Joint Relaying and Spatial Sharing Multicast Scheduling for mmWave Networks

Gek Hong (Allyson) Sim*, Mahdi Mousavi†, Lin Wang‡, Anja Klein†, Matthias Hollick*

* Secure Mobile Networking Lab (SEEMOO), Technische Universität Darmstadt

{asim, mhollick}@seemoo.tu-darmstadt.de

† Communication Engineering Lab, Technische Universität Darmstadt

{m.mousavi, a.klein}@nt.tu-darmstadt.de

‡VU Amsterdam, The Netherlands

lin.wang@vu.nl

Abstract—**Millimeter-wave (mmWave) communication plays a vital role to efficiently disseminate large volumes of data in beyond-5G networks. Unfortunately, the directionality of mmWave communication significantly complicates efficient data dissemination, particularly in multicasting, which is gaining more and more importance in emerging applications (e.g., V2X, public safety). While multicasting for systems operating at lower frequencies (i.e., sub-6GHz) has been extensively studied, they are sub-optimal for mmWave systems as mmWave has significantly different propagation characteristics, i.e., using the directional transmission to compensate for the high path loss and thus promoting spectrum sharing. In this paper, we propose novel multicast scheduling algorithms by jointly exploiting relaying and spatial sharing gains while aiming to minimize the multicast completion time. We first characterize the min-time mmWave multicasting problem with a comprehensive model and formulate it with an integer linear program (ILP). We further design a practical and scalable distributed algorithm named `mmDiMu`, based on gradually maximizing the transmission throughput over time. Finally, we carry out validation through extensive simulations in different scales and the results show that `mmDiMu` significantly outperforms conventional algorithms with around 95% reduction on multicast completion time.**

Index Terms—**Millimeter-wave (mmWave) networks, multicasting, relay, spatial sharing, scheduling.**

I. Introduction

Millimeter-wave (mmWave) communication fulfills the demand for multi-gigabit-per-second (Gbps) throughput and low-latency communication even for extremely dense networks [1], which are usually not easy to sustain with traditional communications operating at sub-6GHz frequencies. Despite its benefits, mmWave communication suffers from very high attenuation, resulting in dramatic penetration loss, due to its high frequency. To compensate for this loss, directional transmissions are typically employed, where the coverage of communication is constrained to a rather small area, e.g., to the line of sight in the extreme case. This limitation poses new challenges in particular to guarantee efficient content dissemination for various delay-sensitive multicast applications (e.g., raw sensory data broadcasting in vehicle-to-everything (V2X) communications to support autonomous driving, high-definition video broadcasting in a concert hall, and public-safety use cases).

Although multicast scheduling has been widely explored for networks operating at sub-6GHz frequencies, the specific benefits and challenges of mmWave multicast scheduling remain understudied [2]. In particular, multicast scheduler designs for sub-6GHz communications assume the availability of omnidirectional transmission, and thus a source node can schedule the transmission to any arbitrary subset of receiving nodes within a certain range simultaneously. However, the restricted coverage of mmWave communication undermines this assumption and renders these designs inapplicable, opening a new research question.

One trivial design for mmWave multicast scheduling can simply employ multiple directional unicast and/or multicast transmissions to sequentially serve all multicast nodes. The behavior of such a scheduler is illustrated in Fig. 1a, where the source node (labeled as ⓪) transmits sequentially in sectors 1 to 5 to serve multicast nodes ①, ②, ③ ④, ⑤, and ⑥, respectively. One can easily observe that this trivial design is extremely inefficient and a straightforward improvement can be applied if we consider beam grouping based on adaptive beamforming [2]–[4]. As shown in Fig. 1b, nodes that are closer to the source nodes (i.e., ①, ②, and ③) are served together with a wider beam, while the father nodes (i.e., ④, and ⑤) and the nodes that are not in proximity with the other nodes (i.e., ⑥) with narrower beams. Although adaptive method provides higher flexibility in grouping the receiving nodes, it however comes at the expense of more complex beamforming and costly antenna architecture.

The above designs rely only on single-hop transmissions, which can be problematic in many practical scenarios. More specifically, there might exist nodes that are not reachable by the source or nodes that are not feasible for high transmission rates due to large distance (i.e., node ⑤ in sector 4) or the presence of blockages [5], [6]. In such cases, a relay-aided transmission is inevitable to ensure reachability and guarantee high-performance multicasting (in terms of throughput and delay). With relay enabled, a node can serve as a transmitting node as soon as it receives the data from another node. As shown in Fig. 1c, upon receiving data from node ⓪ in the first time slot, nodes ③, and ④ act as the relay node for node ① ②, and ⑤, respectively. With this flexibility, we can break down a
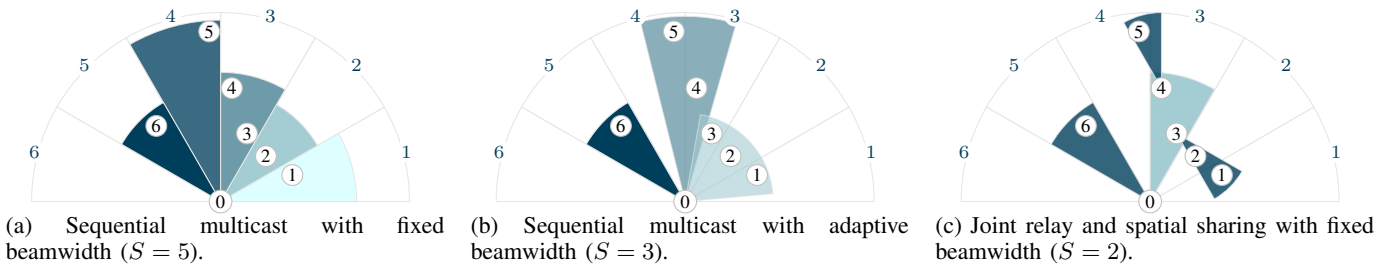
(a) Sequential multicast with fixed beamwidth ($S = 5$).

(b) Sequential multicast with adaptive beamwidth ($S = 3$).

(c) Joint relay and spatial sharing with fixed beamwidth ($S = 2$).

Fig. 1: Different multicast mechanisms with different gray shades represent the transmission/multicast sessions in different time slots, and S indicates the total slots. The number at each sector's arc indicate the index of each sector.

low-rate multicast transmission into a combination of multiple high-rate unicast and/or multicast transmissions that can be scheduled separately. Interestingly, we can then leverage the limited coverage of directional transmissions in mmWave due to the significantly increased spatial gain brought by significantly reduced interference among concurrent (unicast or multicast) transmissions; in Fig. 1c, links ⓪→⑥, ③→① ②, and ④→⑤ occur simultaneously.

We believe the optimal performance of mmWave systems should jointly exploit all these properties of mmWave communication, namely *relaying* and *spatial sharing*. Thus far, the existing works have considered single aspects, but never jointly. This motivates us to design new mmWave multicast scheduling algorithms integrating both relaying and spatial sharing. Unsurprisingly, the joint optimization is complicated and the specific challenge resides in designing efficient communication group composition and spatial sharing scheduling. With both spatial and temporal factors involved, the relay nodes have to be determined gradually and the source and the relay nodes have to select carefully their target nodes depending on how the communication will affect the total completion time. This situation becomes even worse when only limited knowledge about the behavior of the other node with concurrent transmissions is available.

To address these challenges, we provide a comprehensive model and an integer linear program (ILP) to characterize the problem, with the objective of minimizing the multicast completion time (i.e., the time required for all nodes to receive the intended data). The ILP aims to find the optimal scheduling policy that determines the transmitting nodes and their corresponding receivers at each time slot. Specifically, it jointly minimizes the duration of each time slot accounting for all *concurrent transmissions*[1] while selecting the optimal relay node. Exploiting spatial sharing in the relay-aided multicast transmission requires careful scheduling, both spatially and temporally, which is usually not of concern in the conventional multicast. Hence, the problem formulation for directional multicasting is significantly different and inherently more complicated than that of the conventional multicast scheduling in the literature. Ultimately, solving the ILP provides a tight lower bound for the multicast completion time in a mmWave network leveraging both relaying and spatial sharing gains.

To account for the deployment in real-world scenarios in equipment with computational power constraints and to ensure scalability, we further present a lightweight distributed algorithm, namely `mmDiMu`. The high-level idea is to exploit concurrency by allowing each transmitting node to autonomously decide and transmit to its target node(s), regardless of the other concurrent transmissions in the network. The set of target nodes for each transmitting node is determined based on the physical distance of nodes and is updated after every transmission time slot.

The following summarizes the contributions of this paper:

- We identify the challenges and opportunities in mmWave multicast scheduling and provide an ILP formulation that finds the optimal scheduling policy by jointly leveraging relaying and spatial sharing gains.
- Due to the exponential complexity of the ILP-based solution (namely `ILP`), we propose `mmDiMu` heuristic – a *scalable distributed mmWave multicast scheduling algorithm*. This lightweight algorithm has significantly lower complexity, and is more practical than `ILP`.
- We perform extensive simulations to validate the performance of our algorithm in both low- and high-density networks. As expected `ILP` demonstrates a substantial gain in completion time as compared to all other algorithms. While there is a slight gap between `mmDiMu` and `ILP` solutions, we can observe a significant improvement over the existing algorithms, i.e., `FHMOB` in [7], and `OMS` in [8] for sub-6GHz and the adaptive beamwidth algorithm (i.e, `Adapt`) in [2] for mmWave, which to the best of our knowledge represents the state of the art.
- We evaluate interference imposed on unintended receivers by the proposed algorithm and show that the impact of interference is marginal even for high-density scenarios.
- We also provide valuable insights on the design of a mmWave multicast system and design guideline depending on the network's density and system configurations.

The rest of this paper is organized as follows. In Section II, we present the state of the art for multicast scheduling algorithms. Section III includes a description of the system model and its problem formulation. The optimal solution (i.e., based on ILP) is presented in Section IV-A, and Section IV-B presents a lightweight heuristic. The performance evaluation is presented in Section V. In Section VI, we discuss other important aspects to design mmWave multicasting and Section VII concludes our paper.

---

[1] In mmWave communication systems, the terminology of spatial sharing is also commonly referred to as concurrent transmission. In this paper, these terms are used interchangeably.

II. Related Work

As a key technology for beyond-5G networks, mmWave has been considered for many emerging applications (e.g., autonomous driving, public safety, and mobile video streaming) that typically require the distribution of data in large volume with low latency. Unfortunately, directional mmWave links suffer from limited coverage, and it complicates multicasting. Many existing works on mmWave mainly focus on unicast transmissions. With that said, the challenges and benefits of mmWave multicast remain understudied. In this section, we present the state of the art of multicast techniques for both sub-6GHz and mmWave networks, while differentiating them with our proposed approach.

A. Sub-6GHz multicasting

The most basic type of multicasting is broadcast, in which all nodes are served simultaneously. In this case, the transmit rate is limited by the node with the worst channel quality. Improving over this basic technique, many opportunistic multicast techniques are proposed in [9]–[11] and the references therein. These techniques exploit multiuser diversity by opportunistically transmitting to an arbitrary subset of the nodes with better instantaneous channel quality. As a result, they outperform the broadcast scheme and achieve higher throughput. However, this technique still suffers from poor performance when the network has nodes located at its edge. In the extreme case (i.e., when many nodes are located at the edge), it performs similarly to a broadcast scheme.

Overcoming the above issue, the research community has explored multicast beamforming. Multicast beamforming uses the beamforming technique that focuses the transmit signal power at only one direction of interest by adjusting the antenna gains. As a result, it improves the signal-to-noise ratio (SNR) of the nodes in that direction. Authors in [12] publish one of the first work on improving the system throughput with this technique. They first use omnidirectional multicast to transmit to nodes with better channel quality and then use directional multicast to transmit sequentially to the remaining nodes. To further improve the system performance, a better method applies beamforming weights at the antenna leading to the maximization of the worst SNR, at the expense of degrading the SNR of other nodes (i.e., the nodes that are located closer to the transmitter). Many research works demonstrate this technique yields a high system throughput [8], [13], [14] and minimizes completion time [7], [15], [16].

The aforementioned works mainly focus on scheduling the subset of nodes in a system to achieve the intended goal, where neither coverage nor blockage is an issue. Specifically, a source node can simultaneously transmit to any arbitrary subset or even all nodes if desired. Nevertheless, operating at high frequency, mmWave communications are prone to extremely high attenuation and penetration loss. Furthermore, the use of directional transmission (which only covers a small angular area) makes it impossible to serve any arbitrary nodes in the system simultaneously. As a result, the multicasting techniques

designed for sub-6GHz communication yield suboptimal performance for mmWave communication. To shed light on this aspect, we specifically benchmarked the performance of our proposed algorithms to two seminal multicast schedulers used in sub-6GHz systems (i.e., in [7], [8]) in Section V.

B. mmWave multicasting

An initial work addressing the need for the redesign of mmWave multicast scheduling is presented in [3] where the authors emphasize on the use of adaptive beamwidth to improve the grouping of the multicast nodes to achieve higher throughput. Similar work is presented in [2] where the authors investigate the trade-off between transmission beamwidth and achievable SNR to ensure high throughput. These schedulers may require a high level of beamwidth adaptation to form arbitrary beams to provide coverage to the multicast nodes. Therefore, this design increases the complexity and the cost of the antenna design. In contrast, with a highly reduced complexity, the authors in [17] present a practical IEEE 802.11ad compliance approach where a codebook-based scheduler with one radio frequency (RF) chain is applied.

All above-mentioned works consider only single-hop multicasting in which the multicast transmission rate remains limited to the nodes located farthest from the source node without leveraging spatial sharing. Later, the benefits of relay and spatial sharing are separately considered in [18] and [19] to improve the multicast rate and spectral efficiency, respectively. In [18], the authors exploit relaying only to overcome non-line-of-sight paths, but not for performance optimization. In [19], the authors leverage spatial sharing in which they enable the simultaneous transmission of single-hop unicast and multicast sessions to increase network efficiency.

To sum up, all the works mentioned above works either consider *multi-hop relay* or *optimal spatial sharing*, but not jointly. *To the best of our knowledge, we are the first to jointly consider both to minimize the data delivery time for mmWave multicast communications.*

III. System Model and Assumptions

We consider a mmWave network composed of $N + 1$ randomly distributed nodes denoted by set $\mathcal{N} = \{0, 1, ..., N\}$, where node 0 represents the source and the other nodes $n = 1, ..., N$ are interested in receiving data of size B from the source. We assume relaying is enabled in the network, meaning that all the nodes, once receiving the data, can transmit the data to other nodes. We consider a time slotted system where the number of time slots for multicasting the data is denoted by variable S, and the set of time slots is given by $\mathcal{S} = \{1, ..., S\}$. The length of each time slot is not necessarily equal, but we ensure that transmissions happen only within one-hop at each time slot. To exploit spatial sharing, multiple concurrent transmissions can coexist at each time slot.

We call a node that transmits data to other node(s) a parent node (PN), and we denote by $\mathcal{P}^s \subset \mathcal{N}$ the set of PNs at time slot s. Inversely, a node that receives data is called a child node (CN), and we denote by $\mathcal{C}_m^s \subset \mathcal{N}$ the set of CNs of PN m at
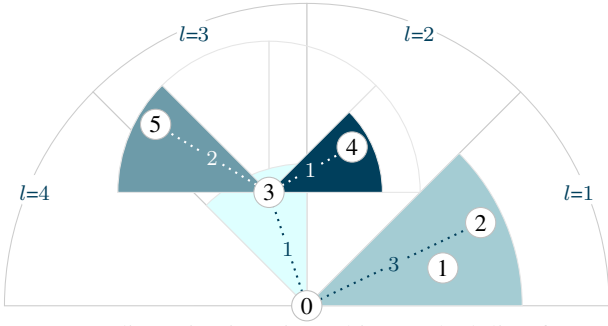
Fig. 2: Data dissemination via multicast scheduling for $L = 8$. The number on the edges indicates the number of required transmission slots.

time slot s. A node can serve as PN in multiple time slots, and the data has to be completely delivered to all its CNs in each of these time slots. Therefore, we have $\mathcal{C}_m^{s_1} \neq \mathcal{C}_m^{s_2}$ for any $s_1, s_2 \in \mathcal{S}$ and $s_1 \neq s_2$. Each node in the network has a *fixed* transmit power and L equal-width orthogonal lobes numbered counterclockwise starting from $0°$, denoted by $\mathcal{L} = \{1, ..., L\}$. For each node $m \in \mathcal{N}$, we denote by $\mathcal{N}_m^l$ the set of nodes that are within the coverage of lobe $l \in \mathcal{L}$ of the node. For example, we have $\mathcal{N}_0^1 = \{1, 2\}$ and $\mathcal{N}_3^4 = \{5\}$ in Fig. 2. Note that, as the lobes are orthogonal, a node can activate more than one lobe simultaneously.

We adopt a path-loss model used in [20] (will be detailed in Section V-A), and the received rate is computed using the Shannon capacity model from [21]. We denote by $\gamma_{m,n}$ the SNR of the signal received at CN n, transmitted from PN m. A node is called a target node (TN) of a PN if its received signal has the lowest SNR as compared to the other CNs within the same lobe of the PN. In fact, the nodes with SNR worse than that of the TN are assumed to be unable to decode the message transmitted by the PN. Note that there is at most one TN in each lobe for a PN. We denote by $\mathcal{G}_m^s$ the set of all TNs of PN m at time slot s. Given $\mathcal{C}_m^s$, the set $\mathcal{G}_m^s$ can be formally defined as

$$\mathcal{G}_m^s = \{n \mid \gamma_{m,n} = \min_u \{\gamma_{m,u}\}, u \in \mathcal{N}_m^l \cap \mathcal{C}_m^s, \forall l \in \mathcal{L}\}. \quad (1)$$

Note that $|\mathcal{G}_m^s| = 0$ means that node m does not transmit at time slot s and $|\mathcal{G}_m^s| = L$ means that node n steers its beam towards all directions, where $|\cdot|$ gives the cardinality of a set. Since the TNs experience the worst channel conditions in comparison to other CNs within the same lobe, the maximum rate which determines the transmission time of a PN depends on the SNR of the set of its TNs $\mathcal{G}_m^s \subseteq \mathcal{C}_m^s$. Given the TN set $\mathcal{G}_m^s$ of a PN m, finding the optimal transmitting rate for the PN is as discussed in [7]. Our focus is on obtaining the optimal $\mathcal{C}_m^s$ for each node m at each time slot s. Note that activating more lobes simultaneously results in lower transmission rate. Let $r^*(\mathcal{G}_m^s)$ be the optimal transmit rate. The time required for PN m to complete the data transmissions to all its TNs (including its CNs) at time slot s is given by

$$t_m(\mathcal{G}_m^s) = \frac{B}{r^*(\mathcal{G}_m^s)}. \quad (2)$$

At each time slot, multiple PN can transmit simultaneously, exploiting spatial sharing. As a result, the duration of a time

slot is determined by the longest transmission at the time slot, that is,

$$t^s(\mathcal{N}) = \max_{m \in \mathcal{N}} \{t_m(\mathcal{G}_m^s)\}. \quad (3)$$

Our objective in this work is to minimize the total duration of all the time slots in $\mathcal{S}$, namely multicast *completion time*, by jointly minimizing t^s and S, and to determine the set of PNs and their corresponding CNs in each time slot. The completion time T can be expressed by

$$T(\mathcal{N}) = \sum_{s \in \mathcal{S}} t^s(\mathcal{N}). \quad (4)$$

The following constraints should be considered. First, all nodes have to receive the data within S time slots, i.e.,

$$\bigcup_{m \in \mathcal{N}, s \in \mathcal{S}} \mathcal{C}_m^s = \mathcal{N} \setminus \{0\}. \quad (5)$$

Then, a node can only transmit data to other nodes if it has already received the data, i.e.,

$$\forall s \geq 2, m \in \mathcal{P}^s \implies m \in \bigcup_{\substack{x \in \mathcal{P}^{s'} \\ 1 \leq s' \leq s-1}} \mathcal{C}_x^{s'} \cup \{0\}. \quad (6)$$

IV. Proposed Approaches

In this section, we describe our solutions to the min-time mmWave multicast scheduling problem. We first provide an ILP formulation that gives an optimal schedule, and then we propose a more scalable distributed algorithm.

A. Optimum Solution by ILP

We first define terms and variables using a toy example in Fig. 2. We define K as the number of elements in the power set of $\mathcal{N} \setminus \{0\}$, excluding the empty set, i.e., $K = 2^N - 1$. In Fig. 2, we have $N = 5$, $K = 31$, and $L = 8$.

- $\mathbf{g}_m^s$ (target vector of PN m in time slot s): a binary vector $\mathbf{g}_m^s = [g_{m,1}^s \ldots g_{m,N}^s]^\mathsf{T} \in \{0, 1\}^N$ in which $(.)^\mathsf{T}$ is the transpose operator and $g_{m,n}^s = 1$ if node n is a TN of PN m in time slot s. For example, in Fig. 2, nodes ② and ③ are the TNs of the source in the first time slot, and hence the target vector is $\mathbf{g}_0^1 = [01100]^\mathsf{T}$. There are K possible combinations for a target vector for each PN.

- $\mathbf{U}$ (target matrix): a binary matrix of size $N \times K$. Each of the columns of $\mathbf{U}$ represents a possible choice for a target vector, where $\mathbf{g}_m^s$ is a column of $\mathbf{U}$. In fact, $\mathbf{U}$ is independent of the nodes, and it shows the state-space of the target vector $\mathbf{g}_m^s, m \in \mathcal{N}$. Precisely, $\mathbf{U} = [\mathbf{u}_1^\mathsf{T}, \ldots, \mathbf{u}_K^\mathsf{T}]$ where $\mathbf{u}_k$ is a $1 \times N$ binary vector. We form $\mathbf{U}$ by filling $\mathbf{u}_k$, $1 \leq k \leq K$, via the reverse (rev) of the N-bit decimal-to-binary (dec2bin) conversion of the index k. For instance, $\mathbf{u}_6 = \text{rev}([\text{dec2bin}(6)]) = \text{rev}([00110]) = [01100]$ and in Fig. 2, based on the definition of TN in (7), we have $\mathbf{g}_0^1 = \mathbf{u}_6^\mathsf{T}$.

- $\mathbf{p}_m^s$ (PN vector of PN m in time slot s): a binary vector $\mathbf{p}_m^s = [p_{m,1}^s, \ldots, p_{m,K}^s]^\mathsf{T} \in \{0, 1\}^K, \forall m \in \mathcal{N}$ and $\|\mathbf{p}_m^s\| \leq 1$. If node m is a PN at time slot s, then $\|\mathbf{p}_m^s\| = 1$, otherwise, $\|\mathbf{p}_m^s\| = 0$. Precisely, $p_{m,k}^s = 1$ if PN m chooses the k-th column of $\mathbf{U}$ as its target vector. Given $\mathbf{p}_m^s$, the TNs of PN m is obtained by

$$\mathbf{g}_m^s = \mathbf{U}\mathbf{p}_m^s. \quad (7)$$

- $\mathbf{N}_m$ (observation matrix): $\mathbf{N}_m = [\mathbf{n}_m^1, ..., \mathbf{n}_m^L]$ is a binary matrix of size $N \times L$, defined for every $m \in \mathcal{N}$. For each node m, $\mathbf{N}_m$ indicates with which lobe can node m cover the other nodes using a single-hop transmission. Precisely, $\mathbf{N}_m(n, l) = 1$ if node n is within lobe l of node m. For network in Fig. 2, we have

$$\mathbf{N}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{8}$$

- $\mathbf{C}_m$ (CNs matrix): a binary matrix of size $N \times K$ that shows if a PN m transmits to its TNs, which of the other nodes fall within the coverage area of the PN. While $\mathbf{g}_m^s$ represents the set $\mathcal{G}_m^s$ of TNs of PN m, defined in (7), $\mathbf{C}_m$ corresponds to the set $\mathcal{C}_m^s$ of CNs of PN m, which can also be served given the TNs in $\mathbf{g}_m^s$. Let $\mathbf{g}_m^s$ be the target vector of PN m corresponding to the k-th column of $\mathbf{U}$, then the elements of the k-th column of $\mathbf{C}_m$, which are equal to 1, represent all the nodes which can be served by such a target vector. To clarify, let node $n \in \mathcal{N}_m^l$ be a target node of PN m given $\mathbf{g}_m^s$ corresponding to the k-th column of $\mathbf{U}$. Based on the definition, since node n as the TN of PN m is always in $\mathcal{C}_m^s$, then, $\mathbf{C}_m(n, k) = 1$. Further, we have $\mathbf{C}_m(u, k) = 1$ if $\gamma_{m,u} \geq \gamma_{m,n}$, $\forall u \in \mathcal{N}_m^l$. Based on item (ii), for the source node in Fig. 2, we have $\mathbf{C}_0(:, 6) = [11100]^\mathsf{T}$ which corresponds to $\mathbf{u}_6^\mathsf{T}$. Given the PN vector $\mathbf{p}_m^s$, we denote all the CNs, covered by PN m, by a binary vector $\hat{\mathbf{c}}_m^s = [\hat{c}_{m,1}^s, ..., \hat{c}_{m,N}^s]^\mathsf{T}$ where $\hat{c}_{m,n}^s = 1$ if node n is covered by PN m at time slot s. $\hat{\mathbf{c}}_m^s$ is thus obtained by

$$\hat{\mathbf{c}}_m^s = \mathbf{C}_m \mathbf{p}_m^s. \tag{9}$$

- $\mathbf{t}_m$ (transmission duration): $\mathbf{t}_m = [t_{m,1}, ..., t_{m,K}] \in \mathbb{R}^K$, $m \in \mathcal{N}$, a real-valued vector . If a PN m chooses the k-th column of $\mathbf{U}$ as its target vector $\mathbf{g}_m^s$, then, $t_{m,k}$ shows the duration of transmission defined in (2).

Matrices $\mathbf{U}, \mathbf{N}_m, \mathbf{C}_m, \mathbf{t}_m$ can be calculated given the distribution of nodes in the network, while $\mathbf{p}_m^s, \forall m \in \mathcal{N}, s \in \mathcal{S}$ are to be found by the ILP. Using these terms, the ILP formulation is provided as follows.

$$\min_{p_{m,k}^s} \quad T(\mathcal{N}) = \sum_{s \in \mathcal{S}} \max_{m \in \mathcal{N}} \{\mathbf{t}_m \mathbf{p}_m^s\} \tag{10a}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} p_{m,k}^s = \begin{cases} 1 & m = 0, s = 1 \\ 0 & m = [1, ..., N], s = 1 \\ \leq 1 & s \geq 2 \end{cases} \tag{10b}$$

$$\sum_{k=1}^{K} p_{m,k}^s \leq \sum_{s'=1}^{s-1} \sum_{x \in \mathcal{N}\backslash\{m\}} \hat{c}_{x,m}^{s'}, \forall m \in \mathcal{N}\backslash\{0\}, s \geq 2 \tag{10c}$$

$$(\mathbf{g}_m^s)^\mathsf{T} \mathbf{n}_m^l \leq 1, \qquad \forall m \in \mathcal{N}, \forall s, \forall l \tag{10d}$$

$$p_{m,k}^s \in \{0, 1\}, \qquad \forall m \in \mathcal{N}, \forall s, \forall k \tag{10e}$$

As mentioned, $p_{m,k}^s \in \{0, 1\}$ in (10) is the decision variable, which determines the TNs of PN m as in (7). (10b) expresses that the source node must transmits at $s = 1$, but not the other nodes. In the following time slots, any of the nodes in $\mathcal{N}$ could be a PN given that it has received the data in any previous

time slots $1 \leq s' \leq s - 1$; the constraint in (10c) indicates this. Finally, (10d) guarantees that the number of TN in a lobe is at most one.

Regarding the complexity, ILP formulation is an NP-hard problem as a special case of the problem has been shown to be NP-hard [22]. Although NP-hard, its running time depends on the number of integer variables. Our proposed ILP has $N \times (2^N + 1)$ variables, and thus a complexity of $O(2^N)$, which exponentially increases with N. Clearly, the ILP-based solution has an exponential time complexity, and it can only be solved for very small problem instances (i.e., small N). For this reason, in the next section, we design a practical and lower complexity heuristic.

B. Distributed Multicast Scheduling

Our distributed multicast scheduling heuristic, namely `mmDiMu`, accounts for both relay and spatial sharing. By having each PN deciding autonomously its CNs to transmit to, `mmDiMu` is scalable and distributed in nature as opposed to the centralized ILP solution. The pseudocode of the algorithm is as shown in Algorithm 1. In what follows, we elaborate on the detail of the algorithm.

We use $\mathcal{W}$ to denote the set of waiting nodes that have not received the intended data. Initially, i.e., at the first time slot, node 0 is the only PN in set $\mathcal{P}^1$, and we have $\mathcal{W} = \{1, ..., N\}$. We use $\mathbf{D}$ to denote the distance matrix, where $\mathbf{D}(m, n)$ represents the distance between nodes m and n. At each of the following time slots $s \geq 2$, we select for each node $n \in \mathcal{W}$ the PN m in $\mathcal{P}^s$ with the least distance $\mathbf{D}(m, n)$. In the case where a node is equidistance from two or more PNs, it will randomly select one of the PNs. After this process, for each node $m \in \mathcal{P}^s$ we obtain its CN set $\mathcal{C}_m^s$ at this time slot, and we apply the opportunistic multicast scheduling that maximizes the sum throughput to select the set of nodes from $\mathcal{C}_m^s$ for PN m to transmit to. The intuition lies in maximizing the achievable rate for each transmission session to promote minimum session transmission time, and thus resulting in minimum completion time. Once receiving the data, a node will be removed from the set $\mathcal{W}$ and added to the PN set $\mathcal{P}^{s+1}$. The above process is repeated until all nodes receive the data. In each time slot, the time for each transmission is recorded as $t_m(\mathcal{C}_m^{s*})$, where $\mathcal{C}_m^{s*}$ is the optimal. The multicast completion time thus can be calculated as $\sum_{s \in \mathcal{S}} \max_{m \in \mathcal{N}} \{t_m(\mathcal{C}_m^{s*})\}$.

V. PERFORMANCE EVALUATION

This section evaluates the performance comparisons between the baseline and our proposed multicast algorithms.

A. Simulation Setup

We consider a uniform and randomly distributed nodes within a 200m×200m area with the source node (i.e., PN 0) located at the center. We adopt the mmWave path-loss model in [20], which is written as,

$$\text{PL[dB]} = \alpha + 10\beta \log_{10}(d_{m,n}) + 20 \log_{10}(f_c) + \chi_\sigma, \tag{11}$$

where $d_{m,n}$ is the distance between the PN m and CN n, f_c is the carrier frequency, and χ_σ represents the shadow

Algorithm 1 Pseudocode of `mmDiMu` algorithm

1: Input: N, $\mathbf{D}$
2: Initialize counters: time slot $s \leftarrow 1$
3: Initialize waiting node set: $\mathcal{W} \leftarrow \{1, ..., N\}$
4: Initialize PN sets: $\mathcal{P}^s \leftarrow \{0\}, \forall s$
5: Initialize CN sets: $\mathcal{C}_m^s \leftarrow \emptyset, \forall m, \forall s$
6: **while** $\mathcal{W} \neq \emptyset$ **do**
7:   **foreach** $m \in \mathcal{W}$ **do**
8:     Select PN: $m = \arg\min_{u \in \mathcal{P}} \mathbf{D}(u, n)$
9:     Store CN: $\mathcal{C}_m^s \leftarrow \mathcal{C}_m^s \cup n$
10:   **foreach** $m \in \mathcal{P}^s$ **do**
11:     Select CN set with max-throughput: $\mathcal{C}_m^{s*}$ is served with rate r_m^s
12:     Update waiting set: $\mathcal{W} \leftarrow \mathcal{W} \backslash \mathcal{C}_m^{s*}$
13:     Update PN set: $\mathcal{P}^{s+1} \leftarrow \mathcal{P}^s \cup \mathcal{C}_m^{s*}$
14:     Compute transmission time of PN m: $t_m(\mathcal{C}_m^{s*}) \leftarrow B/r_m^s$
15:   $s \leftarrow s + 1$
16:   Compute slot-time: $t^s = \max_{m \in \mathcal{P}^s} t_m(\mathcal{C}_m^{s*})$
17: **end**
18: Output multicast completion time $T = \sum_s^S t^s$

TABLE I: Channel parameters.

Parameter	Value
Free space path loss (α)	32.4dB
Carrier frequency (f_c)	73GHz
System Bandwidth (W)	1GHz
Transmit power	14.9dBm [23]
Noise figure	4dB@PN, 7dB@CN
Thermal noise	-174dBm/Hz
Path loss exponent (β)	2.0
Standard deviation (σ)	1.9dB
Shannon capacity (ρ)	$\rho = W \times \min\{\log_2\left(1 + 10^{0.1(\text{SNR}-\delta)}\right), \rho_{\max}\}$ maximum spectral efficiency $\rho_{\max} = 4.6$bps/Hz loss factor $\delta = 1.6$dB
Frame size (B)	1Gbits

fading with zero-mean Gaussian random variable and standard deviation σ in dB. The received rate is computed using the Shannon capacity model in [21]. Table I summarizes the parameter values used in the simulator.

B. Benchmarked Algorithms

This subsection highlights the different algorithms used in the performance comparison.
`ILP`. This is based on solving the ILP presented in Section IV-A. It selects the transmission at each time slot, which globally maximizes the spatial sharing gain while achieving minimum completion time T. Therefore, it provides the lower bound for T. We solve the ILP by employing Gurobi[2] along with CVX[3] in MATLAB environment.
`mmDiMu`. This is our distributed algorithm that considers both relaying and spatial sharing. While suboptimal, `mmDiMu` scales well regardless of the network density. The detail of the algorithm is as presented in Section IV-B. Unlike `ILP`, `mmDiMu` uses a distributed approach, in which each PN makes the transmission decision autonomously.
`OMS` [8]. This algorithm is a sub-category of a multicast with adaptive beamwidth scheduling algorithm. It provides optimal performance for multicast applications in conventional networks, capitalizing on the opportunistic gain. Essentially, `OMS` sorts the nodes according to their channel SNR and serves the subset of nodes that maximizes the instantaneous sum throughput.
`FHOMB` [7]. Finite horizon opportunistic multicast beamforming (`FHOMB`) is designed specifically to minimize the completion time when sending a finite number of packets to multicast

receivers. At each time slot, a subset of nodes is selected such that the estimated completion time is minimized. The estimated completion time is obtained by maximizing the minimum rate using multi-lobe beam; this beam multicasts (usually at a low broadcast rate) to the remaining receivers.
`Adapt` [2]. This is a scalable heuristic which groups the multicast nodes in subgroups using a hierarchical structure to construct the multicast tree. An example scheduling is as depicted in Fig. 1b. Once the subgroups/beam are determined, the source node serves each multicast subgroup sequentially through the beams; the transmit rate at each beam is thus limited by the node with the lowest SNR within each beam.

C. Evaluation Settings

To evaluate the performance of each algorithm, we examine the impact of two main parameters: (1) the number of nodes N and (2) the beamwidth $w = 360°/L$ at the transceivers. Due to the high complexity of ILP, i.e., $O(2^N)$, N is restricted to 10 in scenarios where ILP is involved for comparison. The rest of the algorithms are evaluated for up to $N = 100$. We evaluate the performance for transmitter beamwidth $w = \{15°, 30°, 45°, 60°, 90°\}$. Note that, the transmit beamwidth w has an impact on the transmission gain [24], which we account for in the computation of the receiving rate. Unless mentioned otherwise, at the receiver side, we assume that it uses a quasi-omnidirectional mode for receiving.

To ensure fair performance comparison between the algorithms, all algorithms use the same simulation setting. The minimum beamwidth is determined by w in each simulation scenario in Section V-D, and the beamwidth resolution is thus multiple of w for all the algorithms except `Adapt`. Since `Adapt` operates based on adapting its beamwidth to the multicast group, it can freely adjust its beamwidth as long as the minimum beamwidth is w. For instance, when the simulation has a setting of $w = 45°$, `Adapt` could have any beamwidths between $45°$ and $360°$ while the other algorithms could only have beamwidths that are a multiple of $45°$, i.e., $\{90°, 135°, ..., 360°\}$.

We implemented all the algorithms in Matlab and conducted the comparisons using the above settings. For each data point, we average the data over 200 simulation runs and compute the corresponding 95% confidence interval.

D. Simulation Results

As defined in (4) in Section III, the completion time T is the time required for all network nodes to finally receive the multicast data (by summing up the duration t^s at all time slots). Specifically, it is represented by the time, at which the last multicast node receives its data.

1) Impact of the number of nodes N: Here, we evaluate the impact of different N, $N = \{2, 4, 6, 8, 10\}$, on the completion time T by fixing the transceivers beamwidth $w = 45°$.

As a general trend, Fig. 3 shows that increasing the number of nodes N also increases the completion time T. When N is large, the number of multicast slots required to transmit to all the nodes increases as well. `ILP` performs best as
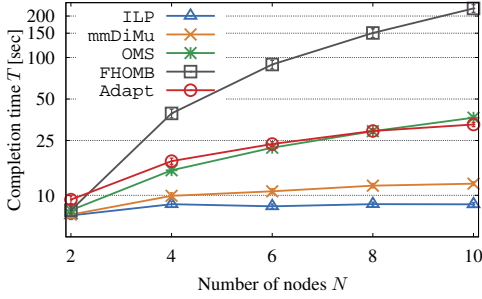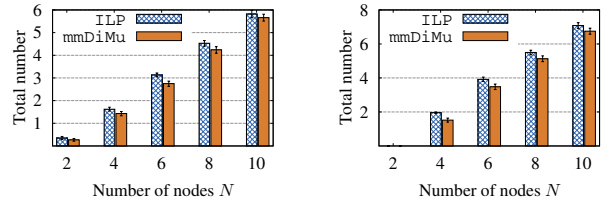
Fig. 3: Completion time T for different N with $w = 45°$.

(a) Relay transmission.

(b) Concurrent transmission.

Fig. 4: Total number of transmissions exploiting relay and spatial sharing for ILP and mmDiMu with $N = \{2, 4, 6, 8, 10\}$.

(a) Relay transmission.

(b) Concurrent transmission.

Fig. 5: The fraction of relay and concurrent transmissions for ILP and mmDiMu with $N = \{2, 4, 6, 8, 10\}$.

TABLE II: An example of multicast scheduling for ILP and mmDiMu for the scenario in Fig. 2.

Algorithm	ILP		mmDiMu	
	Transmission link	Time	Transmission link	Time
time slot, $s = 1$	⓪ → ③	1	⓪ → ③	1
time slot, $s = 2$	⓪ → ①②	3	⓪ → ①②	3
	③ → ⑤	2	③ → ④	1
time slot, $s = 3$	③ → ④	1	③ → ⑤	2
Completion time, T	5sec		6sec	

it picks the best policy which results in minimum T, as formulated in (4). It indeed only requires 23.54%, 3.79%, and 30.33% of the multicast completion time required by OMS, FHOMB, and Adapt, respectively, for $N = 10$. Specifically, ILP achieves a reduction in completion time by up to 96.21% as compared to the other algorithms. Our proposed algorithm mmDiMu also demonstrates a high gain in completion time. It achieves completion time reduction of up to 66.78%, 94.65%, and 62.77% over OMS, FHOMB, and Adapt, respectively.

Interestingly, while OMS performs well in conventional single-hop systems, it performs slightly worse than Adapt as N increases. As N increase, so as the SNR diversity of the nodes. In such a case, OMS will first opportunistically transmit to the node that has higher SNR. This behavior results in excluding the nodes with low SNR initially. As a result, it suffers from low transmitting rate at a later time; it still has to serve the remaining nodes that have lower SNR. Unlike OMS, Adapt groups the nodes based on angular and then divides the group to minimize the transmission time and form a binary tree structure. Therefore, it refrains from the suboptimality that comes from greedily scheduling the nodes with better SNR. On the other hand, OMS performs better than Adapt for smaller N because the probability of having nodes at the edge is much smaller. Furthermore, OMS may use more than one (disjoint) beam to serve all the nodes, while this option is unavailable in Adapt. Therefore, sparse distribution of nodes – this mostly occur when the node density is low (i.e., small N) – harms the performance of Adapt.

Similarly, FHOMB in [7] that performs well in single-hop multicasting, performs poorly here. In FHOMB, a node receives the complete frame over multiple fixed-length time slots. At each slot, the policy (i.e., the subset of nodes to transmit to) which gives the lowest estimated completion time (up to the time all nodes received the frame) is chosen. As mentioned, to determine the estimated completion time, the remaining nodes are served with broadcast. In mmWave networks, broadcasting in all direction results in a very low transmission rate. Therefore, the estimated completion time is significantly longer than a slot time. Here, lower estimated time is favored since it provides a lower total transmission time. In most cases, this comes at the expense of a long slot duration t^s. As seen in Fig. 3, this results in high completion time.
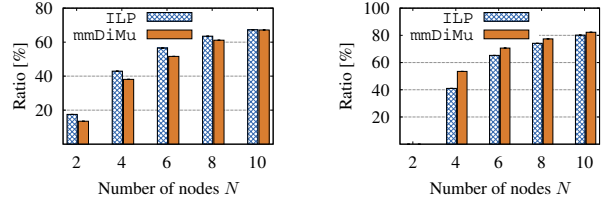
As expected, mmDiMu performs worse than ILP because it autonomously schedules its transmission, disregarding the decision made by other PNs in the system. Let's consider the scenario in Fig. 2 and the corresponding schedule in Table II. The completion time of ILP is 1s lower than that of mmDiMu. Since mmDiMu sorts the nodes according to their SNR, the parent for ④ and ⑤ is ③, and ④ is served first. This results in $t^{s=3} = 2$s. However, ILP is aware that scheduling node ⑤ first results in optimal completion time. As N increases, the occurrence of this event increases as well. This reflects in the higher gain for ILP for larger N.

Remark: The low complexity mmDiMu only requires 29.15% additional completion time, in the worst case $N = 10$, as compared to ILP. Nevertheless, this additional time is significantly lower than that required by other algorithms.

2) The importance of joint relaying and spatial sharing: The substantial gain in the completion time demonstrated by our proposed algorithms (i.e., ILP and mmDiMu) emphasizes the importance of leveraging the relaying and spatial sharing gains jointly in mmWave multicast networks. To shed light on this aspect, Fig. 4 and Fig. 5 depict the number and ratio, respectively, of the relay and concurrent transmissions for ILP and mmDiMu. A transmission is a relay transmission if the transmitter is not the source node. A transmission pair is defined as a concurrent transmission if there is more than one transmission within the same time slot. For instance, in Table II, the number of relay transmission is 2 (i.e., ③→④ and ③→⑤), and the number of concurrent transmissions is 2 (i.e., ⓪→① ② and ③→⑤) for ILP.

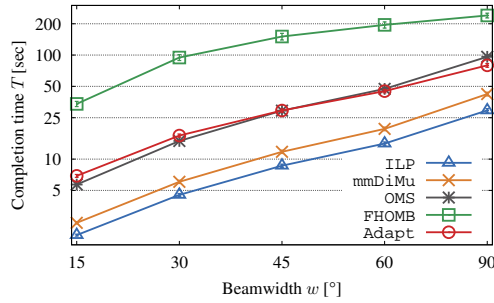In Fig. 4, the total number of relay and concurrent transmissions increases consistently with N. This increase is due to a

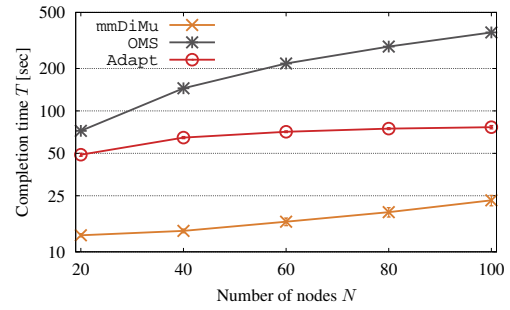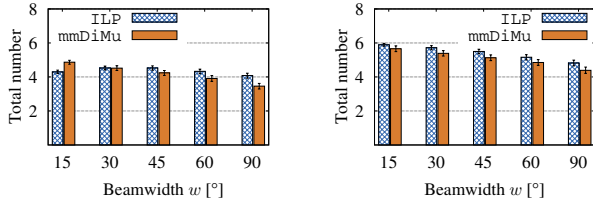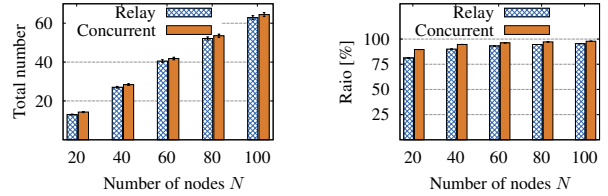Fig. 6: Completion time T for different w. $N = 8$.

Fig. 8: Completion time T for $w = 45°$.

(a) Relay transmission.   (b) Concurrent tranmsission.

Fig. 7: Total number of relay and concurrent transmissions for ILP and mmDiMu with $w = \{15°, 30°, 45°, 60°, 90°\}$.

(a) Number of transmissions.   (b) Ratio of transmission.

Fig. 9: Total number and ratio for relay and concurrent transmissions for mmDiMu with $N = \{20, 40, 60, 80, 100\}$.

higher communication diversity. We observe the total number of relay (in Fig. 4a) and concurrent (in Fig. 4b) transmissions of ILP is consistently higher than that of mmDiMu. This indeed contributes to ILP outperforming mmDiMu. Firstly, ILP has a precise view of the entire network and knows the optimal policy; it first transmits to the nodes that can transmit with a high rate to another node later while maximizing spatial sharing gain. Unlike ILP, at each slot, each PN in mmDiMu opportunistically transmits to the CN set that maximizes the instantaneous sum throughput; the set of selected CNs is usually those that are located nearer to the PN. As a result, the CN set may not necessarily be the optimal set to relay the data to the remaining nodes at a later time. Secondly, each CN in mmDiMu only selects one PN. That said, a CN does not choose a secondary PN even if it potentially allows concurrent transmissions. As a result, this reduces the number of relay and concurrent transmissions of mmDiMu, and thus resulting in a higher completion time (as shown in Fig. 3).

Further, we observe a high ratio of relay (up to 70%) and concurrent (up to 80%) transmissions over the corresponding total number of transmission for both ILP and mmDiMu. Precisely, a high number of concurrent transmission (in Fig. 4b) does not directly translate into a high number of the ratio (in Fig. 4b), but it highly depends on the total number of transmissions. This ratio confirms a large fraction of the performance gain roots from the exploitation of relaying and spatial sharing.
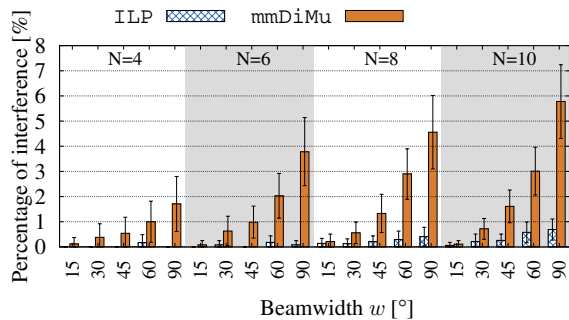
Remark: The gain achieved by ILP and mmDiMu mainly comes from the extensive exploitation of relaying and spatial sharing. This confirms the importance of leveraging these gains for mmWave multicast networks.

3) Impact of beamwidth w: This section evaluates the impact of $w = \{15°, 30°, 45°, 60°, 90°\}$ on the completion time T, while fixing the number of nodes $N = 8$.
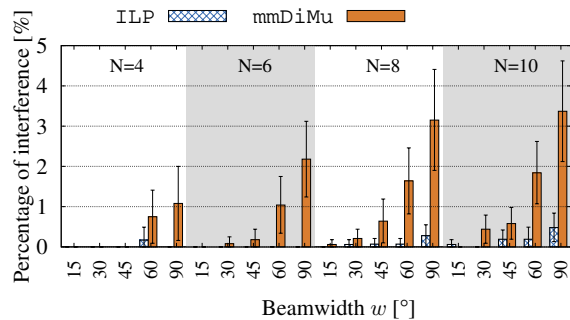
Fig. 6 shows a general trend that increasing beamwidth increases the completion time T. A wider beamwidth w results in lower transmitting and receiving gains, which in turns results in a low data rate and a longer transmission time, and thus a high completion time T. This especially makes an impact on the algorithms that do not leverage relay. In particular, CNs located far away from the source node have to be served with very low transmission rates. Therefore, we observed an abrupt increase in the completion time of OMS, FHOMB and Adapt; the transmission time increases by 90.55s, 206.20s, and 72.74s, respectively, as w increases from $15°$ to $90°$. By manipulating relay, these CNs are reachable through a closer relay PN, resulting in a higher transmission rate. Therefore, the increase in transmission time for ILP and mmDiMu is lesser, i.e., only 15.26s and 22.66s, respectively, as w increases from $15°$ to $90°$. Although the increase seems insignificant, it is still non-negligible. A wider w improves the coverage area and a PN could cover more CNs. As a result, the number of relay and concurrent transmissions reduces, and the completion time increases. This is evident from the decreasing number of these transmissions as w increases, as depicted in Fig. 7.

Remark: Although a wider beamwidth increases the completion time, ILP and mmDiMu are less impacted by it, as compared to the other algorithms.

4) Scalability: All previous results in this section only consider a maximum N of 10. This is due to the complexity and scalability issue of ILP. Nevertheless, it remains important as it provides insights on the algorithm performance difference to the optimal ones. Here, we demonstrate that even with a large N, our proposed mmDiMu algorithm achieves a significant gain as oppose to OMS and Adapt. FHOMB is removed from the comparison as it performs poorly even for cases with smaller

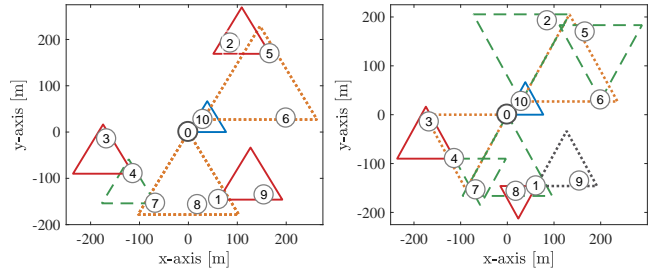(a) Omni-directional antenna receivers.   (b) Directional antenna receivers.

Fig. 10: Percentage of interference between concurrent transmission of ILP and mmDiMu for up to $N = 10$.

N. We set $N = \{20, 40, 60, 80, 100\}$ with $w = 45°$ for the following results.

Fig. 8 depicts a significant increase in the completion time T for OMS, but not for Adapt, as N increases. As N is large and continuously increases, the value of the lowest SNR does not change much, so does the transmit rate at each beam of the Adapt algorithm. However, OMS greatly suffers from its opportunistic decisions. mmDiMu experiences less increment in completion time than OMS. Increasing N increases the number of relay nodes and the opportunity of spatial sharing. This is evident from Fig. 9, where the number and ratio of relay and concurrent transmissions increase with N.

Remark: mmDiMu scales very well with the network density and achieves a significant reduction in the completion time as compared to OMS and Adapt.

5) Impact of interference: In theory, mmWave links mimic a pencil beam, and thus interference is negligible. However, the current off-the-shelf mmWave devices have a wider beam. In addition, the limitation in antenna design renders this assumption valid only in theory. Here, we evaluate the impact of transmit beamwidth $w = \{15°, 30°, 45°, 60°, 90°\}$ for N up to 10 and 100 for ILP and mmDiMu, respectively, on the probability of mutual interference between concurrently transmitting pairs. We evaluate for interference characteristic for two type of antenna receiving modes: quasi-omnidirectional and directional, In the first case, the receiver suffers from interference as long as it is within the beam's coverage of the transmitter. This type of receiving mode is as employed by default in the existing off-the-shelf devices (i.e., TP-Link Talon AD7200 multi-band wifi router [25]). In a very recent work on improving beam alignment in the mmWave device [26], the authors are able to adaptively adjust the existing codebook available in the IEEE 802.11ad devices and optimize the beam pattern to obtain a higher directionality beam. This shows the feasibility of implementing such receiving mode and thus it is important to also evaluate for interference when the receiver is in directional receiving mode. In this case, to cause interference, not only that the receiver has to be within the beam coverage of the interfering transmitter, but the transmitter must also be within the coverage area of the receiving beam. Fig. 10a and Fig. 10b depict the percentage of mutual interference between the concurrently transmitting links as beamwidth w increases for the respective

(a) ILP with four slots.   (b) mmDiMu with five slots.

Fig. 11: Interference analysis for $N = 10$ and $w = 60°$. The difference color represents the transmission at different slots.

type of receiver. Note that, due to the complexity of OMS, the interference percentage is only shown for up to $N = 10$.

Quasi-omnidirectional receiver: Fig. 10a shows a general trend in which the percentage of interference increases with beamwidth w and number of nodes N. As w increases, so as the coverage area a transmitter, and thus increasing the probability of interfering the nearby nodes. As N increases, so does the density of the network. That said, the probability that one or more receiving node falling within the coverage area of a transmitter is higher, and thus the percentage of interference. Although increasing beamwidth causes higher interference, it only leads to a maximum percentage of interference of up to 6% (see Fig. 10a) in the largest N scenario; there is no interference in most scenario. Since the source of interference is due to the frequency of concurrent transmissions, mmDiMu experiences a slightly higher interference than ILP. mmDiMu indeed has a higher ratio of concurrent transmission as compared to ILP (refer Fig. 5b). As shown in the example scenario in Fig. 11, ILP and mmDiMu has 7 and 10 total transmissions, respectively. Out of those, ILP and mmDiMu has 5 and 9 links, respectively, involved in concurrent transmission, which results a ratio of 71.43% and 90%, respectively. While ILP has no interference among the communication links, transmission from node ⓪ of mmDiMu (see Fig. 11b) causes interference to node ⑦ when node ④ transmits to ⑦ simultaneously with ⓪→①. Nevertheless, even with omni-directional receiving mode, the percentage of interference in both algorithms is kept below 6%.

Directional receiver: When the receiver uses directional receiving mode, the percentage of interference becomes smaller
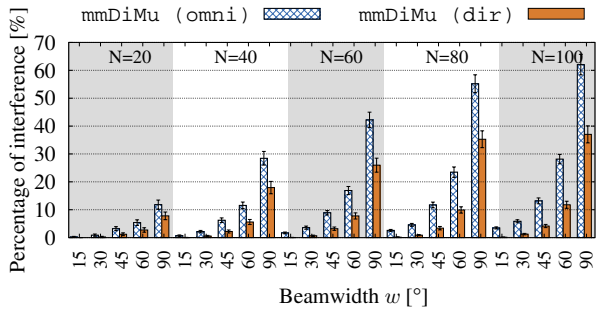
Fig. 12: Percentage of interference between concurrent transmission of `mmDiMu` for omni-directional (omni) and directional (dir) receiver.

(a) Synchronous transmission

(b) Asynchronous transmission

Fig. 13: The difference between synchronous and asynchronous scheduler.

(see Fig. 10b). This is due to the reason that interference only occur when the transmitter is within the beam of the receiver's beam, and the beam coverage is limited by the receiver's beamwidth w. For instance, while simultaneous transmission of ⓪→① and ④→⑦ causes interference in the quasi-omnidirectional receiver's case, here, nodes ① and ⑦ use directional reception, and thus avoiding interference from nodes ④ and ⓪, respectively; the interfering nodes ④ and ⓪ are not within the directional receiving beam of nodes ① and ⑦, respectively. Therefore, we observed drops in the percentage of interference by up to 2.4% (i.e., when $N = 10$, $w = 90°$ for `mmDiMu`). The general performance trend is as seen in Fig. 10a for the same reasons explained above.

Scenario with N up to 100: In Fig. 12, we show the interference's percentage of `mmDiMu` for up to $N = 100$ in order to provide some insight onto implementation setup for higher density scenarios. As seen, using directional receiving mode clearly provides a much lower percentage of interference for scenarios with higher density and beamwidth; the reduction is up to 25.01% for $N = 100$ and $w = 90°$. If the location of the receiver is known and the accuracy of beam alignment is high, using narrow beamwidth such as $15°$ only has percentage of interference of up to 0.13% in the worst case; many practical research work on mmWave use horn antenna with $w = 7°$ [27].

Remark: Even for beamwidth as wide as $w = 45°$, `mmDiMu` manages to keep the interference's percentage below 5% for $N = 100$. We foresee future mmWave devices with highly directional and adjustable beam, in which, given any scheduling decision, the interference between the concurrently transmitting pairs in dense network can be further minimized.

VI. DISCUSSION AND FUTURE WORK

We dedicate this section to discuss the aspects that are out of scope of this paper yet are crucial to consider in developing a mmWave multicast scheduling algorithm.

A. Mobility

The directional mmWave communication limits the coverage area. In this paper, we assume the nodes are static during the transmission period; a transmission period is on average equivalent to 1s for transmitting a data frame of 1Gbits. We
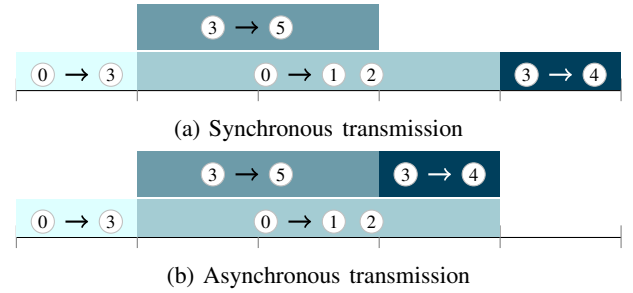
deem this a valid assumption for a slow mobility network such as sports stadiums, concert halls, or urban vehicular scenarios with low speed cars. For networks with high-speed nodes such as high-speed train, uninterrupted connectivity can be ensured with features such as nodes tracking and beam switching.

B. Asynchronous Scheduling

In this paper, we model the system as a slotted system with a synchronous slot, as shown in Fig. 13 (which is based on the scenario in Fig. 2), where each transmission starts only at the beginning of each slot. Our proposed `ILP` algorithm is indeed designed to minimize the time difference between the simultaneously communicating pairs, but some small time gaps may persist. For instance, at time slot 2 ($s = 2$) in Fig. 13a, ③→⑤ requires only 2s to complete transmission, while ⓪→① ② requires 3s. The time gap of 1s at ③ can potentially be used for another communication. As shown in Fig. 13b, ③→④ starts immediately as ③→⑤ is completed. This way, the network transmission time is reduced by 1s. While asynchronous scheduling improves the network transmission time, it results in a higher complexity algorithm. Therefore, the trade-off between gain and complexity must be considered carefully.

C. Scheduling Synchronization

In this paper, we assume that the scheduling decision is known by all the multicast nodes, and thus synchronization of transmissions among the nodes is feasible. However, the broadcast of scheduling information using mmWave is unreliable; mmWave is prone to blockages and suffers from high propagation loss. Therefore, the algorithm relies on information dissemination using the robust sub-6GHz transmissions. In fact, the protocol, namely fast session transfer (FST)[4], that supports the coordination between mmWave and sub-6GHz interface for such purpose has already been outlined in the IEEE 802.11ad standard [28]. Specifically, the multicast nodes can exchange important scheduling information via sub-6GHz interface and the mmWave interface is dedicated for high rate data transmission only. In vehicular networks, such information can be exchanged via the robust dedicated short-range communication (DSRC) radio interface operating at 5.9GHz.

---

[4] FST transfers the session between two physical channel to exchange information.

D. Blockages

A link is identified as a blocked link in the absence of either LOS or NLOS path. Based on the IEEE 802.11ad standard, this information could be obtained via nodes discovery phase upon the network initialization. In particular, during this phase, each communication pair performs beam training. When a transceiver pair fails to discover each other, the link between them is blocked. While this is out of the scope of this paper, it can nevertheless be easily extended by removing a CN from its PN within the CNs matrix upon the identification of a blocked link. Proceeding with our algorithms, the blocked CNs will receive data only from a non-blocked relay PN.

VII. CONCLUSION

In this paper, we investigate the challenge of multicasting in mmWave networks. We consider to jointly leverage *relay transmission* to improve the reachability and link rate and *spatial gain* by enabling simultaneous unicast and/or multicast communications. We formulate the problem with an ILP and provide a distributed solution called `mmDiMu`. The ILP solution generates optimal scheduling decisions while suffering from poor scalability. `mmDiMu` performs closely to the optimal and can scale to large networks with very dense settings due to its distributed nature. We show through extensive simulation that our proposed optimal `ILP` and distributed `mmDiMu` solutions provide significant gain over the multicast scheduling methods designed for sub-6GHz networks, in which we achieve up to 96.21% reduction in completion time. Furthermore, in comparison with the adaptive beamwidth algorithm (namely `Adapt`) proposed for mmWave multicasting, we gain up to 78.22% in completion time. Noteworthily, although interference reduction is excluded from the optimization objective, we achieve an impressively low (i.e., 5%) total interference even with $45°$ beamwidth in high-density network scenarios.

There are still interesting open problems, such as studying the impact of user mobility, blockage, the tradeoff between efficiency, and complexity in asynchronous scheduling. We leave these for future work.

REFERENCES

[1] M. Xiao, S. Mumtaz, Y. Huang, L. Dai, Y. Li, M. Matthaiou, G. K. Karagiannidis, E. Bjoernson, K. Yang, C. L. I, and A. Ghosh, "Millimeter Wave Communications for Future Mobile Networks," *IEEE JSAC*, vol. 35, no. 9, pp. 1909–1935, Sept 2017.

[2] A. Biason and M. Zorzi, "Multicast via Point to Multipoint Transmissions in Directional 5G mmWave Communications," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 88–94, Feb 2019.

[3] H. Park, S. Park, T. Song, and S. Pack, "An Incremental Multicast Grouping Scheme for mmWave Networks with Directional Antennas," *IEEE Communications Letters*, vol. 17, no. 3, pp. 616–619, Mar 2013.

[4] A. Biason and M. Zorzi, "Multicast Transmissions in Directional mmWave Communications," in *EW*, May 2017, pp. 1–7.

[5] X. Lin and J. G. Andrews, "Connectivity of Millimeter Wave Networks With Multi-Hop Relaying," *IEEE Wireless Communications Letters*, vol. 4, no. 2, pp. 209–212, Apr 2015.

[6] J. Du, E. Onaran, D. Chizhik, S. Venkatesan, and R. A. Valenzuela, "Gbps User Rates Using mmWave Relayed Backhaul With High-Gain Antennas," *IEEE JSAC*, vol. 35, no. 6, pp. 1363–1372, Jun 2017.

[7] G. H. Sim and J. Widmer, "Finite Horizon Opportunistic Multicast Beamforming," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1452–1465, Mar 2017.

[8] T. P. Low, P. C. Fang, Y. W. P. Hong, and C. C. J. Kuo, "Multi-Antenna Multicasting with Opportunistic Multicast Scheduling and Space-Time Transmission," in *Globecom*, Dec 2010, pp. 1–5.

[9] U. C. Kozat, "On the Throughput Capacity of Opportunistic Multicasting with Erasure Codes," in *INFOCOM*, Apr 2008.

[10] T. P. Low, M. O. Pun, Y. W. P. Hong, and C. C. J. Kuo, "Optimized Opportunistic Multicast Scheduling (OMS) over Wireless Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 2, pp. 791–801, Feb 2010.

[11] G. H. Sim, J. Widmer, and B. Rengarajan, "Opportunistic Finite Horizon Multicasting of Erasure-Coded Data," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 705–718, Mar 2016.

[12] S. Sen, J. Xiong, R. Ghosh, and R. R. Choudhury, "Link Layer Multicasting with Smart Antennas: No Client Left Behind," in *ICNP*, Oct 2008, pp. 53–62.

[13] T. H. Chang, Z. Q. Luo, and C. Y. Chi, "Approximation Bounds for Semidefinite Relaxation of Max-Min-Fair Multicast Transmit Beamforming Problem," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3932–3943, Aug 2008.

[14] E. Aryafar, M. A. Khojastepour, K. Sundaresan, S. Rangarajan, and E. Knightly, "ADAM: An Adaptive Beamforming System for Multicasting in Wireless LANs," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1595–1608, Oct 2013.

[15] K. Sundaresan, K. Ramachandran, and S. Rangarajan, "Optimal Beam Scheduling for Multicasting in Wireless Networks," in *MobiCom*, Jun 2009, pp. 205–216.

[16] H. Zhang, Y. Jiang, K. Sundaresan, S. Rangarajan, and B. Zhao, "Wireless Multicast Scheduling With Switched Beamforming Antennas," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1595–1607, Oct 2012.

[17] S. Naribole and E. Knightly, "Scalable Multicast in Highly-Directional 60-GHz WLANs," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2844–2857, Oct 2017.

[18] H. Chu, P. Xu, W. Wang, and C. Yang, "Joint Relay Selection and Power Control for Robust Cooperative Multicast in mmWave WPANs," *Science China Information Sciences*, vol. 59, no. 8, p. 082301, Jan 2016.

[19] W. Feng, Y. Li, Y. Niu, L. Su, and D. Jin, "Multicast Spatial Reuse Scheduling over Millimeter-wave Networks," in *IEEE IWCMC*, Jun 2017, pp. 317–322.

[20] G. R. MacCartney, T. S. Rappaport, and A. Ghosh, "Base Station Diversity Propagation Measurements at 73 GHz Millimeter-Wave for 5G Coordinated Multipoint (CoMP) Analysis," in *IEEE Globecom (GC Workshops)*, Dec 2017, pp. 1–7.

[21] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter Wave Channel Modeling and Cellular Capacity Evaluation," *IEEE JSAC*, vol. 32, no. 6, pp. 1164–1179, Jun 2014.

[22] M. Zagalj, J. Hubaux, and C. C. Enz, "Minimum-energy Broadcast in All-wireless Networks: : NP-completeness and Distribution Issues," in *MobiCom*, 2002, pp. 172–182.

[23] G. R. MacCartney and T. S. Rappaport, "Rural Macrocell Path Loss Models for Millimeter Wave Wireless Communications," *IEEE JSAC*, vol. 35, no. 7, pp. 1663–1677, July 2017.

[24] P. Wade. (2000) Feeds for Parabolic Dish Antennas. [Online]. Available: https://www.qsl.net/n1bwt/app-6a.pdf

[25] L. TP-Link Technologies Co. (2018) Talon AD7200 Multi-Band Wi-Fi Router. [Online]. Available: https://www.tp-link.com/us/products/details/cat-9_AD7200.html

[26] J. Palacios, D. Steinmetzer, A. Loch, M. Hollick, and J. Widmer, "Adaptive Codebook Optimization for Beam Training on Off-the-Shelf IEEE 802.11Ad Devices," in *MobiCom*, 2018, pp. 241–255.

[27] G. H. Sim, A. Asadi, A. Loch, M. Hollick, and J. Widmer, "Opp-relay: Managing Directionality and Mobility Issues of Millimeter-wave via D2D Communication," in *COMSNETS*, Jan 2017, pp. 144–151.

[28] "IEEE Standard: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY): Enhancements for Very High Throughput in the 60 GHz Band," *IEEE Std 802.11ad-2012 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012 and IEEE Std 802.11aa-2012)*, pp. 1–628, Dec 2012.