Differentiated Service Support in Wireless Networks with Multibeam Antennas

L. Rajya Lakshmi and Biplab Sikdar

Abstract—**Multibeam antenna arrays (MBAAs) have the capability to improve the capacity of a wireless network by facilitating simultaneous transmissions to multiple users. However, in practical deployments of wireless personal or local area networks (WPANs/WLANs) where piconet coordinators or access points (PNCs/APs) are deployed with MBAAs, it is quite likely to observe non-uniform node densities in various regions. To optimally utilize MBAAs in such scenarios, concurrent transmission scheduling in WPANs/WLANs is formulated as a multi-objective optimization problem. Then, a practical heuristic transmission scheduler that aims to maximize the number of concurrent communications by dynamically configuring the directions of beams is proposed. In addition, a service tag based fair scheduler is also proposed to achieve weighted fairness in WPANs/WLANs with MBAAs. Our results show that the performance improvements provided by the proposed heuristic scheduler are higher for antennas with lower beamwidths, and as the non-uniformity in the network increases, the traffic supported in the network increases in the range 24-41%, compared to the existing methods. The proposed service tag based scheduler can further improve the network throughput and achieve better fairness at the cost of a few beam direction reconfigurations, as compared to the existing methods and our heuristic scheduler.**

Index Terms—**STDMA scheduling, MBAAs, WPANs/WLANs, fair resource allocation.**

I. Introduction

Multibeam antenna arrays (MBAAs) consist of multiple antenna elements. Using advanced signal processing algorithms, MBAAs can adjust the magnitude and phase of the transmitted signals to form multiple directional beams in the required directions, and nulls in the unwanted directions [1]. As a result, they facilitate simultaneous transmissions to multiple neighbors of a node and provide many advantages such as increased throughput, enhanced spatial reuse, and improved coverage range.

Resource scheduling has been an important issue to achieve efficient and fair resource utilization. The existing scheduling methods for wireless networks with MBAAs have been designed to facilitate concurrent communications in multi-hop networks [1], [2], indoor cellular scenarios [3], and wireless local area networks (WLANs) [4]–[6]. However, in typical deployments of wireless personal area networks (WPANs)/WLANs, all regions surrounding the pico net coordinators/access points (PNCs/APs) may not have an equal number of nodes. If the beams of MBAAs are fixed in certain directions, then a beam with lower demand completes its data transmission before a beam with higher demand. Consequently, as time progress, the number of simultaneous transmissions may decrease gradually. *To maximize the number of simultaneous transmissions supported by the available power, and thereby the network throughput, an efficient methodology is required to dynamically configure the directions of beams towards locations with active nodes.*

Fair scheduling is an important and active research area to provide differentiated service to flows with different weights. In the literature, various methods have been proposed for fair resource allocation in packet scheduling [7]–[9], and different kinds of wireless networks [10]–[17]. The most popular fairness model proposed for wireline networks, the *fluid fair queueing* (FFQ) model, ensures that in any small window of time, each backlogged flow receives service in proportion to its normalized weight. For ad hoc networks where location dependent contention has to be considered while addressing fairness among the flows, both centralized [10], [12], [13] and distributed [11], [14] scheduling methods have been proposed. However some of the unique characteristics of WPANs/WLANs with MBAAs discussed below prevent the direct applicability of the existing fair scheduling methods.

- Most of the existing literature on fair queueing assumes either perfect or partial per-flow information, which is not available for the PNCs/APs. In particular, the PNC has a very limited information about the uplink flows in the form of the transmission requests received from the nodes. Utilizing this limited information, identifying continuously backlogged flows and maintaining fairness among the flows with different weights is non-trivial.
- The existing centralized fair schedulers generate schedules on per-slot basis, which results in a huge control overhead when frames are subdivided into a large number of slots. Since the transmitting/receiving nodes change on per-slot basis, the beamforming overhead is also very high. Thus, new methods that incur a lower overhead need to be developed.
- Fair scheduling needs further investigation in order to support directional transmissions. To utilize the full potential of MBAAs, fair allocation has to be performed while utilizing the available energy and maximizing the number of beams busy with transmission/reception.

To address the issues identified above, this paper develops two schedulers that target to handle *control and beamforming overhead reduction* and *fairness maintenance* separately. The main contributions of this paper can be summarized as follows. To maximize the number of simultaneous trans-

L. Rajya Lakshmi and Biplab Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (e-mail: rajya99@gmail.com, bsikdar@nus.edu.sg).

missions while minimizing the control overhead and considering power constraints, concurrent transmission scheduling in WPANs/WLANs with MBAAs is formulated as a multi-objective optimization problem. To generate a concurrent transmission schedule in real-time, we propose a heuristic scheduler which progressively configures beams so that each beam addresses almost equal fraction of the pending demand. In addition, an efficient strategy is also developed, to reduce beam configuration overhead. To provide weighted fair allocation to flows with different weights, we propose a service tag based fair scheduler called "Maximize Busy Beams using Swapping Window Fair Scheduler (MB^2SW-FS)". It maximizes the number of simultaneous communications by dynamically adjusting the directions of beams and generates schedules that involve a lower control overhead compared to the existing methods.

The rest of the paper is organized as follows. The related research is discussed in Section II. The importance of dynamic beam configuration, the system model, and the proposed heuristic scheduler are presented in Sections III, IV, and V, respectively. Section VI obtains the performance bounds of the proposed heuristic scheduler. The details of MB^2SW-FS are described in Section VII. The analytical properties of MB^2SW-FS are presented in Section VIII. Section IX evaluates the performance of the proposed methods. Section X concludes the paper.

II. Related Work

A. Scheduling in WLANs/WPANs

In the related research, a contention based channel access mechanism is proposed in [4] for WLANs, where APs are deployed with MBAAs. The access mechanisms proposed in [5], [6] address various problems such as receiver blocking, mobility related issues, beam load unbalance, quality of service (QoS) provisioning, etc. that may arise in WLANs with multibeam APs. To address QoS provisioning in ad hoc networks with MBAAs, a bandwidth reservation protocol is proposed in [1]. To improve network throughput and reduce delay, a distributed receiver-oriented multiple access scheduling protocol is proposed in [2]. Spatial time division multiple access (STDMA) mechanisms for 60 GHz WPANs with directional antennas have been proposed in [18]–[20]. In contrast to these works, this paper addresses maximization of the number of simultaneous transmissions while minimizing the beam direction reconfigurations, by formulating STDMA scheduling of WPANs/WLANs with MBAAs as an optimization problem. To obtain a schedule in real-time, we also propose a heuristic scheduler. The beam tracking method proposed in [21], configures beams in such a way that the number of nodes covered by each beam is maximized. In contrast, our proposed method aims to maximize the number of simultaneous transmissions when node distribution is non-uniform. When enough power is available at the central controller to form more than one beam, the proposed method maximizes the busy beams by forming each beam in such a way that scope for other beams to get configured is maximized.

B. Fair Scheduling

The FFQ model [7], [8] addresses packet fair queueing in wireline networks by modeling each flow as a fluid flow passing through a channel of capacity C and ensures a fair share of channel allocation to each flow in proportion to its normalized weight. The start-time fair queuing (SFQ) model [9] seeks to achieve fairness in spite of the variations in the server capacity.

In the case of ad hoc networks, the lookahead packet fair queueing model proposed in [10] uses a lookahead window to exploit the spatial reuse while bounding the unfairness among the flows. The packet fair queueing models proposed in [12] target to maximize the network throughput. In each slot, along with the flows that obey the fairness constraints, the maximum possible number of flows that violate the fairness constraints but are eligible for concurrent transmission are also scheduled. A credit based fair queueing model is proposed in [13] for code division multiple access based ad hoc networks. On the other hand, with the distributed fairness models proposed in [11], [14], each node coordinates and exchanges its scheduling decisions with its neighbors to achieve end-to-end fairness for multi-hop flows. For ultra-wide band or millimeter-wave (mmWave) WPANs, the weight based fair queueing method proposed in [15] performs scheduling one slot at a time by dynamically computing the control parameters of various flows that make a tradeoff between fairness and the network throughput.

Unlike the existing literature on fair scheduling which considers the scenario where the nodes are deployed with omni-directional antennas, MB^2SW-FS considers a more challenging scenario where the PNCs/APs are deployed with MBAAs and the other nodes are deployed with directional antennas. In contrast to the existing fair schedulers that assume packet-level information of the flows, MB^2SW-FS uses a very limited flow-related information. By dynamically adjusting the directions of beams, MB^2SW-FS maximizes the number of beams busy with communication, and generates schedules that do not change on per-slot basis, which allows nodes to communicate for multiple slots, thus reduces the control overhead.

The methods developed in this paper are different from our previous work [16], [17] in many aspects. In [16], [17], we handle fair allocation in the scenarios where nodes are deployed with single-beam directional antennas, whereas the present paper addresses optimal and fair utilization of MBAAs. The service tag based fair scheduler developed in this paper is capable of providing better fairness compared to the heuristic methods proposed in [16], and it results in a lower control overhead compared to the service tag based fair scheduler proposed in [17].

III. Importance of Dynamic Beam Configuration

We first illustrate the importance of dynamic beam direction configuration using the network shown in Fig. 1. The *cumulative downlink demand* of the sector formed by a beam is the total downlink demand of all nodes located in the sector. Assume that the PNC/AP has sufficient power to form 8 beams simultaneously. Assuming the directions of beams are fixed as

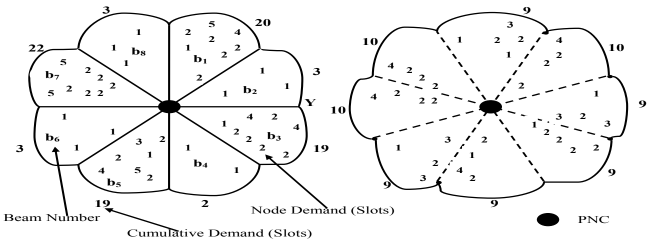


Fig. 1: Fixed beam directions (left) and dynamic beam directions (right).

Case	NBB	IR (N)	PD (N)	Interference (N)
1	1	R_1	3	10
2	2	R_2	1	5
3	2	R_3	1	6

Demand of each node is 5 slots
PD(N): pending demand of node N after 20 slots

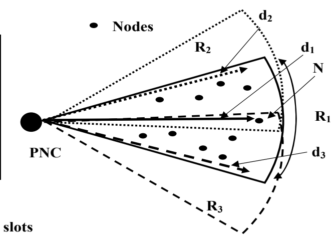


Fig. 2: Interference region of a node.

shown in the left part of Fig. 1, some of the beams remain idle during most of the data transmission period due to the non-uniform distribution of nodes and demands. Beam 4 becomes idle from slot 3. Slot 4 onwards, in addition to beam 4, beams 2, 6, and 8 remain idle, and so on. To satisfy the total traffic demand of the network, 22 slots are required. However, from slot 3, if the PNC changes the directions of beams as shown in the right part of Fig. 1, then all the beams will be busy until slot 11 and the total traffic demand can be accommodated with only 12 slots.

To understand the relation between the interference of a node and the number of busy beams (NBB) (i.e., the number of possible concurrent non-overlapping beams) in the network, consider the network in Fig. 2 where the PNC wishes to configure a beam in the direction of node N, and assume that the PNC has sufficient power to form two simultaneous beams. The figure shows three possible cases, Case 1, Case 2, and Case 3 where the PNC configures one of its beams in directions d_1, d_2, and d_3, respectively, to transmit data to node N. The area covered by the sector of a beam formed to cover node N is called the interference region (IR) of N, the set of nodes, say Z, located in that sector are collectively called the interfering nodes (IN) of N, and the cardinality of Z is called the interference of N. The IRs of node N corresponding to three cases are marked with solid, dotted, and dashed lines, respectively. From the table in Fig. 2, in Cases 2 and 3, the interference of node N is lower and an additional beam can be scheduled in the network (and thus NBB is 2), compared to Case 1. When the nodes in a beam are served in round robin order starting from node N, one slot for a node at a time, the pending demand (PD) of node N is lower in Cases 2 and 3, compared to Case 1. Hence, by configuring the beams in directions with lower interference, the number of busy beams can be increased and the pending demand can be decreased. Our scheme exploits this insight, and for each node N, the PNC identifies a transmission (or beam formation) direction with the least interference, N^D, to utilize during the STDMA schedule generation (Section V-A).

IV. System Model

The considered network architecture resembles a WPAN/WLAN with a number of devices/nodes, and a central coordinator deployed with a MBAA acts as the PNC/AP. The PNC is located at position $(0, 0)$ in the two-dimensional space of the network area and the other nodes are deployed at randomly selected locations. Except for the PNC, all other devices are deployed with steerable directional antennas. With short wavelength at the mmWave, it is possible to fabricate a large number of antenna elements in the transceivers. A directional beam can be produced by the involvement of a certain number of antenna elements, thus the number of beams that can be formed by the MBAA is limited. We assume that the MBAA is capable of forming B non-overlapping beams simultaneously and the beamwidth of each beam can vary between β_{min} and β_{max}. For directional antennas, we use the cone plus sphere model given in [22]. This model is employed in the two-dimensional space assuming all nodes in WPAN are in a plane. With this model, the antenna gain consists of a mainlobe of given beamwidth β and spherical side lobes of beamwidth $(2\pi - \beta)$ and the antenna gains of these lobes are defined as $\vartheta \frac{2\pi}{\beta}$ and $(1-\vartheta)\frac{2\pi}{(2\pi-\beta)}$, respectively, where ϑ is the antenna radiation efficiency. Each node identifies its neighbors using neighbor discovery schemes [23] and computes transmission rates of links to its neighbors [8]. The PNC finds the locations of other devices using a low-dimensional maximal-likelihood (ML) classifier that finds the locations of nodes based on the changes in statistics of sparse beamspace multiple-input and multiple-output (MIMO) channel matrix [24].

Network time is divided into superframes and each superframe consists of four periods: beacon, beam alignment, random access, and data transmission. In beacon periods (BPs) the PNC transmits synchronization and scheduling information. During beam alignment period (BAP), devices and PNC align their beams for data transmission. Devices send their traffic demands to the PNC in the random access period (RAP) of a superframe. Based on these demands, the PNC generates a STDMA schedule for the devices to utilize the slots in the data transmission period (DTP) of the upcoming superframe. The discussion presented in Section V assumes downlink data transmissions. However, the methods developed therein can also be used to schedule uplink data transmissions.

In mmWave networks beam alignment process has been adopted to find the directions of beams for data transmission. This process consists of two phases: sector-level and beam-level beam alignment. By tracking the locations of devices over the time, the sector-level beam-alignment overhead can be reduced substantially [25]. Hence, we assume that the PNC and the other devices know in which sectors their neighbors are located. Since this paper mainly focuses on mmWave communications in indoor environments, this assumption is valid. During beam-alignment phase, the PNC and the other devices search all possible alignment directions by transmitting pilot signals using all possible beam vector combinations.

The RAP of each frame is divided into a number of mini-slots. Each node selects a mini-slot randomly and transmits its

TABLE I: Important notations used in the proposed methods.

Notation	Meaning
β_{min} and β_{max}	The minimum and maximum beamwidth
n	Number of nodes in the network
$W_f(t_1, t_2)$	The service received by flow f in the interval (t_1, t_2)
B	Number of beams available at the PNC
l_{N_i} and c_{N_i}	Traffic demand and transmission rate of N_i
$y_i^{t,\nu}$	Whether N_i is assigned to b_ν in the t-th slot or not
$a_i^{t,\nu}$	Number of slots allocated to N_i in t-th stage
$\beta^{t,\nu}$	Beamwidth of b_ν in t-th stage
$\mathbf{S^t}$	t-th stage
δ_t	Whether t-th stage exists or not
f_{ij}	Angle subtended by N_i and N_j at the PNC
g_{ijx}	Set to 1, if $(f_{ix} \leq f_{ij})$ and $(f_{jx} \leq f_{ij})$
T	Number of available slots
N_i	A node
σ	Swapping window size
IR(N)	Interference region of N
IN(N)	Set of nodes in IR(N)
ω_f	Weight of flow f
σ_1	Threshold on stage duration
γ	Transmission range of a beam
d_{ref}	Vector drawn from $(0,0)$ to $(\gamma, 0)$
$I(f)$, $F(f)$	The *start* and *finish* tags of f
$U(\vec{d_1}, \vec{d_2})$	Angle $\vec{d_1}$ and $\vec{d_2}$ subtend at PNC
$\widetilde{U}(\vec{d_1})$	Angle $\vec{d_1}$ and $\vec{d_{ref}}$ subtend at PNC
θ_t	Duration of t-th stage
Ω_t	Set of beams scheduled in t-th stage

transmission request to the PNC. These request commands are transmitted through a frame that includes the PHY preamble, the frame header, and the payload. The preamble helps in frame detection, synchronization, and channel estimation. The header includes the PHY header, the MAC header, and the header check sequence which help in frame decoding. The payload can support upto 1 Mbytes of data.

According Shannon's theory, the transmission rate of a link (N_i, N_j) is determined by its SINR. The signal power received at node N_i is dependent on various parameters such as path loss, shadowing, fading, etc. The path loss at distance d can be estimated as: $PL(d)[dB] = PL(d_o)[dB] + 10\eta log_{10}\frac{d}{d_o} + X[dB]$, where, η is the path loss exponent and $X[dB]$ is the shadow fading. $PL(d_o)[dB]$ is the path loss exponent at the reference distance d_o, and can be calculated as: $PL(d_o)[dB] = 10log_{10}(g_t g_r \frac{\lambda^2}{16\phi^2 d^2 Q}) + A_{NLOS}$, where λ is the wavelength, Q is the system loss factor, g_t and g_r are the antenna gains of transmitter and receiver, and A_{NLOS} is the attenuation value for NLOS scenarios.

For node N_i located in the sector of beam b_ν, the power received from the PNC can be expressed as: $P_r^{i,b_\nu} = kP_t(h_{i,b_\nu})d_i^{-\eta}$, where $k = 10^{PL(d_o)}/10$ and h_{i,b_ν} is the small-scale fading gain, which is modeled using Rayleigh distribution. The received SINR of node N_i is: $SINR_i = \frac{P_r^{i,b_\nu}}{WN_o + m\sum_{b\neq b_\nu} P_r^{i,b}}$ where, m is the multi-user interference factor and W is the system bandwidth. Now, according to Shannon's theory, the data rate of node N_i can be computed as [19]:

$$R_i \leq \pi W log_2\left(1 + \frac{P_r^{i,b_\nu}}{WN_o + m\sum_{b\neq b_\nu} P_r^{i,b}}\right), \quad (1)$$

where π is efficiency of the transceiver design.

Whenever the link between node N_i and the PNC gets blocked, N_i selects one of its neighbors, N_j, that has the best transmission rate and sends its transmission request to N_j. Node N_j forwards the transmission request of N_i to PNC. It also receives data from the PNC on behalf of N_i and then forwards the same to N_i. Some of the important notations used in the proposed methods are listed in Table I.

V. STDMA Scheduling in Multibeam Networks

Consider a network with n devices. The traffic demand of node N_i is represented by l_{N_i}, and $c_{N_i}(\mathbf{p})$ represents its per-slot transmission rate, where $\mathbf{p} = \{p_1, p_2, \cdots, p_B\}$, and p_1, p_2, $\cdots$, p_B denote the transmission powers of beams b_1, b_2, $\cdots$, b_B, respectively. The STDMA schedule of a frame consists of a set of stages. In each stage $\mathbf{S^t}$, a set of beams, Ω_t, are scheduled for data transmission, where $|\Omega_t| \leq B$ and $|.|$ denotes the cardinality of a set. An STDMA schedule that consists of m stages can be represented as:

$$\Psi = \mathbf{S^1}(\theta_1) + \mathbf{S^2}(\theta_2) + \cdots + \mathbf{S^m}(\theta_m),$$

where θ_t is the duration of the t-th stage (in terms of slots) and $\mathbf{S^t}$ represents the set of scheduled beams and the scheduled nodes of the t-th stage. Specifically, $\mathbf{S^t} = ((\hat{\mathbf{b}}_\mathbf{1}, \beta^{t,1}, p_1), (\hat{\mathbf{b}}_\mathbf{2}, \beta^{t,2}, p_2), \cdots, (\hat{\mathbf{b}}_\mathbf{B}, \beta^{t,B}, p_B))$, where $\hat{\mathbf{b}_\nu} = ((y_1^{t,\nu}, a_1^{t,\nu}), (y_2^{t,\nu}, a_2^{t,\nu}), \cdots, (y_n^{t,\nu}, a_n^{t,\nu}))$ and $\beta^{t,1}$, $\beta^{t,2}$, $\cdots$, $\beta^{t,B}$ are the beamwidths of b_1, b_2, $\cdots$, b_B, respectively. If the binary variable $y_k^{t,\nu}$ is set to 1, then, in stage $\mathbf{S^t}$, node N_k is assigned to beam b_ν and scheduled to receive data for $a_k^{t,\nu}$ number of slots. In addition, for each beam b_ν, if $\sum_{k=1}^n y_k^{t,\nu} > 0$, then $\sum_{k=1}^n y_k^{t,\nu} a_k^{t,\nu} \leq \theta_t$. In addition, $\sum_{\nu=1}^B p_\nu \leq P_{max}$, where P_{max} is the maximum power available at the PNC.

The optimal STDMA schedule of a frame which exploits dynamic beam configuration should minimize the unsatisfied demand (objective one with weight w_1) as well as the number beam direction reconfigurations (objective two with weight $w_2 < w_1$). To formulate the optimal schedule generation problem, first we define some more required variables. Let δ_t be a binary variable that indicates whether the t-th stage exists or not. f_{ij} is the angle subtended by N_i and N_j at the PNC. Finally, g_{ijx} is 1 if the angles the pairs of nodes (N_i, N_x) and (N_j, N_x) subtend at the PNC are less than or equal to the angle (N_i, N_j) subtend at the PNC, and 0 otherwise.

The optimal scheduling problem (**P1**) can be formulated as:

$$\min_{\delta_t, \theta_t, \mathbf{S^t}} \left(w_1 \sum_{i=1}^{n} \left(\left\lceil \frac{l_{N_i}}{c_{N_i}(\mathbf{p})} \right\rceil - \sum_{t=1}^{m}\sum_{\nu=1}^{B} y_i^{t,\nu} a_i^{t,\nu} \right) + w_2 \sum_{t=1}^{m} \delta_t \right) \quad (2)$$

such that

$$\sum_{i=1}^{n} y_i^{t,\nu} a_i^{t,\nu} \leq \delta_t \theta_t \;\forall\; t, b_\nu, \;\; \sum_{t=1}^{m} \delta_t \theta_t \leq T,$$
$$\sum_{\nu=1}^{B} p_\nu \leq P_{max}, \;\; (\beta^{t,\nu} \geq \beta_{min} \wedge \beta^{t,\nu} \leq \beta_{max}) \;\forall\; t, b_\nu \quad (3)$$

$$\sum_{\nu=1}^{B} y_i^{t,\nu} \begin{cases} \leq 1, & \text{if } l_{N_i} > 0 \\ = 0, & \text{if } l_{N_i} = 0 \end{cases} \forall\; i, t, b_\nu \quad (4)$$

$$y_j^{t,\nu} = 0, \text{if } y_i^{t,\nu} > 0 \text{ and } f_{ij} \geq \beta^{t,\nu}, \ \forall\ i, j, t, b_\nu \tag{5}$$

$$\sum_{\nu_1=1,\ \nu_1 \neq \nu}^{B} y_x^{t,\nu_1} = 0, \text{if } y_i^{t,\nu} > 0, y_j^{t,\nu} > 0, \text{and } g_{ijx} = 1 \quad \forall\ i, j, x, t, b_\nu \tag{6}$$

$$\sum_{t=1}^{m} \sum_{\nu=1}^{B} y_i^{t,\nu} a_i^{t,\nu} \begin{cases} \geq 0,\ \leq \left\lceil \frac{l_{N_i}}{c_{N_i}(\mathbf{p})} \right\rceil, & \text{if } l_{N_i} > 0 \\ = 0, & \text{if } l_{N_i} = 0 \end{cases} \forall\ i. \tag{7}$$

Condition (3) indicates that, in a stage, the total number of slots allocated to the nodes located in any scheduled beam should not exceed the duration of the stage, and the total number of available slots is T. It also indicates that the total power allocated to all beams cannot be more than the power available at the PNC, and the beamwidth of each beam should be within the range $[\beta_{min}, \beta_{max}]$. Conditions (4) to (6) ensure that, in a stage, the sets of nodes located in any two beams are disjoint. Condition (6) indicates that, in stage t, if N_i and N_j are located in beam b_ν, then the nodes that are in the sector formed by N_i, N_j, and PNC cannot not be located in other beams. Condition (7) indicates that a node is allocated bandwidth only if its traffic demand is more than zero and the allocated bandwidth may span over multiple stages.

Problem (**P1**) is a mixed integer nonlinear programming problem because of the second order terms involving both binary and integer variables in the objective function and in Conditions (3) and (7), and such problems are NP-hard in general. Using relaxation techniques such as the reformulation-linearization technique [26], we can linearize these nonlinear terms. Then, Problem **(P1)** can be solved using lexicographic methods by assigning priorities to the objectives in (2). The optimization softwares take a long computational time to solve such problems, which is not suitable for practical networks with small frame durations [18]. Thus we develop a heuristic scheduling algorithm to obtain a schedule in real-time.

A. Heuristic STDMA Scheduling Method

With the proposed STDMA scheduling method, STDMA-P ("P" stands for "Proposed"), the stages that constitute the schedule of a frame are generated one by one. At the beginning of each stage generation, all beams are in the unscheduled state. During each stage generation, the beams are configured in such directions so that the pending demand is evenly distributed among all scheduled beams.

To generate the first stage, S^1, the cumulative demand (θ_{cum}) of the network is computed. To address this demand quickly, ideally, each beam should address $\lceil \frac{\theta_{cum}}{B} \rceil$ demand. To balance the load among scheduled beams, we propose the following algorithm for beam direction configuration:

- **Step-I:** Pick an unscheduled beam b and set its beamwidth as: $\beta = \beta_{min}$.
- **Step-II:** Assuming antenna beamwidth is β, find N^D, IR(N), and IN(N), $\forall\ N$; then find the node N_1 with the highest pending demand (L_1) in its IR.
- **Step-III:** In case $L_1 \geq \frac{\theta_{cum}}{B}$ or β has reached β_{max}, then configure the direction of beam b as N_1^D, assign all nodes in IN(N_1) to b, set the transmission power of b, p_b, appropriately, decrement P_{max} by p_b, and switch to Step-I to configure the direction of the next unscheduled beam if $P_{max} > 0$. Otherwise, increment β by 1^o and execute Steps II and III.

Algorithm 1 STDMA-P Scheduling Algorithm

BEGIN:

1: T is the total available slots; $N_p = \{N_1, N_2, \cdots, N_n\}$; $t = 1$;
2: **while** (T > 0 and $N_p \neq \phi$) **do**
3: $S^t = \phi$; $N_{temp} = N_p$; $\theta_t = 0$; $\theta_{cum} = \sum_{N \in N_{temp}} l_N$; $\theta_{pb} = \lceil \theta_{cum}/B \rceil$; Initialize P_{max};
4: **for** all $b = 1\ to\ B\ and\ N_{temp} \neq \phi$ **do**
5: $\beta = \beta_{min}$; $N_i = \phi$;
6: **while** (1) **do**
7: Find the IRs of nodes in N_{temp} and the cumulative demands of their INs;
8: Find node N_j whose IN does not interfere with the already scheduled beams and the cumulative demand of nodes in IN(N_j) is the highest among all nodes; Go to Line 15 provided N_j is empty;
9: **if** $(L_j < \theta_{pb}$ and $\beta < \beta_{max})$ **then**
10: $\beta \mathrel{+}= 1^o$;
11: **else**
12: $N_i = N_j$; Find, p_b, the power required to transmit to farthest node in IN(N_i) with the least possible rate; p_b = $\min(p_b, P_{max})$; compute cumulative demand in terms of slots (L_i) of IN(N_i); break;
13: **end if**
14: **end while**
15: **if** $(\theta_t == 0 \mid\mid \theta_t > L_i)$ **then**
16: $\theta_t = \min{(L_i, T)}$;
17: **end if**
18: $S^t = S^t \cup \{(\text{IN}(N_i), \beta, p_b)\}$; $N_{temp} = N_{temp} \setminus \text{IN}(N_i)$; P_{max} $- = p_b$; break if $P_{max} == 0$;
19: **end for**
20: $T = T - \theta_t$; Generate pairings for (S^t, θ_t); $t = t + 1$; Update N_p, the set of nodes with pending demands;
21: **end while**

END;

Steps I to III are repeatedly executed until enough power is not available to schedule a beam or all beams are configured successfully or no more beams can be configured without causing interference to already configured beams. Now, the least cumulative demand (θ_{min}) among the sectors of all scheduled beams is identified and the duration of $\mathbf{S^1}$ is set as $\theta_1 = \min{(T, \theta_{min})}$, and the number of free slots available in the frame is then updated as $T = (T - \theta_1)$. Since some of the scheduled beams may have more than one node in their sectors, we have to generate a schedule for the nodes located in each scheduled beam to utilize the stage duration. To achieve this, the pairing generation process explained below is used to generate {node, reception time} pairings. The stage generation process described above is repeated until either all demands are addressed successfully or no more free slots are left. The pseudo-code for STDMA-P is shown in Algorithm 1. The worst-case complexity of the algorithm is $O(TB(\beta_{max} - \beta_{min})n)$.

Pairing Generation: To generate the pairings of a stage $\mathbf{S^t}$, the beams scheduled in $\mathbf{S^t}$ are processed one by one. At the start of the processing of each scheduled beam b_ν, a temporary variable $\hat{\theta}$ is initialized with the stage duration. Until $\hat{\theta} > 0$, the nodes assigned to b_ν are picked in the increasing order of their demands and scheduled for data reception. After scheduling each node N, $\hat{\theta}$ is decremented by the number of slots allocated to N, to reflect the number of remaining free slots that can be allocated to the nodes assigned to b_ν.

Step-1: To illustrate the pairing generation process, consider stage $\mathbf{S^1}$ with duration θ_1 as an example. We start with an arbitrary beam b_1 scheduled in $\mathbf{S^1}$ and set $\hat{\theta}$ as $\hat{\theta} = \theta_1$. Now, from the nodes assigned to b_1, we pick the node N_m with the smallest demand and set its reception duration ($a^{1,b_1}_{N_m}$) as $\min(\lceil l_{N_m}/c_{N_m}(\mathbf{p})\rceil, \hat{\theta})$. Then, l_{N_m} and $\hat{\theta}$ are updated as $l_{N_m} = \max(0, (l_{N_m} - c_{N_m}(\mathbf{p})a^{1,b_1}_{N_m}))$ and $\hat{\theta} = (\hat{\theta} - a^{1,b_1}_{N_m})$. If $\hat{\theta} > 0$, then the other nodes assigned to b_1 are scheduled in the increasing order of their demands until $\hat{\theta}$ becomes 0. The same process is repeatedly executed with the other scheduled beams, to generate the complete list of pairings of $\mathbf{S^1}$.

Step-2: After processing all scheduled beams, the cumulative residual demand of some of the scheduled beams might get addressed completely which makes these beams idle. To reconfigure these idle beams (with beamwidth β_{min}) towards the nodes with pending demands and then to extend the duration of $\mathbf{S^1}$, STDMA-P executes the *stage extension* process explained below, provided some free bandwidth is available in the current frame, that is, $T > 0$ and $P_{max} > 0$. Until $P_{max} > 0$, idle beams are picked one by one and configured in direction N_e^D, if N_e has the least interference among all unassigned nodes whose INs are disjoint with the set of nodes assigned to the non-idle beams of $\mathbf{S^1}$ and P_{max} is decremented by the power assigned to configured idle beam. If at least one idle beam gets scheduled in this process, then θ_{min}, the least cumulative pending demand among all non-idle beams, is computed and the extended duration of $\mathbf{S^1}$ is set as $\theta_1 = \min(T, \theta_{min})$. Then the number of free slots in the frame is updated as $T = (T - \theta_1)$, and new pairings are generated as explained in Step-1. In case none of the idle beams gets scheduled, then the pairing generation process is concluded. Each beam direction reconfiguration involves a training overhead. For example, when the number of antenna elements is small (e.g., 4-10), the feasible training overhead is 50 symbols [27]. The stage extension process reduces the beamforming overhead by scheduling only idle beams.

VI. Analysis

In this section, we obtain the upper and lower bounds on the expected number of beams scheduled in a stage and the approximation ratio of STDMA-P. Intuitively, the expected number of scheduled beams is the highest when the PNC forms all beams with the least possible antenna beamwidth β_{min}. Thus, to obtain the upper bound, we set $\beta = \beta_{min}$ and to obtain the lower bound we set $\beta = \beta_{max}$. The procedure to obtain both the bounds is the same as explained below. Consider a network that consists of n nodes. Assuming non-uniform random distribution of nodes, the number of nodes covered by the i-th scheduled beam is represented as α_i. To form a beam in direction N^D, IR(N) should not overlap with the already scheduled beams. Let R be the probability that IR(N) does not overlap with a scheduled beam b. IR(N) may overlap with beam b, if node N is in either of the sectors (of beamwidth β) that are adjacent to the sector of beam b. Hence, the area of the possible overlapping region is: $A_o = \pi\gamma^2(2\beta/2\pi)$, where γ is the transmission range of a beam. Assuming uniform coverage range for all beams, the coverage area of the PNC is: $A = \pi\gamma^2$. Thus the probability that IR(N) does not overlap with beam b is: $R = (1 - A_o/A)$, that is, $R = (1 - \beta/\pi)$.

Let $P(x, y)$ be the probability of successfully configuring x beams after checking y beams. STDMA-P forms the first beam with probability 1, hence we have, $P(1, 1) = 1$. If the first beam gets scheduled successfully, then STDMA-P considers the second beam, which can be scheduled successfully provided there exists at least one node whose IR does not overlap with the already scheduled beams. The number of nodes covered by the first beam is α_1. The probability that the IRs of all remaining $(n-\alpha_1)$ nodes to have some overlap with the first beam is: $(1 - R)^{(n-\alpha_1)}$. Hence, the probability that the second beam is not scheduled, in other words, the probability of scheduling only one out of the B available beams[1] is: $P(1, j) = P(1, B) = P(1, 1)(1 - R)^{n-\alpha_1}$, for $2 \leq j \leq (B - 1)$. And, the probability of successfully scheduling two beams after checking two beams is: $P(2, 2) = P(1, 1)(1 - (1 - R)^{n-\alpha_1})$.

Generalizing the beam formation process to the i-th beam, the PNC considers the i-th beam for scheduling if and only if the first $(i - 1)$ beams are scheduled successfully. The i-th beam can be scheduled successfully provided there exists at least one node whose IR does not overlap with the already scheduled $(i-1)$ beams. Hence, the probability of not scheduling the i-th beam, in other words, the probability of scheduling only $(i - 1)$ beams out of B beams and $P(i, i)$ are:

$$P(i-1, B) = P(i-1, i-1)\Big(1 - R^{(i-1)}\Big)^{n-\sum_{j=1}^{i-1}\alpha_j};$$
$$P(i, i) = P(i-1, i-1)\left(1 - \Big(1 - R^{(i-1)}\Big)^{n-\sum_{j=1}^{i-1}\alpha_j}\right). \tag{8}$$

Then, the expected number of scheduled beams is: $E[\varrho] = \sum_{i=1}^{B} iP(i, B)$. By iteratively obtaining $P(i, B)$, $1 \leq i \leq B$, we can get $E[\varrho]$ as:

$$E[\varrho] = \sum_{i=1}^{B} i\left(\prod_{j=0}^{i-1}\Big(1-(1-R^j)^{n-\sum_{k=1}^{j}\alpha_k}\Big)\right)(1-R^i)^{n-\sum_{k=1}^{i}\alpha_k}. \tag{9}$$

The optimal scheduler schedules B' beams, where $B' \leq B$. Hence, the approximation ratio of STDMA-P (ζ) is: $\zeta = E[\varrho]/B'$, $\quad \zeta \geq E[\varrho]/B$. By substituting $E[\varrho]$ in ζ, we get

$$\zeta \geq \frac{1}{B}\sum_{i=1}^{B} i\left(\prod_{j=0}^{i-1}\Big(1-(1-R^j)^{n-\sum_{k=1}^{j}\alpha_k}\Big)\right)(1-R^i)^{n-\sum_{k=1}^{i}\alpha_k}. \tag{10}$$

$E[\varrho]$ is computed while assuming that the PNC has enough power to communicate with all available beams. However, in practical scenarios, the power of PNC is limited, thus the upper bound on the number of busy beams is: $\min(\lceil P_{max}/p_{min}\rceil, E[\varrho])$, where p_{min} is the minimum transmission power required by any node in the network and $E[\varrho]$ is computed by setting β as β_{min}. Similarly, the lower bound is:

[1] If it is not possible to schedule the second beam, then the other beams also cannot be scheduled.

$\min(\lceil P_{max}/p_{max}\rceil, E[\varrho])$, where p_{max} the maximum transmission power required by any node and $E[\varrho]$ is computed by setting β as β_{max}.

VII. MB^{2}SW-FS: Maximize Busy Beams using Swapping Window Fair Scheduler

STDMA-P targets to maximize the number of busy beams while minimizing the number of beam direction reconfigurations, but it does not address fair resource allocation. In this section, we propose a service tag based fair scheduler, MB^{2}SW-FS, which achieves fair resource allocation by *ensuring service to the flow that receives the least service at any time t*. MB^{2}SW-FS implements dynamic beam direction adjustment in addition to dynamic beam configuration, to maximize the fair resource allocation to each flow, and minimizes the control overhead by allowing nodes to be active in the scheduled beams for more than one slot.

G denotes the set all backlogged flows. For flow f, the PNC is one end point, and the other end point of f is denoted as N_f. To communicate with N_f, the PNC identifies a data transmission direction, N_f^d, which is the direction of the vector pointing from the PNC to N_f. The vector drawn in direction N_f^d is denoted as $\overrightarrow{N_f^d}$. Function $U(\overrightarrow{d_1}, \overrightarrow{d_2})$ returns the angle $\overrightarrow{d_1}$ and $\overrightarrow{d_2}$ subtend at the PNC, and function $\widetilde{U}(\overrightarrow{d_1})$ returns the angle subtended by $\overrightarrow{d_{ref}}$ and $\overrightarrow{d_1}$ at the PNC in the anti-clockwise direction, where $\overrightarrow{d_{ref}}$ is the vector that starts at the PNC and ends at point $(\gamma, 0)$.

The PNC has a very limited information about the uplink flows, compared to the downlink flows. Thus we explain our proposed fair scheduler while considering uplink flows. However, by making appropriate changes, it can also be used to schedule downlink flows. We assume that in each slot, one packet of size L bytes can be transmitted. Each flow f is assigned with a weight ω_f, and flow f can be in one of the two states: *backlogged* or *idle*. The schedule generated by MB^{2}SW-FS consists of a number of stages, similar to that of STDMA-P. In each scheduled beam of stage $\mathbf{S^t}$, only one node is active for θ_t number of slots, which is in contrast to STDMA-P, where the stage duration may be utilized by multiple nodes. Thus, the structure of a stage $\mathbf{S^t}$ is nothing but a list of pairings: $\mathbf{S^t} = (\mathbf{\hat{b}_1}, \mathbf{\hat{b}_2}, \cdots, \mathbf{\hat{b}_B})$, where, $\mathbf{\hat{b}_1} = (f_1, \theta_t)$, $\mathbf{\hat{b}_2} = (f_2, \theta_t)$, $\cdots$, $\mathbf{\hat{b}_B} = (f_B, \theta_t)$, where f_1, f_2, $\cdots$, f_B represent the flows scheduled for data transmission in beams b_1, b_2, $\cdots$, b_B, respectively.

A. Tag Assignment

In the RAP of each frame, the PNC receives the transmission requests of the flows with pending traffic demands and converts these requests into a number of slot requests based on the transmission rates of the corresponding links. To define the order in which these slot requests should be served to maintain fairness among the flows, each slot request is assigned two service tags, which are defined based on the virtual time of the network that is maintained using the SFQ model [9]. Initially, the virtual time is set as 0, and it is maintained only for DTPs, thus there will be no change in it during BAPs, RAPs, and BPs. At time t, the virtual time is set as the minimum start tag among all flows with pending slot requests.

To reduce the control overhead involved in keeping track of the service tags of slot requests, the PNC maintains two service tags for each flow f: a *start tag* $(I(f))$ and a *finish tag* $(F(f))$, which are nothing but the start and finish tags of the first unserved slot request of f, respectively. Two additional tags ($I^{'}(f)$, $F^{'}(f)$) are maintained for each flow f to keep track of the start and finish tags of the first slot request that is yet to be scheduled. Assume that the PNC receives the traffic demand of f, μ_f, at time $A(t)$. The PNC converts μ_f into slot requests, sets *the pending slot requests counter,* $\mu_f^{'}$, as: $\mu_f^{'} = \lceil \mu_f/L \rceil$, and updates the tags of flow f as follows:

- If flow f is in idle state, then its state is changed to backlogged and its tags are set as:
$$I(f) = I^{'}(f) = V(A(t)), F(f) = F^{'}(f) = I(f) + \frac{L}{\omega_f}, \tag{11}$$
where $V(A(t))$ is the virtual time at time $A(t)$.
- If flow f is in backlogged state, then there will be no change in its start and finish tags.

In a RAP, if the PNC does not receive a transmission request from flow f which is in backlogged state, then the PNC updates the state of f as idle, resets $I(f)$, $F(f)$, $I^{'}(f)$, and $F^{'}(f)$ as ∞, and $\mu_f^{'}$ as 0.

To generate the schedule of a frame, MB^{2}SW-FS processes the pending slot requests in a number of rounds. In each round, it executes the processes explained in Sections VII-B (*fair allocation*) and VII-C (*spatial reuse beyond swapping window*). In some of the rounds, MB^{2}SW-FS may generate multiple stages, but it finalizes/outputs only one stage in each round. Thus, to carry forward the stages from one round to the next, MB^{2}SW-FS maintains the pending stages in Θ. For each stage $\mathbf{S^t} \in \Theta$, MB^{2}SW-FS maintains two tags, $I(\mathbf{S^t})$ and $F(\mathbf{S^t})$ which represent the least start and finish tags among all slot requests scheduled in $\mathbf{S^t}$.

B. Fair Allocation

In WPANs with MBAAs, in each slot all B beams can receive data simultaneously. To achieve fair allocation, the slot requests should be served in the increasing order of their start tags. However, if we schedule the flows to be strictly fair, then, in the worst case, only one slot request with the least start tag is served in each slot, which leads to the under utilization of the capability of MBAAs. To prevent this, at any time t, instead of a single slot request with the least start tag, a set of slot requests that have lower start tags than others are considered for scheduling [10]. A swapping window is defined in terms of virtual time as $[V(t), V(t)+\sigma]$, where $\sigma = \sigma_1(\max_{f \in G} \lceil \frac{L}{\omega_f} \rceil)$, where σ_1 is a constant. All slot requests with start tags within the window are eligible for scheduling, and are thus called the *eligible slot requests*. We can swap the service order of the eligible slot requests to increase the number of active beams in each slot. In the place of individual eligible slot requests, their respective flows called *eligible flows* are scheduled in stages, and the stage duration specifies the number of eligible slot requests scheduled. Whenever MB^{2}SW-FS generates a new stage to accommodate eligible slot requests, it sets the duration of that stage as ∞ or undefined so as to accumulate the slot requests of the flows that could be scheduled in the upcoming rounds and to generate stages that last for multiple slots.

We can find two types of stages in Θ, the stages that explicitly specify the scheduled flows and slot requests, that is, the stages whose duration is defined, denoted by a set $\Theta^{'} \subseteq \Theta$, and the stages that specify *only* the scheduled flows, that is, the stages whose duration is undefined or ∞, denoted by a set $\Theta^{''} = (\Theta \backslash \Theta^{'})$. To make this more clear, consider the scenario where $|\Theta| = 0$, that is, a network that just came into existence. Based on the swapping window, the eligible slot requests are identified and one or more new stages with undefined duration are generated to schedule these eligible slot requests. Then, the stage $\mathbf{S^1} \in \Theta^{''}$ with the least finish tag is extracted, and its duration is set as, say x slots, where $x \geq 1$. Setting θ_1 as x slots may lead to unfair resource utilization, since irrespective of their weights, we are explicitly scheduling x number of slot requests of each flow that is scheduled in $\mathbf{S^1}$. To restore fairness, the flow f_{max} that has received the highest share of channel allocation is identified, and compared to f_{max}, the extra share of allocation (in terms of slots) to be received by each other flow f is computed as its *pending fair share* (φ_f). Then, φ_f number of slot requests of f are explicitly scheduled either in the existing stages or in new stages, which generates a number of stages in $\Theta^{'}$. At this juncture, we can have both types of stages in Θ. For each $\mathbf{S^t} \in \Theta^{'}$, $I(\mathbf{S^t}) < I(\mathbf{S^u})$, $\forall \mathbf{S^u} \in \Theta^{''}$.

Fair allocation works in four phases. In Phase-I, the eligible slot requests are identified and scheduled. Phase-II finalizes the stage $\mathbf{S^c}$ that accommodates the slot request with the least finish tag. If θ_c is undefined, then it is fixed first. Then, to restore fairness, pending fair shares are computed and accommodated in the existing or in new stages in Phase-III. In Phase-IV, the duration of $\mathbf{S^c}$ is possibly extended, provided enough bandwidth is available to serve all pending fair shares. Then, the unscheduled/idle beams of $\mathbf{S^c}$ are possibly scheduled by exploiting swapping beyond window. The detailed working process of each phase is explained below.

Phase-I (Generate stages with eligible flows): If f is a flow such that $I^{'}(f)$ is within the window $[V(t), V(t)+\sigma]$, then the number of eligible slot requests of f are: $\lambda_f = \lceil \frac{(V(t)+\sigma-I^{'}(f))\omega_f}{L} \rceil + 1$. If f is already scheduled in a stage $\mathbf{S^t} \in \Theta^{''}$, then nothing else needs to be done. Otherwise, flow f is possibly scheduled in one of the existing stages with unscheduled beams, $\mathbf{S^1}$, following the process explained in Step-I and II, which is also given in Algorithm 2.

Step-I: Lines 2-4 check whether forming a beam in direction N_f^d causes any interference to the already scheduled flows of $\mathbf{S^1}$ or not. If not, then the direction of an unscheduled beam of $\mathbf{S^1}$, b_{temp}, is set as N_f^d in line 5. Otherwise, beam direction adjustment (BDA) is explored in Step-II.

Step-II: Assume that $\Omega_1^{'} \subseteq \Omega_1$ is the set of beams that cover the nodes interfered by forming a beam in direction N_f^d. In the BDA process, the beams in $\Omega_1^{'}$ are picked one by one, and the direction of the picked beam is adjusted so that the chances of configuring an extra non-overlapping beam is maximized. For

Algorithm 2 Algorithm for steps I and II of Phase-I

BEGIN:
1: b_{ol} is the overlapping region between two adjacent beams; schd $= 0$;
2: **if** $\exists\ b_k \in \Omega_1$ such that $(U(\overrightarrow{N_f^d}, \overrightarrow{N_{f_k}^d}) \leq (\beta/2 + b_{ol}))$ **then**
3: Go to Line 6;
4: **end if**
5: set b_{temp}^d as N_f^d; push b_{temp} onto Ω_1; schd$=1$; Go to line 19;
6: **if** $U(\overrightarrow{N_f^d}, \overrightarrow{N_{f_i}^d}) > b_{ol}$, $\forall\ b_i \in \Omega_1$ **then**
7: Find $\Omega_1^{'} \subseteq \Omega_1$ such that for each $b \in \Omega_1^{'}$, b is available for BDA and $(U(\overrightarrow{N_f^d}, \overrightarrow{b^d}) \leq (\beta/2 + b_{ol}))$;
8: **end if**
9: **for** $i = 1$ to $|\Omega_1^{'}|$ **do**
10: **if** N_f falls on the right of $\overrightarrow{b_i^d}$ **then**
11: $b_{temp}^d = (\widetilde{U}(\overrightarrow{b_i^d}) - b_{ol} - \frac{\beta}{2})$; $b_i^{nd} = (\widetilde{U}(\overrightarrow{b_i^d}) - b_{ol} + \frac{\beta}{2})$;
12: **else**
13: $b_{temp}^d = (\widetilde{U}(\overrightarrow{b_i^d}) + b_{ol} + \frac{\beta}{2})$; $b_i^{nd} = (\widetilde{U}(\overrightarrow{b_i^d}) + b_{ol} - \frac{\beta}{2})$;
14: **end if**
15: **if** $(U(\overrightarrow{b_{temp}^d}, \overrightarrow{N_{f_j}^d}) > (\beta/2 + b_{ol}) \wedge U(\overrightarrow{b_i^{nd}}, \overrightarrow{N_{f_j}^d}) > (\beta/2 + b_{ol}))$, $\forall\ b_j \in \Omega_1$ **then**
16: $b_i^d = b_i^{nd}$; Include b_{temp} in Ω_1; schd $= 1$; break;
17: **end if**
18: **end for**
19: **if** schd$==1$ **then**
20: Update $I^{'}(f)$ as $I^{'}(f) \mathrel{+}= \min(\theta_1, \lambda_f)\frac{L}{\omega_f}$, and $F^{'}(f)$ as $F^{'}(f) \mathrel{+}= \min(\theta_1, \lambda_f)\frac{L}{\omega_f}$, provided θ_1 is not ∞;
21: $\lambda_f = \max(0, (\lambda_f - \theta_1))$; push (f, θ_1) onto $\mathbf{S^1}$;
22: **end if**
END;

any $b_i \in \Omega_1$, if $U(\overrightarrow{b_i^d}, \overrightarrow{N_f^d}) = b_{ol}$, as shown in Fig. 3(a), then in whatever way we adjust b_i^d (one possible way is shown in Fig. 3(b)), it is not possible to cover N_f with another beam without causing interference to N_{f_i}, where N_{f_i} is the end node of the flow scheduled in b_i. Similar is the case when $U(\overrightarrow{b_i^d}, \overrightarrow{N_f^d}) < b_{ol}$. Based on this analysis, MB^{2}SW-FS checks whether BDA is possible or not in line 6. If so, then it finds $\Omega_1^{'}$ in line 7. A beam $b_i \in \Omega_1^{'}$ is picked and depending upon whether N_f is on the right or left of $\overrightarrow{b_i^d}$, as shown in Fig. 3(c) and (d), the direction an unscheduled beam b_{temp} and the new direction of b_i, b_i^{nd}, are set in lines 10-14. These directions are adjusted in such a way that N_{f_i} is at the border of the sector formed by b_i, but not in the overlapping region of b_i and b_{temp}. If none of the flows scheduled in $\mathbf{S^1}$ gets interfered by forming beams in directions b_{temp}^d and b_i^{nd}, then b_i^d is updated as b_i^{nd}, and b_{temp} is included in Ω_1 in lines 15-17, and b_i and b_{temp} are marked as unavailable for BDA in $\mathbf{S^1}$. Otherwise, the other beams in $\Omega_1^{'}$ are tested similarly. If MB^{2}SW-FS succeeds to schedule f in $\mathbf{S^1}$, then the tags of f, λ_f, and the pairing list of $\mathbf{S^1}$ are updated in lines 19-22.

The other stages with unscheduled beams are processed similarly, and f is possibly scheduled in these stages, if $\lambda_f > 0$. After processing all existing stages, if λ_f is still more than zero, then a new stage $\mathbf{S^u}$ is generated, the direction of a beam b is set as N_f^d, and b is included in Ω_u. $I(\mathbf{S^u})$ and $F(\mathbf{S^u})$ are set as $I(f)$ and $F(f)$, respectively. θ_u is set as ∞, a pairing (f, θ_u) is pushed onto $\mathbf{S^u}$, and $\mathbf{S^u}$ is included in Θ. The process explained in this phase is repeatedly executed until all eligible slot requests are scheduled in some stage.

Phase-II (Finalize one stage): The stage finalization process is given in Algorithm 3. The stage $\mathbf{S^c} \in \Theta$ with the least finish tag is extracted in line 1. Assume that $\mathbf{S^c} \in \Theta^{'}$. If the

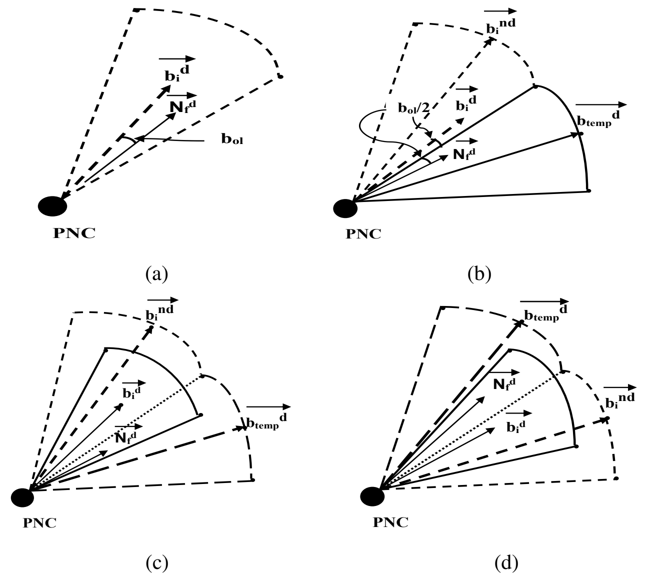


Fig. 3: Examples of beam direction adjustment.

Algorithm 3 Algorithm for Stage Finalization

BEGIN:
1: Extract $\mathbf{S^c} \in \Theta$, the stage with the least finish tag; $x = 0$;
2: **if** $\mathbf{S^c} \in \Theta^{'}$ **then**
3:   **if** $\theta_c > T$ **then**
4:     Generate a copy of $\mathbf{S^c}$, $\mathbf{S^{c_2}}$; $\theta_{c_2} = \theta_c - T$; $\theta_c = T$; $\Theta = \Theta \cup \mathbf{S^{c_2}}$;
5:   **end if**
6: **else**
7:   Find $\mu^{'}_{min}$, the least residual demand among the flows scheduled in $\mathbf{S^c}$; $\theta_c = \min(T, \sigma_1, \mu^{'}_{min})$;
8:   $I^{'}(f) \mathrel{+}= \theta_c \frac{L}{\omega_f}$, $F^{'}(f) \mathrel{+}= \theta_c \frac{L}{\omega_f}$, for all f scheduled in $\mathbf{S^c}$; $x = 1$;
9: **end if**
10: $T = T - \theta_c$;
11: $I(f) \mathrel{+}= \min(\mu^{'}_f, \theta_c) \frac{L}{\omega_f}$, $F(f) \mathrel{+}= \min(\mu^{'}_f, \theta_c) \frac{L}{\omega_f}$, $\mu^{'}_f = \max(0, (\mu^{'}_f - \theta_c)$, for all f scheduled in $\mathbf{S^c}$;
12: **if** $x == 1$ **then**
13:   Find f_{max} with the highest start tag; Set $\varphi_g = \min(\mu^{'}_g, \lceil \frac{(I(f_{max}) - I(g))\omega_g}{L} \rceil)$, $\forall\, g \in G \setminus \{f_{max}\}$;
14:   $I^{'}(g) \mathrel{+}= \varphi_g \frac{L}{\omega_g}$, $F^{'}(g) \mathrel{+}= \varphi_g \frac{L}{\omega_g}$, $\forall\, g \in G \setminus \{f_{max}\}$;
15: **end if**
END;

---

available free slots are not enough to serve all slot requests scheduled in $\mathbf{S^c}$, that is, $T < \theta_c$, then a copy of $\mathbf{S^c}$, $\mathbf{S^{c_2}}$, is generated with the slot requests which cannot be served in $\mathbf{S^c}$ in line 4 and then $\mathbf{S^{c_2}}$ is included in Θ, in the hope that these slot requests will be served in the upcoming frame. Then, T is updated in line 10, and the tags and pending demands of the flows scheduled in $\mathbf{S^c}$ are updated appropriately, in line 11.

In case $\mathbf{S^c} \in \Theta^{''}$, then its duration is set in line 7. We are restricting the duration of each stage to be at most σ_1, to limit the unfairness among the flows. T and then the parameters of the scheduled flows are updated in lines 10 and 11. The flow f_{max} that has received the highest share of channel allocation is identified. Then, compared to f_{max}, the extra share of channel allocation to be received by each other flow g is computed in line 13, and $I^{'}(g)$ and $F^{'}(g)$ are also updated in line 14, since in Phase-III, these pending fair shares will definitely be accommodated either in the existing stages or in new stages.

Algorithm 4 Algorithm for Extension of a Stage

BEGIN:
1: $\mu^{'}_{min}$ be the minimum residual demand among the flows scheduled in $\mathbf{S^c}$;
2: $\theta^{+} = \min((\sigma_1 - \theta_c), T^{'}, \mu^{'}_{min})$; $\theta_c \mathrel{+}= \theta^{+}$;
3: $I(f) \mathrel{+}= \theta^{+}(\frac{L}{\omega_f})$, $F(f) \mathrel{+}= \theta^{+}(\frac{L}{\omega_f})$, $I^{'}(f) \mathrel{+}= \theta^{+}(\frac{L}{\omega_f})$, $F^{'}(f) \mathrel{+}= \theta^{+}(\frac{L}{\omega_f})$, $\mu^{'}_f \mathrel{-}= \theta^{+}$, for all f scheduled in $\mathbf{S^c}$; $T \mathrel{-}= \theta^{+}$;
4: **if** $|\Omega_c| < B$ **then**
5:   From the flows with pending slot requests, find the flow f with the least $I^{'}(f)$;
6:   Find G_1, the set of flows with pending slot requests and whose $I^{'}(.)$ falls within the window $[I^{'}(f), I^{'}(f) + \sigma]$;
7:   **while** $|\Omega_c| < B$ and $|G_1| > 0$ **do**
8:     Pick the flow $g \in G_1$ with the least $F^{'}(.)$ value; Execute steps I and II of Phase-I to schedule g in $\mathbf{S^c}$;
9:     **if** g gets scheduled in $\mathbf{S^c}$ **then**
10:       $I(g) \mathrel{+}= \min(\mu^{'}_g, \theta_c) \frac{L}{\omega_g}$; $F(g) \mathrel{+}= \min(\mu^{'}_g, \theta_c) \frac{L}{\omega_g}$;
11:       $I^{'}(g) \mathrel{+}= \min(\mu^{'}_g, \theta_c) \frac{L}{\omega_g}$; $F^{'}(g) \mathrel{+}= \min(\mu^{'}_g, \theta_c) \frac{L}{\omega_g}$; $\mu^{'}_g = \max(0, (\mu^{'}_g - \theta_c)$;
12:     **end if**
13:   **end while**
14: **end if**
END;

Phase-III (Address pending fair shares): To accommodate the pending fair shares in the existing stages, for each stage $\mathbf{S^{t_1}} \in \Theta^{''}$, θ_{t_1} is set as: $\theta_{t_1} = \min(\sigma_1, \varphi_{f_1})$, where φ_{f_1} is the minimum pending fair share among the flows scheduled in $\mathbf{S^{t_1}}$. For each flow f scheduled in $\mathbf{S^{t_1}}$, φ_f is updated as: $\varphi_f = \varphi_f - \theta_{t_1}$. Then, each flow g with non-zero φ is tested for scheduling in each stage in Θ (following the process explained in Step-I and II of Phase-I). If g gets scheduled in $\mathbf{S^{t_2}} \in \Theta$, then φ_g is updated as $\max(0, (\varphi_g - \theta_{t_2}))$. Then, until $\varphi_g > 0$, a new pairing $\mathbf{S^u}$ is generated and the direction of a beam in $\mathbf{S^u}$ is set as N^d_g; θ_u, $I(\mathbf{S^u})$, and $F(\mathbf{S^u})$ are set as $\min(\sigma_1, \varphi_g)$, $I(g)$, and $F(g)$, respectively, and φ_g is decremented by θ_u; finally, $\mathbf{S^u}$ is included in Θ.

Phase-IV (Fair extension of the current stage): In this phase, MB^{2}SW-FS tries to extend the duration of $\mathbf{S^c}$ first, and then possibly schedules the unscheduled beams of $\mathbf{S^c}$, following Algorithm 4. If the remaining bandwidth in the current frame is more than enough to serve all stages in $\Theta^{'}$ that accommodate the pending fair shares of flows, and the duration of $\mathbf{S^c}$ not yet reached the threshold on the stage duration, that is, $\theta_c < \sigma_1$, then MB^{2}SW-FS extends the duration of $\mathbf{S^c}$, otherwise not. Let $T^{'}$ be the number of slots that will remain free after serving all stages in $\Theta^{'}$. Then the possible extension of θ_c is computed and θ_c is updated in line 2 of Algorithm 4. The parameters of the scheduled flows and T are updated appropriately in line 3.

To schedule the unscheduled beams of $\mathbf{S^c}$, the eligible flows are identified first. From the flows with unserved slot requests, the flow f with the minimum $I^{'}(f)$ is identified in line 5, and a set, G_1, of flows whose $I^{'}(\cdot)$ fall within the window $[I^{'}(f), I^{'}(f) + \sigma]$ is identified in line 6. Now, the flows from G_1 are considered in the increasing order of their $F^{'}(\cdot)$ and possibly scheduled in $\mathbf{S^c}$, and then their tags and pending demands are updated in lines 7-13.

C. Spatial Reuse Beyond Swapping Window

To schedule the remaining unscheduled beams of $\mathbf{S^c}$, MB^{2}SW-FS considers all those flows which are not scheduled in $\mathbf{S^c}$ even though these flows do not satisfy the fairness constraint. To ensure fair allocation in this phase, the cumulative service received by each flow f in this phase is maintained in ψ_f. The flows are picked in the increasing order of ψ/ω values, and possibly scheduled in $\mathbf{S^c}$ using Algorithm 2, until all beams are scheduled successfully or all flows are tested. In case a beam b gets scheduled to serve flow f, then ψ_f is incremented by $\min(\mu'_f, \theta_c)$, b and (f, θ_c) are pushed onto Ω_c and $\mathbf{S^c}$, respectively, and μ'_f is decremented by $\min(\mu'_f, \theta_c)$.

VIII. Analytical Properties of MB^{2}SW-FS

In this section we establish the analytical properties of MB^{2}SW-FS in the form of upper and lower bounds on the fair service of flows, the maximum unfairness between any two flows, and the minimum throughput of a flow. These properties are related to the service received by each flow f in the fair allocation phase denoted as W_f.

Lemma 1. *If flow f is continuously backlogged in an interval $[t_1, t_2]$, then the fair service received by f in $[t_1, t_2]$, $W_f(t_1, t_2)$, is lower bounded by $(\omega_f(v_2 - v_1) - L)$, that is, $\omega_f(v_2 - v_1) - L \leq W_f(t_1, t_2)$, where $v_1 = V(t_1)$ and $v_2 = V(t_2)$.*

Proof: Proof is given in Appendix A. □

Lemma 2. *If flow f is continuously backlogged in an interval $[t_1, t_2]$, then the fair service received by f in that interval is upper bounded by $(\omega_f(v_2+3\sigma-v_1)+L)$, that is, $W_f(t_1, t_2) \leq \omega_f(v_2 + 3\sigma - v_1) + L$, where $v_1 = V(t_1)$ and $v_2 = V(t_2)$.*

Proof: Proof is given in Appendix B. □

From Lemmas 1 and 2, Theorem 1 follows directly.

Theorem 1. *Let f_1 and f_2 be two flows that are continually backlogged in an interval $[t_1, t_2]$. The difference in the fair service received by these flows in that interval is upper bounded by:*

$$\left| \frac{W_{f_1}(t_1, t_2)}{\omega_{f_1}} - \frac{W_{f_2}(t_1, t_2)}{\omega_{f_2}} \right| \leq (3\sigma + \frac{L}{\omega_{f_1}} + \frac{L}{\omega_{f_2}}). \quad (12)$$

Proposition 1. *For any two flows f and g that are continually backlogged in an interval [t_1, $t_1 + \Delta t$], MB^{2}SW-FS provides the same normalized fair service in the long-term, in other words,*

$$\lim_{\Delta t \to \infty} \frac{W_f(t_1, t_1 + \Delta t)}{W_g(t_1, t_1 + \Delta t)} = \frac{\omega_f}{\omega_g}. \quad (13)$$

Proof: Proposition 1 can be proved using Lemma 1. □

Proposition 2. *Let f be a flow that is continually backlogged in an interval [t_1, t_2]. MB^{2}SW-FS provides a minimum fair share of channel capacity C to flow f as given below:*

$$W_f(t_1, t_2) \geq C \frac{\omega_f}{k \sum_{g \in G} \omega_g}(t_2 - t_1) - \Delta, \quad (14)$$

where k and Δ are network dependent constants.

Proof: Proof is given in Appendix C. □

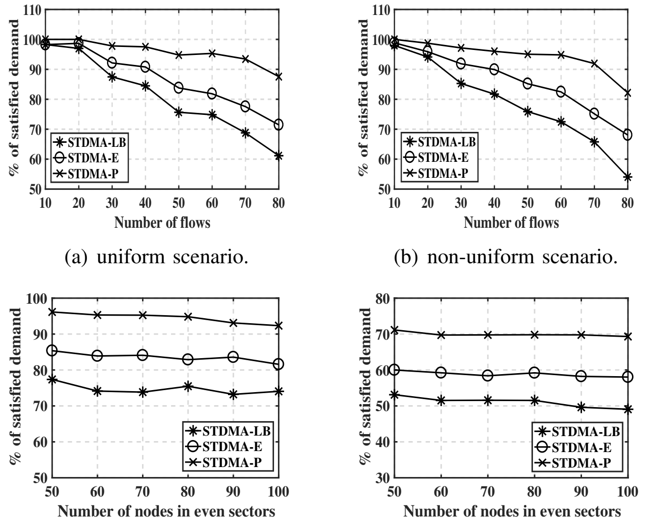
IX. Simulation Results

Performance of the proposed schedulers is evaluated using our own simulator coded in C++. Due to the increasing interest in the directional antenna based mmWave communications [28], [29], the performance evaluation is conducted in a mmWave WPAN setting for supporting indoor services with high data rates. 80 nodes are deployed in a circular region with radius 10 m and the PNC is located at the center [30]. The system bandwidth is 1200 MHz, the background noise level is -134 dBm/MHz [31], and node transmission power is 0.1 mW. P_{max} of PNC is set as 20 dBm. The path loss exponent is 2 and the path loss is 71.5 dB at the reference distance of 1.5 m [32]. The number of flows in the network is varied between 10 and 80 and flows are established between randomly selected nodes and the PNC. The bandwidth requirements of the flows are uniformly distributed between 1.5 and 3.5 Gbps.

A. STDMA-P vs Existing Methods

For this set of simulations, two node deployment scenarios are considered. In the first scenario called "uniform", nodes are uniformly deployed in the considered region. In the second scenario called "non-uniform", the circular region is divided into sectors of beamwidth $2\pi/B$ and numbered from 1 to B, and $3/4$ of the nodes are deployed in the even numbered sectors and the remaining nodes in other sectors, to create a network with non-uniform node density. The average percentage of satisfied demand and the average of number of times beam directions are changed per frame are considered as the performance metrics. The presented results are averaged over 25 simulation runs.

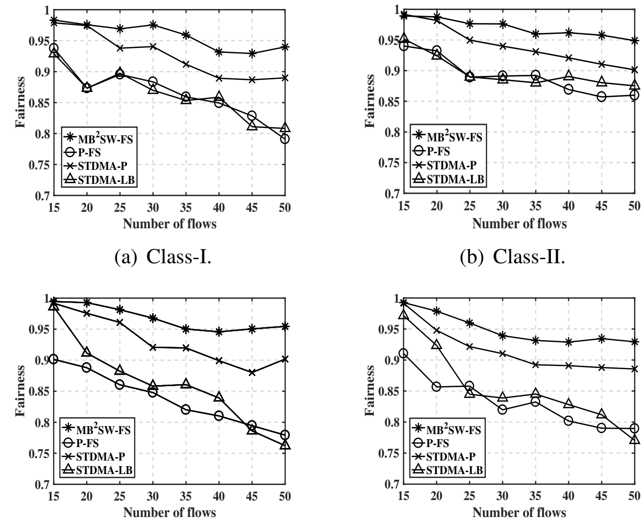
The performance of STDMA-P is compared with that of two other schedulers: *"STDMA-LB"* and *"STDMA-E"*. The beam partition method proposed in [5] is modified for our scenario and called STDMA-LB. Its working procedure is as follows. If the number of nodes with pending traffic demand is n_1, then n_{sec} is defined as: $\max((n_1/B), 1)$. Starting from the line PNC-Y shown in the left part of Fig. 1 and moving in the anti-clockwise direction, nodes are grouped into a sector until the number of nodes in the sector is at least n_{sec} or the angle subtended by the sector at the PNC is at least β. Similarly, the other sectors are formed until all nodes are assigned to some sector. Then, until unscheduled beams and power is available, a beam is configured to cover one of the formed sectors and its transmission power is set appropriately and P_{max} is decremented by the assigned power. The nodes situated in a sector are served in the increasing order of their traffic demands. STDMA-E, a variant of STDMA-P, does not perform stage extension, and forms beams such that the number of nodes in the scheduled beams is almost the same. Consider the t-th stage generation using STDMA-E, and assume that the number of nodes with pending traffic demand is n_2. STDMA-E configures the direction of an unscheduled beam b as N^D, that is, the transmission direction of node N with interference x, if $|x - (n_2/B)|$ is the lowest among all unassigned nodes whose INs are disjoint with the set of nodes assigned to already scheduled beams of stage t. The transmission power of b is set appropriately and P_{max} is updated correspondingly. Similarly,



(a) uniform scenario. (b) non-uniform scenario.

(c) Beamwidth: 10^o. (d) Beamwidth: 20^o.

Fig. 4: Number of flows vs percentage of satisfied demand.



(a) Class-I. (b) Class-II.

(c) Class-I (Mobile scenario). (d) Class-II (Mobile scenario).

Fig. 5: Number of flows vs fairness.

the other beams are scheduled until $P_{max} > 0$ and pending demands and unscheduled beams are left.

The number of busy beams is higher in the "uniform" scenario compared to the "non-uniform" scenario, for all three schedulers. Hence, as shown in Fig. 4(a) and (b), the percentage of satisfied demand is higher in the "uniform" scenario compared to the "non-uniform" scenario. However, due to dynamic beam configuration, STDMA-P and STDMA-E succeed to schedule a larger number of flows and satisfy a larger fraction of the demand, compared to STDMA-LB. For the results shown in Fig. 4(a) and (b), β_{min} and β_{max} are set as 2^o and 10^o, respectively.

In the next set of simulations, performance of the three schedulers is evaluated in the "non-uniform" scenario as the number of nodes in the even sectors increases from 50 to 100. The total number of nodes in the network is 100 and the number of flows is 50. As the level of non-uniformity in the node density increases, STDMA-P and STDMA-E consistently outperform STDMA-LB as shown in Fig. 4(c) and (d). Due to its load-balancing beamwidth selection, STDMA-P consistently outperforms the other two methods. In addition, by performing stage extensions, STDMA-P results in a lower number of beam direction changes/reconfigurations, compared to STDMA-E.

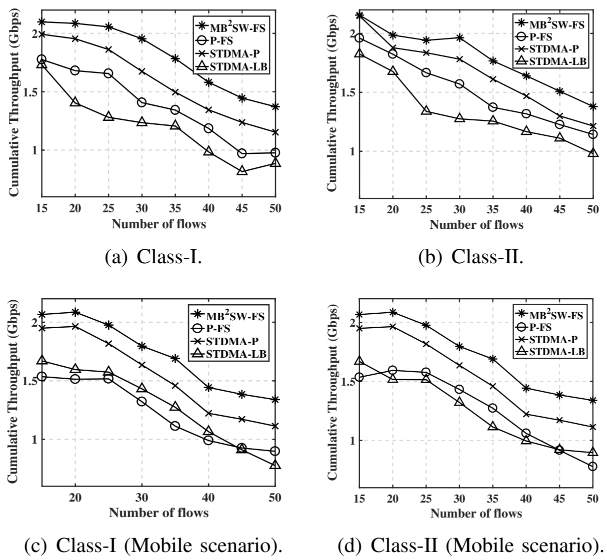
B. MB^{2}SW-FS vs Existing Schedulers

In this section, the performance of MB^{2}SW-FS is compared with the packet fair scheduler for local fairness (P-FS) [12], STDMA-LB, and STDMA-P in terms of flow throughput and fairness. The presented results are obtained in the "uniform" scenario with antenna beamwidth set to 20^o, and σ_1 set to 5. The scheduling process of P-FS is the same as that given in [12]. To transmit data to N_f, P-FS forms a beam b in direction N_f^d and the nodes in the sector of beam b are considered as the neighborhood of N_f. Each flow is assigned with a weight randomly selected from the group of considered weights: 0.33, 0.26, 0.24, and 0.20. Thus, the flows in the network can be categorized into four service classes (Class-I to IV). The fairness of each service class is computed using Jain's fairness index [33], while considering the cumulative bandwidth allocated to individual flows of that class. The presented results are averaged over 50 simulation runs.

Performance of MB^{2}SW-FS and STDMA-P is also evaluated in a mobile scenario where nodes move with walking speed in randomly selected directions for random durations and then remain static for randomly selected durations and repeat this cycle till the end of simulation. In addition links may get blocked for random durations.

Fig. 5(a) and (b) show the fairness of Class-I and II using all four methods. With the increase in the number of flows, the variance in the flow interference increases. Consequently, the variance in the received service of flows increases. Thus, the fairness indices of most of the service classes show a steeper trend as the number of demanding flows increases from 15 to 50. However, MB^{2}SW-FS consistently outperforms all other methods due to two reasons: (1) dynamic beam direction adjustment, and (2) fair allocation in the spatial reuse phase. STDMA-P schedules beams in such a way that the load is balanced among the scheduled beams but does not consider fairness among the flows. P-FS schedules beams greedily by considering nodes in the increasing order of their interference during its spatial reuse phase. Hence, the flows with a higher interference may receive a lower share of channel allocation compared to the flows with a lower interference. Thus the fairness indices with P-FS and STDMA-P are lower compared to that of MB^{2}SW-FS. STDMA-LB sometimes achieves better fairness indices than P-FS, but at the cost of reduced throughput. In the mobile scenario, by exploiting relay links, MB^{2}SW-FS and STDMA-P achieve better fairness than other two scheduler as shown in Fig. 5(c) and (d).

Fig. 6(a) and (b) show the cumulative throughput of Class-I and II with all four methods. P-FS and STDMA-LB do not implement dynamic beam configuration and beam direction adjustment. STDMA-P targets to minimize the beamforming overhead by executing the stage extension process which may not schedule all idle beams. On the other hand, MB^{2}SW-



(a) Class-I. (b) Class-II.

(c) Class-I (Mobile scenario). (d) Class-II (Mobile scenario).

Fig. 6: Number of flows vs cumulative throughput.

FS maximizes the number of busy beams by dynamically adjusting beam directions. In addition, it does not completely sacrifice throughput for fairness and implements spatial reuse beyond fairness constraints in a fair manner. Thus, it achieves a better throughput than STDMA-P, P-FS, and STDMA-LB. Similar performance is observed in the mobile scenario. However nodes may not find suitable relays each time their links get blocked, as a result, the throughput of a service class in the presence of blocked links is slightly lower than that in the absence of the blocked links as shown in Fig. 6(c) and (d).

As the number of flows increases from 15 to 50, the average number of beam direction reconfigurations (BDR) per frame with all four methods are: the BDR of STDMA-LB is 0; with STDMA-P, the BDR varies between 4 and 10 (if we consider stage extensions as beam direction reconfigurations); with MB^2SW-FS, the BDR is consistently maintained around 23; finally, with P-FS, the BDR is equal to the number of slots available in a frame, that is, 100. The results presented in this section show that MB^2SW-FS provides better throughput and fairness at the cost of a few extra beam direction reconfigurations. If we want to minimize the beamforming overhead while achieving a reasonably good throughput, then STDMA-P is the best choice.

X. Conclusion

This paper proposed two STDMA schedulers (STDMA-P and MB^2SW-FS) for WPANs/WLANs where the PNCs/APs are deployed with MBAAs. STDMA-P aims to balance the load among the beams and configures beams in such directions so that each beam covers the least possible number of nodes, and thereby increases the number of simultaneous non-overlapping beams supported by the available power. Also, when a beam becomes idle, STDMA-P tries to reconfigure that beam without disturbing the other non-idle beams, to reduce the beamforming overhead. On the other hand, MB^2SW-FS, by assigning service tags to various flows, allocates fair shares of bandwidth to flows with different weights. It improves network throughput and fairness by dynamically adjusting the directions of beams. Simulation results show that in non-uniform network scenarios, STDMA-P satisfies 24-41% higher demand than existing methods, and the improvement is greater for smaller beamwidths in the network. On the other hand, in highly loaded situations, MB^2SW-FS achieves 42% more throughput while providing 25% more fairness, compared to the existing methods.

Appendix

A. Proof of Lemma 1

If $\omega_f(v_2 - v_1) - L \leq 0$, then the lemma is true, since $W_f(t_1, t_2) \geq 0$. Hence, consider the case where $\omega_f(v_2 - v_1) - L > 0$. Since f is backlogged in the interval $[t_1, t_2]$ and $\omega_f(v_2 - v_1) - L > 0$, there exists one or more slot requests that receive service in the interval (v_1, v_2). Assume that, r_f^k is the first slot request to receive service in the open interval (v_1, v_2), and for r_f^k, it is true that $I(r_f^k) = F(r_f^{k-1})$. To prove this, we consider the possible cases for the start tag of r_f^k.
(1). According the service tag assignment, $I(r_f^k) \not< F(r_f^{k-1})$.
(2). Assume that $I(r_f^k) > F(r_f^{k-1})$ which implies r_f^k arrives at the PNC after r_f^{k-1} finishes service. If r_f^{k-1} finishes service at time $t \geq t_1$, then it is not possible for flow f to be in the backlogged state continually in the interval $[t_1, t_2]$, since $I(r_f^k) > F(r_f^{k-1})$. Hence, r_f^{k-1} cannot finish service at time $t \geq t_1$. If r_f^{k-1} finishes service at time $t < t_1$, then r_f^k arrives at time $A(r_f^k) \leq t_1$, since flow f is continually backlogged in the interval $[t_1, t_2]$. According to the tag assignment process, $I(r_f^k) = \max(V(A(r_f^k)), F(r_f^{k-1}))$. It is also true that $I(r_f^k) > v_1$, since r_f^k is served in the open interval (v_1, v_2). But, $V(A(r_f^k)) \leq v_1$ $(= V(t_1))$, since the virtual time of the network is a non-decreasing function of time. Hence, we conclude that $I(r_f^k) = F(r_f^{k-1})$.
From the arguments in (1) and (2), it is proved that $I(r_f^k) = F(r_f^{k-1})$. Since $I(r_f^k) > v_1$, $F(r_f^{k-1}) > v_1$. This implies $I(r_f^{k-1}) \leq v_1$. From the definition of tags, we get $F(r_f^{k-1}) \leq v_1 + \frac{L}{\omega_f}$ and $I(r_f^k) \leq v_1 + \frac{L}{\omega_f}$.

Assume that r_f^{k+m} is the last slot request of f to receive service in the interval $[t_1, t_2]$, which implies $F(r_f^{k+m}) \geq v_2$. We can also get $F(r_f^{k+m})$ as: $F(r_f^{k+m}) = I(r_f^k) + \sum_{i=0}^{m} \frac{L}{\omega_f}$, from the definition of fairness model. This implies $I(r_f^k) + \sum_{i=0}^{m} \frac{L}{\omega_f} \geq v_2$, and $\sum_{i=0}^{m} \frac{L}{\omega_f} \geq v_2 - I(r_f^k)$. Since $I(r_f^k) \leq v_1 + \frac{L}{\omega_f}$, we conclude that $\sum_{i=0}^{m} \frac{L}{\omega_f} \geq v_2 - v_1 - \frac{L}{\omega_f}$, and $\sum_{i=0}^{m} L \geq \omega_f(v_2 - v_1) - L$. Since, $\sum_{i=0}^{m} L$ is the service received by flow f in the interval $[t_1, t_2]$, $W_f(t_1, t_2)$, we get, $W_f(t_1, t_2) \geq \omega_f(v_2 - v_1) - L$. Hence, the lemma is proved.

B. Proof of Lemma 2

With MB^2SW-FS, the slot requests served in the interval (v_1, v_2) can have start tag at least v_1 and at most $(v_2 + 3\sigma)$. The slot requests of flow f that receive service in (v_1, v_2) and have start tag at least v_1 and at most $(v_2 + 3\sigma)$ and finish tag at most $(v_2 + 3\sigma)$ can be represented as: $\mu_1 = \{r_f^k | v_1 \leq I(r_f^k) \leq (v_2 + 3\sigma) \wedge F(r_f^k) \leq (v_2 + 3\sigma)\}$. Since one packet can be transmitted in each slot and from the tag assignment process,

$v_1 + \sum_{r_f^k \in \mu_1} \frac{L}{\omega_f} \leq (v_2 + 3\sigma)$. This implies, $\sum_{r_f^k \in \mu_1} \frac{L}{\omega_f} \leq (v_2 + 3\sigma) - v_1$, $\sum_{r_f^k \in \mu_1} L \leq \omega_f (v_2 + 3\sigma - v_1)$.

The slot requests of flow f that receive service in the interval (v_1, v_2) and have start tag at most $(v_2 + 3\sigma)$ and finish tag more than $(v_2 + 3\sigma)$ can be represented as: $\mu_2 = \{r_f^k | v_1 \leq I(r_f^k) \leq (v_2 + 3\sigma) \wedge F(r_f^k) > (v_2 + 3\sigma)\}$. Only one slot request belongs to μ_2. Hence, $\sum_{r_f^k \in \mu_2} L = L$. Since the slot requests served in μ_1 and μ_2 together represent the service received by f in $[t_1, t_2]$, $W_f(t_1, t_2)$, hence, the lemma follows.

C. Proof of Proposition 2

For the proof of Proposition 2, ω_f is interpreted as the rate assigned flow f.

Let $V(t_1) = v_1$. $\widetilde{W}(v_1, v_2)$ denotes the cumulative service received by all flows in the network in the interval $[v_1, v_2]$. From Lemma 2, $\widetilde{W}(v_1, v_2)$ can be obtained as: $\widetilde{W}(v_1, v_2) = \sum_{f \in G} \omega_f (v_2 + 3\sigma - v_1) + \sum_{f \in G} L$. In each slot, B beams are available for data reception/transmission. However, in the worst case, only one beam gets scheduled in each slot, in the fair allocation phase. Hence, $\sum_{f \in G} \omega_f = \frac{C}{B}$. This implies, $\widetilde{W}(v_1, v_2) = \frac{C}{B}(v_2 + 3\sigma - v_1) + \sum_{f \in G} L$. Now, define v_2 as: $v_2 = v_1 + t_2 - t_1 - 3\sigma - \frac{B \sum_{f \in G} L}{C}$. Substituting v_2 in $\widetilde{W}(v_1, v_2)$, we get $\widetilde{W}(v_1, v_2) = \frac{C}{B}(t_2 - t_1)$.

Let $t^{'}$ be such that $V(t^{'}) = v_2$. Assume that, function $T(W)$ denotes the time required to serve packets with the cumulative length W. Then, $t^{'} \leq t_1 + T(\widetilde{W}(v_1, v_2))$, and $t^{'} \leq t_1 + T(\frac{C}{B}(t_2 - t_1))$. However, assuming consistent transmission rate for all links in the interval (v_1, v_2), $T(W) \leq \frac{BW}{C}$. Hence, $t^{'} \leq t_1 + \frac{C}{B}(t_2 - t_1)\frac{B}{C} \leq t_2$. From Lemma 1, we know that $W_f(t_1, t^{'}) \geq \omega_f(v_2 - v_1) - L$. Since $t^{'} \leq t_2$, we get $W_f(t_1, t_2) \geq \omega_f(v_1 + t_2 - t_1 - 3\sigma - \frac{B \sum_{f \in G} L}{C} - v_1) - L \geq \omega_f(t_2 - t_1 - 3\sigma - \frac{B \sum_{f \in G} L}{C}) - L$. Since, ω_f is the rate assigned to flow f,

$$W_f(t_1, t_2) \geq \frac{C\omega_f}{B \sum_{g \in G} \omega_g}(t_2 - t_1) - \Delta,$$

$$\Delta = \frac{C\omega_f}{B \sum_{g \in G} \omega_g}(3\sigma) + \omega_f \frac{\sum_{f \in G} L}{\sum_{g \in G} \omega_g} + L.$$

Hence, the proposition follows.

References

[1] I. Jawhar and J. Wu, "Resource allocation in wireless networks using directional antennas", *Proc. PerCom*, pp. 318-327, 2006.

[2] L. Bao and J. J. Garcia-Luna-Aceves, "Transmission scheduling in ad hoc networks with directional antennas", *Proc. ACM MobiCom*, pp. 48-58, 2002.

[3] F. Shad, T. D. Todd, V. Kezys and J. Litva, "Indoor SDMA capacity using a smart antenna base station", *Proc. IEEE ICUPC*, vol. 2, pp. 868-872, 1997.

[4] J. Wang, Y. Fang and D. Wu, "Enhancing the performance of medium access control for WLANs with multi-beam access point", *IEEE Trans. Wireless Comm.*, vol. 6, no. 2, pp. 556-565, 2007.

[5] Z. Tang, X. Xing and F. Jiang, "Providing balanced and enhanced transmission for WLANs with multi-beam access point", *Proc. IEEE CNSR*, pp. 242-248, 2008.

[6] Z.-T. Chou, C.-Q. Huang, and J. M. Chang, "QoS provisioning for wireless LANs with multi-beam access point", *IEEE Transactions on Mobile Computing*, vol. 13, no. 9, pp. 2113-2127, 2014.

[7] A. Demers, S. Keshav and S. Shenker, "Analysis and simulation of a fair queueing algorithm", *Proc. ACM SIGCOMM'89*, pp. 1-12, 1989.

[8] Y. Niu, Y. Li, D. Jin, L. Su, and D. Wu, "Blockage robust and efficient scheduling for directional mmWave WPANs", *IEEE Trans. Veh. Technol.*, vol. 64, no. 2, pp. 728-742, 2014.

[9] P. Goyal, H. M. Vin, and H. Cheng, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks", *Technical Report TR-96-02, University of Texas at Austin*, 1996.

[10] H. Luo and S. Lu, "A topology-indepedent fair queueing model in ad hoc wireless networks", *Proc. IEEE ICNP*, pp. 325-335, 2000.

[11] H. Luo, J. Cheng, and S. Lu, "Self-coordinating localized fair queueing in wireless ad hoc networks", *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 86-98, 2004.

[12] H. Luo, S. Lu, and V. Bharghavan, "A new model for packet scheduling in multihop wireless networks", *Proc. ACM MOBICOM'00*, pp. 76-86, 2000.

[13] H. L. Chao and W. Liao, "Fair scheduling with QoS support in wireless ad hoc networks", *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 2119-2128, 2004.

[14] H. L. Chao and W. Liao, "Fair scheduling in mobile ad hoc networks with channel errors", *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1254-1263, 2005.

[15] Z. Yang, L. Cai, and W.-S. Lu, "Practical scheduling algorithms for concurrent transmissions in rate-adaptive wireless networks", *Proc. IEEE INFOCOM'10*, pp. 1-9, 2010.

[16] L. R. Lakshmi and B. Sikdar, "Fair scheduling of concurrent transmissions in directional antenna based WPANs/WLANs", *Proc. IEEE ICC'18*, pp. 1-6, 2018.

[17] L. R. Lakshmi and B. Sikdar, "Blockage aware fair scheduling with differentiated service support in mmWave WPANs/WLANs", *IEEE Transactions on Mobile Computing, Accepted*, 2019.

[18] I. K. Son, S. Mao, M. X. Gong, and Y. Li, "On frame-based scheduling for directional mmWave WPANs", *Proc. IEEE INFOCOM*, pp. 2149-2157, 2012.

[19] J. Qiao, L. X. Cai, X. S. Shen and J. W. Mark, "Enabling multi-hop concurrent transmissions in 60 GHz wireless personal area networks" *IEEE Trans. Wireless Comm.*, vol. 10, no. 11, pp. 3824-3833, 2011.

[20] L. R. Lakshmi and B. Sikdar, "STDMA scheduling for WLANs and WPANs with non-uniform traffic demand", *Proc. IEEE LANMAN'17*, pp. 1-6, 2017.

[21] P. Zhou, X. Fang, and Y. Long, "Throughput and robustness guaranteed beam tracking for mmWave wireless networks", *Proc. IEEE CIC-ICCC*, pp. 1-6, 2017.

[22] R. Ramanathan, "On the performance of ad hoc networks with beam forming antennas", *Proc. ACM MobiHoc*, pp. 95-105, 2001.

[23] J. Ning, T.-S. Kim, S. V. Krishnamurthy, and C. Cordeiro, "Directional neighbor discovery in 60 GHz indoor wireless networks", *Proc. ACM MSWiM*, pp.365-373, 2009.

[24] H. Deng and A. Sayeed, "Mm-wave MIMO channel modelling and user localization using sparse beamspace signatures", *Proc. IEEE SPAWC*, pp. 130-134, 2014.

[25] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with eyes closed: mm-wave beam steering without in-hand measurement", *Proc. IEEE INFOCOM*, pp. 2416-2424, 2015.

[26] H. D. Sherali and W. P. Adams, "A reformulation-linearization technique for solving discrete and continuous nonconvex problems", Boston, MA: Kluwer Academic, 1999.

[27] S. Singh, F. Ziliotto, U. Madhow, E. Belding, and M. Rodwell, "Blockage and directivity in 60 GHz wireless personal area networks: from cross-layer model to multihop MAC design", *IEEE J. Sel. Areas Commun.*, vol. 27, no. 8, pp. 1400-1413, 2009.

[28] IEEE 802.15 WPAN Millimeter Wave Alternative PHY Task Group 3c (TG3c). Available: http://www.ieee802.org/15/pub/TG3c.html.

[29] IEEE Draft Standard 802.11ad, "Wireless LAN MAC and PHY specifications - Amendment 4: enhancements for very high throughput in the 60GHz band", 2012.

[30] J. H. Kwon and E. J. Kim, "Asymmetric directional multicast for capillary Machine-to-Machine using mmWave communications", *Sensors*, vol. 16, no. 4, pp. 515, 2016.

[31] IEEE P802.15-05-0493-27-003c, "TG3c Selection Criteria", Jan., 2007.

[32] N. Moraitis and P. Constantinou, "Indoor channel measurements and characterization at 60 GHz for wireless local area network applications", *IEEE Trans. Antennas Propag.*, vol. 52, no. 12, pp. 3180-3189, 2004.

[33] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: an explanation", *ATM Forum Document Number: ATM Forum/990045*, Feb. 1999.