unicast scheduling problem can be captured as a special case of the multicast problem with $C = 2$, where there is no channel reuse across relays in the access hop. This gives us efficient unicast scheduling algorithms as well with guarantees of $\left(1 - \frac{1}{N}\right)$ for the DP model, and 0.4 for the CP model.

## VII. Performance Evaluation

An event-driven packet-level network simulator written in C++ coupled with the GNU LP kit is considered for evaluation of the proposed solutions. A single-cell relay-enabled OFDMA downlink system is considered, with a cell radius of 600 m. MS are uniformly distributed within the cell, while RS are distributed uniformly within a region of $250 \text{ m} \leq r \leq 350 \text{ m}$ from the BS. The relay channel model proposed for the 802.16m standard [1] is considered and incorporates path loss, log-normal shadowing, and Rayleigh fading. Specifically, for the BS-RS links, we use the Type-D line-of-sight path-loss model that is recommended for the above-rooftop-to-above-rooftop urban links, while for the BS-MS and RS-MS links, we use the

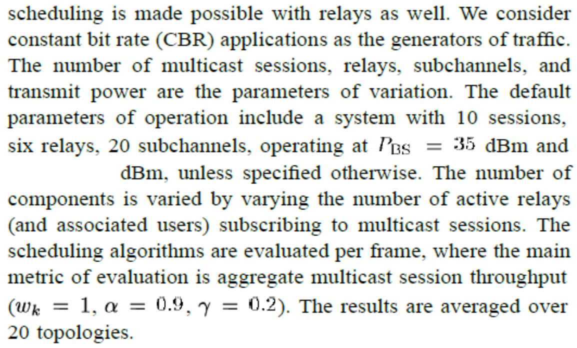
sion (steps 10 and 11). It can be seen that GSCP has a time

Type-E non-line-of-sight path-loss model that is recommended for the above-rooftop-to-below-rooftop urban links. A standard deviation of 3.4 and 8 dB for log-normal shadowing is applied for the BS-RS and BS/RS-MS links, respectively. Each user's Rayleigh channel has a Doppler fading equivalent to a velocity of 3–10 km/h. In addition, based on the multicast strategy, interference and/or cooperation from relays operating on the same channel also contribute to link rates. The feedback of such link rates from the MS (through RS) and RS is assumed to be made available to the BS through standard feedback procedures in DP and CP modes [1]. Note that all works on channel-dependent scheduling per frame rely on such rate feedback and a coherent channel at the frame granularity.



(a) (b) (c) (d)

Fig. 3. Performance of multicast scheduling algorithms. (a) Impact of sessions (DP). (b) Impact of sessions (CP). (c) Impact of channels (DP). (d) Impact of channels (CP).



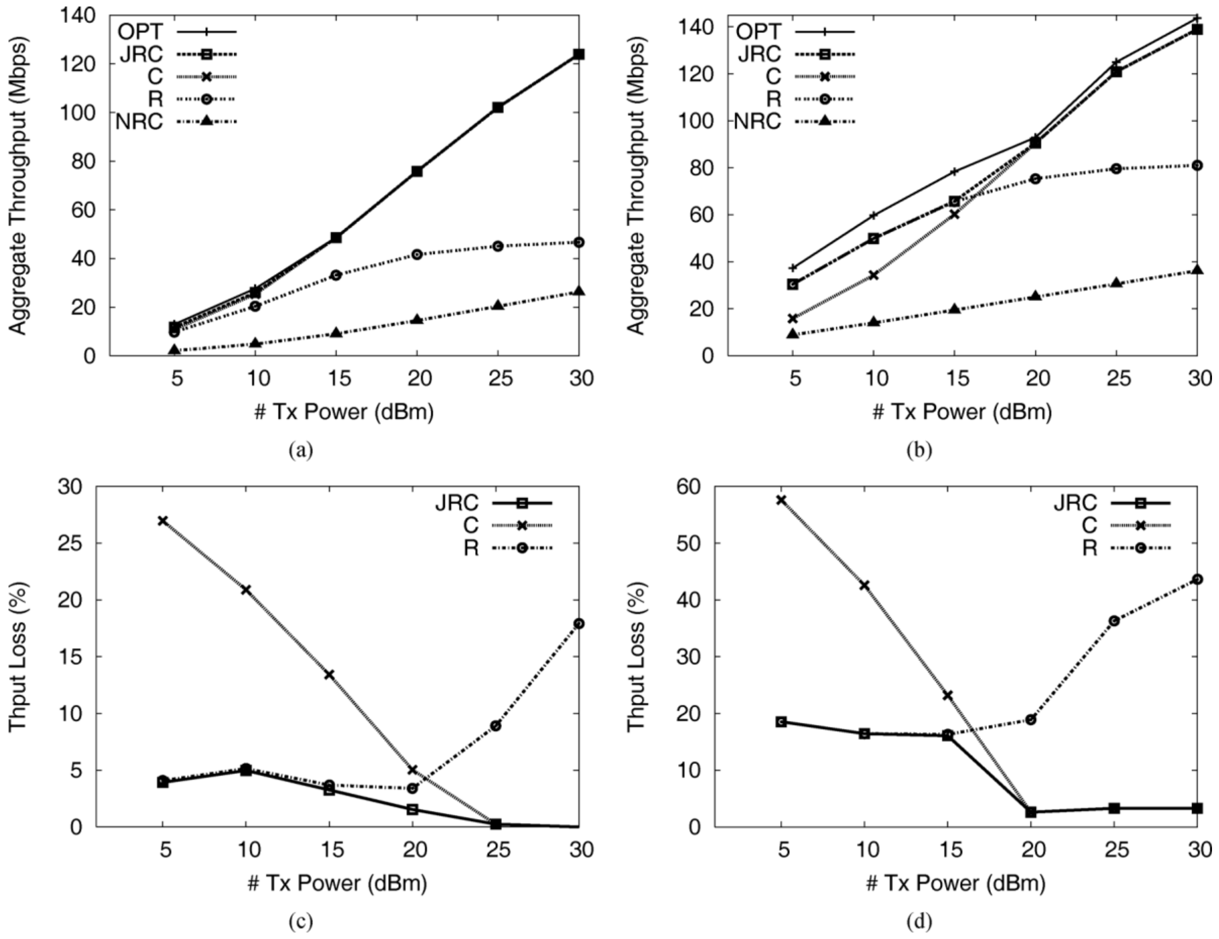
scheduling is made possible with relays as well. We consider constant bit rate (CBR) applications as the generators of traffic. The number of multicast sessions, relays, subchannels, and transmit power are the parameters of variation. The default parameters of operation include a system with 10 sessions, six relays, 20 subchannels, operating at $P_{\rm BS} = 35$ dBm and dBm, unless specified otherwise. The number of components is varied by varying the number of active relays (and associated users) subscribing to multicast sessions. The scheduling algorithms are evaluated per frame, where the main metric of evaluation is aggregate multicast session throughput ($w_k = 1$, $\alpha = 0.9$, $\gamma = 0.2$). The results are averaged over 20 topologies.

### *A. Evaluation of Scheduling Algorithms*

We first evaluate the efficiency of the LP-based and greedy algorithms in JRC by comparing them to the optimal fractional solution (upper bound, OPT) returned by the LP relaxations of the corresponding integer programs. The topologies are gener-ated by selecting three out of six relays randomly to be active and subscribing their associated MS to multicast sessions. De-pending on the distribution of the active relays, the number of components in the topology varies from two to four.

Impact of Sessions: Fig. 3(a) and (b) presents the throughput results for the DP and CP models, respectively, for increasing number of sessions. It can be that both LSDP and LSCP1 algo-rithms perform within 15% of their optimal values, providing a much better average-case performance than their worst-case guarantee. Furthermore, their low complexity, greedy counter-parts (GSDP, GSCP) also perform very close to that of their respective LP-based algorithms, thereby indicating their effec- tiveness in practical scenarios. Note that OPT only serves as a loose upper bound for benchmarking the performance of our algorithms. In reality, the actual optimal solution would be lesser than this upper bound, resulting in a much smaller performance loss for our algorithms. Increasing the number of sessions pro- vides room for larger session multiplexing gain, resulting in higher aggregate multicast throughput. However, as the number of sessions increases, the ability to push more flow into the network through fractional (infeasible) allocations (OPT) in-creases, and this causes the performance of our algorithms to diverge a little from the upper bound (although the gap is less than 15%).

Impact of Channels: Fig. 3(c) and (d) present the throughput loss (from optimal) results for DP and CP models, respectively, with increasing number of OFDMA subchannels. For CP, the loss in optimality is less than 15% in Fig. 3(d). In the presence of channel diversity in CP, it is important to carefully assign channels to users. The suboptimality of wrong decisions, how-ever, gets amortized when the number of channels is large as observed in Fig. 3(d). While channel diversity is the key for



(a) (b) (c) (d)

Fig. 4. Performance of joint multicast strategy. (a) DP: six relays. (b) CP: six relays. (c) DP: three relays. (d) CP: three relays.

better performance in CP, the lack of it in DP places all the performance burden on how well the available channels are uti-lized. We hadshownin SectionIV thatLSDPhasa performance guarantee that gets better with increasing number of channels. This can be observed in Fig. 3(c), where we stress-test LSDP by considering only topologies with four components. The cor-responding performances of LSDP and GSDP indicate that the loss in optimality does decrease significantly to less than 15% even with 10 subchannels. Furthermore, the peak in the result arises because of the starting point (on -axis) being one sub-channel, for which the problem is not hard and can hence be solved optimally (zero throughput loss).

### *B. Evaluation of Joint Multicast Strategy*

We compare the performance of our JRC strategy against individual cooperation (C) and reuse (R) strategies. Note that all these three strategies use our proposed LSDP (LSCP1) al-gorithm for the DP (CP) model. We also consider the baseline strategy that does not allow for cooperation or reuse (NRC) be-tween relays on the access hop, and the fractional LP-relaxation solution that returns the best of Cand R (OPT).Giventhelackof commercial relay deployments yet, we have varied parameters like transmit power of relays and number of relays (with typical values from 802.16m standard [1]) to create different scenarios and understand the relative importance of reuse and cooperation strategies.

Fig. 4(a) and (b) presents the throughput results as a function of transmit power of the relays. All six relays are chosen in the topologies. With the activation of all relays, the signal power reaching the users situated in the boundary between two adjacent relays is comparable to interference power, making the signal-to-interference-plus-noise ratio (SINR) low at all subchannels for DP. This reduces the number of components, thereby making session multiplexing gain insufficient to out-weigh cooperation at all transmit powers for DP [Fig. 4(a)]. With CP, however, the situation is different in Fig. 4(b), where reuse strategy outperforms cooperation at low to moderate transmit powers, while cooperation outperforms only at higher transmit powers. This can be attributed to the higher-session multiplexing gain available from channel diversity with CP even for smaller number of components. In both models, we find that JRC, using a combination of cooperation and reuse, follows the best strategy at all transmit powers, providing a gain of 50%–200% over individual strategies. It also performs very close to the LP bound and provides a large gain of several folds over the baseline strategy.

Fig. 4(c) and (d) presents the throughput loss results when three out of the six relays are randomly chosen to be active, resulting in some topologies with larger number of components (maximum 4). With potentially higher number of components, we find that session multiplexing gain outweighs cooperation gain at low to moderate transmit powers for both DP and CP models, while cooperation dominates at higher transmit powers. JRC helps reduce throughput loss from the LP bound by 20%–50% over individual strategies. Furthermore, the loss is kept small and decreases with increasing power, where the number of components in the system correspondingly decreases.

In summary, depending on various parameters (number of ac-tive relays, transmit power, number of components, DP versus CP modes, etc.), the relative importance of reuse versus coop-eration strategies varies. This emphasizes the need for a joint reuse and cooperation scheme like JRC that automatically tries to adopt the strategy (or a combination of strategies) that best serves the current network condition.

## VIII. CONCLUSION

We considered the problem of multicast scheduling in two-hop OFDMA relay networks. We showed that intelligent grouping of relays for cooperation is needed to address the tradeoff between cooperation and session multiplexing gains. We designed efficient scheduling algorithms (with performance guarantees) at the core of the multicast strategy to address the tradeoff and maximize aggregate multicast flow. Design of network coding mechanisms for multicast retransmissions and its joint incorporation with OFDMA scheduling deserves independent attention and forms an interesting avenue for further research.

## REFERENCES

[1] IEEE 802.16m 2011 Part 16, Air Interface for Broadband Wireless Ac-cess Systems—Advanced Air Interface, IEEE 802.16m, May 2011.

[2] 3GPP, ―Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description,‖ TS 36.300, Sep. 2012.

[3] Z.Zhang,Y.He,andK.P.Chong,― Opportunisticdownlinkscheduling for multiuser OFDM systems,‖ in Proc. IEEE WCNC, Mar. 2005, vol. 2, pp. 1206–1212.

[4] G. Song and Y. Li, ―Cross-layer optimization for OFDM wireless networks—Part I: Theoretical framework,‖ IEEE Trans. Wireless Commun., vol. 4, no. 2, pp. 614–624, Mar. 2005.

[5] A.HottinenandT.Heikkinen,― SubchannelassignmentinOFDMrelay nodes,‖ in Proc. CISS, Mar. 2006, pp. 1314–1317.

[6] K. Sundaresan and S. Rangarajan, ―On exploiting diversity and spatial reuse in relay-enabled wireless networks,‖ in Proc. ACM MobiHoc, May 2008, pp. 13–22.

[7] S. Deb, V. Mhatre, and V. Ramaiyan, ―WiMAX relay networks: Op-portunistic scheduling to exploit multiuser diversity and frequency se-lectivity,‖ in Proc. ACM MobiCom, Sep. 2008, pp. 163–174.

[8] 3GPP TSG E-UTRAN, ―General aspects and principles for interfaces supporting multimedia broadcast multicast service (MBMS) within E-UTRAN,‖ Release 11, vol. TS 36.440, 2012.

[9] ―Multihop Relay Specification,‖ in Amendment to IEEE Std. for LAN/ MAN—Part 16: Air Interface for Fixed and Mobile Broadband Wire-less Access Systems, P802.16j, Mar. 2006.

[10] S. Mengesha and H. Karl, ―Relay routing and scheduling for capacity improvement in cellular WLANs,‖ in Proc. WiOpt, Mar. 2003.

[11] N. Challa and H. Cam, ―Cost-aware downlink scheduling of shared channels for cellular networks with relays,‖ in Proc. IEEE Int. Conf. Perform., Comput., Commun., 2004, pp. 793–798.

[12] H. Viswanathan and S. Mukherjee, ―Performance of cellular net-works with relays and centralized scheduling,‖ IEEE Trans. Wireless Commun., vol. 4, no. 5, pp. 2318–2328, Sep. 2005.

[13] M. Herdin, ―A chunk based OFDM amplify-and-forward relaying scheme for 4G mobile radio systems,‖ in Proc. IEEE ICC, Jun. 2006, vol. 10, pp. 4507–4512.

[14] A. So and B. Liang, ―Effect of relaying on capacity improvement in wireless local area networks,‖ in Proc. IEEE WCNC, Mar. 2005, vol. 3, pp. 1539–1544.

[15] M. Andrews and L. Zhang, ―Scheduling algorithms for multi-carrier wireless data systems,‖ in Proc. ACM MobiCom, Sep. 2007, pp. 3–14.

[16] K. Sundaresan and S. Rangarajan, ―Efficient algorithms for leveraging spatial reuse in OFDMA relay networks,‖ in Proc. IEEE INFOCOM, Apr. 2009, pp. 1539–1547.

[17] H.Won,H.Cai,D.Y.Yun,K.Guo,andA. Netravali,―Multicastsched-ulingincellularpacketdatanetworks,‖inProc.IEEEINFOCOM,Apr. 2007, pp. 1172–1180.

[18] S. Deb, S. Jaiswal, and K. Nagaraj, ―Real-time video multicast in WiMAX networks,‖ in Proc. IEEE INFOCOM, Apr. 2008, pp. 1579–1587.

[19] J. Du, M. Xiao, and M. Skoglund, ―Capacity bounds for relay-aided wireless multiple multicast with backhaul,‖ in Proc. WCSP, Oct. 2010, pp. 1–5.

[20] D. Gunduz, O. Simeone, A. J. Goldsmith, H. V. Poor, and S. Shamai, ―Multiple multicasts with the help of a relay,‖ IEEE Trans. Inf. Theory, vol. 56, no. 12, pp. 6142–6158, Dec. 2010.

[21] P. Fan, C. Zhi, C. Wei, and K. B. Letaief, ―Reliable relay assisted wire-less multicast using network coding,‖ IEEE J. Sel. Areas Commun., vol. 27, no. 5, pp. 749–762, Jun. 2009.

[22] J. Du, M. Xiao, and M. Skoglund, ―Cooperative network coding strate-gies for wireless relay networks with backhaul,‖ IEEE Trans. Wireless Commun., vol. 59, no. 9, pp. 2502–2514, Sep. 2011.