

Frame-Based Medium Access Control for 5G Wireless Networks

In Keun Son · Shiwen Mao · Yihan Li · Min Chen ·
Michelle X. Gong · Theodore (Ted) S. Rappaport

© Springer Science+Business Media New York 2015

Abstract Millimeter wave (mmWave) communications is one of the key technologies for future 5th generation (5G) wireless networks. In this paper, we investigate the problem of medium access control (MAC) in mmWave networks. We develop a frame-based scheduling directional MAC protocol, termed FDMAC, to achieve the goal of leveraging collision-free concurrent transmissions to fully exploit spatial reuse in mmWave networks. The high efficiency of FDMAC is achieved by amortizing the scheduling overhead over multiple concurrent, back-to-back transmissions in a row. The core of FDMAC is a graph coloring-based scheduling algorithm, termed greedy

coloring (GC) algorithm, that can compute near-optimal schedules with respect to the total transmission time with low complexity. FDMAC is analyzed and evaluated with simulations.

Keywords 5G wireless · 60GHz · Frame-based scheduling · Graph coloring · mmWave communications

1 Introduction

The next generation mobile communication networks, or the 5th generation (5G) networks, are to address the grand challenge of 1000 times wireless data increase by 2020 as compared to the 2010 level [2]. As indicated in the Qualcomm report, more spectrum and small cells are the key elements in the solution to meet this challenge. In this paper, we consider the problem of medium access control in millimeter wave (mmWave) small cells. With the massive bandwidth in the mmWave band, many bandwidth-demanding new applications can be easily supported in mmWave wireless networks [3, 30]. On the other hand, small cells are the enabler of efficient spatial reuse of the spectrum, which has achieved the largest increase in wireless network capacity in the past decades, comparing to other technical advances such as advanced modulation and coding techniques.

The 60 GHz band has a huge amount of unlicensed bandwidth (i.e., up to 7 GHz) that is available in most parts of the world. Several standards have been or are being defined to achieve multi-gigabit rates for 60 GHz networks, such as ECMA-387 [4], IEEE 802.15.3c [5], and IEEE 802.11ad [6]. Although high data rates up to multi-gigabps can be supported, mmWave communications in the 60 GHz band suffer severe attenuation. The propagation attenuation of 60 GHz signals in free space is 22 dB higher than that

I. K. Son
Defense Acquisition Program Administration (DAPA)
Republic of Korea, Seoul, South Korea
e-mail: soninkeun@gmail.com

S. Mao (✉) · Y. Li
Department of Electrical, Computer Engineering,
Auburn University, 200 Broun Hall, Auburn,
AL 36849-5201, USA
e-mail: smao@ieee.org

Y. Li
e-mail: yli@auburn.edu

M. Chen
School of Computer Science and Technology, Huazhong
University of Science and Technology, Wuhan, China
e-mail: minchen2012@hust.edu.cn

M. X. Gong
Google, Mountain View, CA, USA
e-mail: michellegong@google.com

T. S. Rappaport
NYU WIRELESS NYU-Poly Brooklyn Campus 2 MetroTech
Center, 9th Fl, Brooklyn, NY 11201, USA
e-mail: tsr@nyu.edu

of 5 GHz signals, and atmospheric absorption for 60 GHz signals ranges from 15 to 30 dB/km [7, 11]. The 60 GHz systems have been shown viable for short range communications [8] which require high data rate transmissions over short distances (e.g., High-definition television (HDTV) or mass storage synchronization), while some feasibility study for cellular networks has been made [7]. Due to the small wavelength, many small antennas (e.g., like the head of a pin) can be assembled and arrayed in a small platform. Beamforming has been used as an essential technique to overcome attenuation. Directional transmissions should be explicitly considered in the design of MAC protocols for mmWave networks [31].

As a result of directional listening and transmissions, the signal strength is usually very low at third party nodes, making it difficult to perform carrier sense. This is often referred to as the *deafness problem*. On the other hand, such reduced interference to neighboring links can be exploited for significantly enhanced spatial reuse. It is possible to schedule concurrent transmissions at multiple links without interfering each other, so as to improve network capacity. As shown in [9], the links in an mmWave network can be regarded as *pseudo-wired*, with negligible interference to each other. New scheduling algorithms and directional MAC protocols are needed to ensure communicating nodes meet each other for transmissions, and to exploit concurrent wired-like transmissions for maximizing spatial reuse.

Although mmWave prototype chipsets are emerging [10], their performance in a *network setting* and the design of efficient 60 GHz networking protocols remain open areas of future research [11]. Efficient MAC scheduling is indispensable to fully harvest the multi-gigabps physical layer (PHY) rates for high application layer throughput [11, 12]. In this paper, we investigate the problem of efficient scheduling in mmWave small cell networks. We consider an mmWave small cell consisting of a piconet coordinator (PNC) and multiple devices (DEVs).¹ Assuming some bootstrapping mechanism [13], the nodes are aware of each other's locations and always point their beams to the PNC when idle. The PNC collects traffic demands from the DEVs and computes schedules to enable concurrent directional transmissions. We consider *frame-based scheduling* in this paper and term the proposed protocol *frame-based directional MAC* (FDMAC). The objective is to leverage collision-free concurrent transmissions to fully exploit spatial reuse in mmWave small cells.

In FDMAC, network time is partitioned into a sequence of non-overlapping intervals, termed *frames*. Each frame

consists of two phases: (i) a *scheduling phase*, when the PNC collects traffic demands from DEVs and computes a transmission schedule, and (ii) a *transmission phase*, when DEVs start concurrent transmissions following the schedule. Packets arriving during the current frame will be stored at the PNC/DEVs and then scheduled to be transmitted in the next frame. In the scheduling phase, the PNC collects traffic demands from the DEVs, computes a schedule to accommodate the traffic demands, and then transmits the schedule to the DEVs. A schedule consists of (i) a sequence of topologies, each indicating how the DEVs are paired to form directional links, and (ii) a sequence of time intervals, each indicating how long each topology should sustain. During the transmission phase, the PNC/DEVs pair with each other and start transmitting packets for a number of time slots, as specified in the schedule.

The core of FDMAC is a scheduling algorithm that can compute a schedule for given traffic demands, such that the total transmission time is minimized. We formulate the frame-based scheduling problem as a mixed integer nonlinear programming (MINLP) problem, and develop a *greedy coloring (GC) algorithm* based on graph coloring to compute near-optimal solutions with low computational complexity. We also develop an enhancement to FDMAC that can transmit packets that arrive during the current frame whenever possible, for further improved network capacity.

We analyze the proposed FDMAC protocol and derive its performance lower and upper bounds. We also evaluate FDMAC with simulations. Our simulations validate the superior performance of FDMAC, which is shown to outperform two state-of-the-art mmWave MAC schemes with regard to delay and throughput, while achieving comparable fairness performance.

The remainder of this paper is organized as follows. We discuss related work in Section 2. The system model and problem statement are given in Section 3. We present the GC algorithm and its enhancement in Section 4. Simulation results are given in Section 5. Section 6 concludes this paper.

2 Related work

Various technologies have been exploited in the past to enhance the wireless network capacity [14–17]. There have been considerable work on directional MAC protocols in the literature (e.g., see [18]). Both standards for multi-gigabit 60 GHz WPANs, i.e., ECMA-387 [4] and IEEE 802.15.3c [5], adopt Time Division Multiple Access (TDMA) for data communications. Two existing MAC protocols recently proposed for 60 GHz networks [20, 21] are also based on TDMA. Due to the bursty nature of data traffic, the required medium time by a busy node is often highly unpredictable.

¹Without loss of generality, we use the wireless Personal Area Network (WPAN) terms for the nodes in the small cell [5]. The PNC can be a small cell base station, or it can be a regular node that is elected to be the coordinator to the nodes in a small area if the 5G network operates in the ad hoc mode.

A TDMA-based MAC protocol may cause either high control overhead for on-the-fly medium reservation, or under- or over-allocated medium time for individual users.

In centralized mmWave WPAN MAC schemes, all data transmissions are coordinated by the PNC to maintain synchronous pointing between two DEVs. Gong et al. propose a directional CSMA/CA protocol to exploit virtual carrier sensing and depend on the PNC to distribute network allocation vector (NAV) information [19]. This scheme extends the conventional CSMA/CA framework and mainly focuses on solving the deafness problem. It does not fully exploit spatial reuse for concurrent transmissions.

Another centralized approach, the multihop relay directional MAC (MRDMAC) protocol is based on the conventional AP-based single hop MAC architecture for keeping primary connectivity [22]. Most data transmissions are via the PNC with a sequential polling policy. When the direct link toward the PNC is blocked, the PNC selects an intermediate node to relay the traffic. MRDMAC solves the deafness problem by PNC's weighted round robin scheduling and overcomes blocked direct links by relaying. However, it has similar limitation on not utilizing concurrent transmissions, since most transmissions go through the PNC.

Distributed MAC protocols are motivated by the fact that mmWave, especially the 60 GHz band, can attain high medium utilization with highly directional point-to-point communications. The recently proposed memory-guided directional MAC (MDMAC) [23] and directional-to-directional MAC (DtDMAC) [24] are fully distributed schemes. Both directional MAC protocols can alleviate the deafness effect by incorporating a Markov state transition diagram. MDMAC assumes time division multiplexing (TDM) while DtDMAC uses an exponential backoff procedure for asynchronous operation.

3 System model and problem statement

3.1 System model

Consider an mmWave small cell consisting of a PNC and multiple DEVs. Time is divided into non-overlapping, equal-length intervals, i.e., *time slots*. The PNC synchronizes the DEV clocks and coordinates medium access in the small cell to accommodate traffic requests from the DEVs.

Let there be n DEVs in the small cell and one DEV, denoted as DEV1, is selected to serve as the PNC. We assume a bootstrapping scheme in place (e.g., see [13]) such that each DEV has update-to-date network topology/node location information. It can steer its directional beam toward either the PNC or another DEV for transmitting/receiving data. When idle, all the DEVs point their beams to the PNC. The PNC polls traffic requests from DEVs and pairs them

for directional transmissions. The PNC can also serve as a transmitter or receiver during the transmission phase. We use the term *node* to refer to a PNC or a DEV if there is no need to distinguish them.

The directional connectivity formed in each time slot can be modeled as a directed graph (or, *digraph*). Let $G^t(V, E^t)$ be such a connectivity graph in time slot t , where V is the set of nodes and E^t is the set of directional links. If node i is scheduled to transmit to node j in time slot t , then we have edge $e_{ij}^t \in E^t$. Note that e_{ij}^t and e_{ji}^t are different edges due to different transmitter/receiver roles for the two endpoints.

3.2 Frame-based scheduling and matching

We focus on *frame-based scheduling* and term the proposed scheme *frame-based directional MAC* (FDMAC). In the following, we first define a few terms and then describe the FDMAC protocol.

Definition 1 (*Frame*) With FDMAC, time is divided into a sequence of non-overlapping frames, each consisting of a *scheduling phase*, where a transmission schedule is computed by the PNC, and a *transmission phase*, where DEVs start concurrent transmissions following the schedule.

In the n -node small cell, every node maintains $n - 1$ *virtual queues*, each storing the packets destined to a distinct neighbor. The backlogs of the virtual queues are transmitted to the PNC at the beginning of a frame, i.e., during the scheduling phase, and are used as input for computing the transmission schedule.

Definition 2 (*Traffic demand vector d_i*) For node i , the traffic demand vector d_i is an n -element vector, where each element d_{ij} represents the number of time slots required for node i to transmit all its packets for node j , i.e., to clear the backlog of virtual queue j at node i . Note that $d_{ii} = 0$.

Definition 3 (*Traffic demand matrix D*) The traffic demand matrix D is an $n \times n$ matrix, where the i th row is the traffic demand vector d_i , for all i .

The operation of FDMAC is illustrated in Fig. 1. In FDMAC, traffic received in a frame is first cumulated in the virtual queues, and then gets transmitted in the next frame. A frame consists of a scheduling phase with duration t_{ph}^{sch} followed by a transmission phase with duration t_{ph}^{tr} . In the beginning of the scheduling phase, all the DEVs point to the PNC, and the PNC polls the DEVs for their traffic demand vectors d_i . In the example shown in Fig. 1a, we have $d_2 = [3 \ 0 \ 3 \ 0]$, meaning DEV2 requests three time slots for transmitting to DEV1 and three time slots for DEV3. It takes

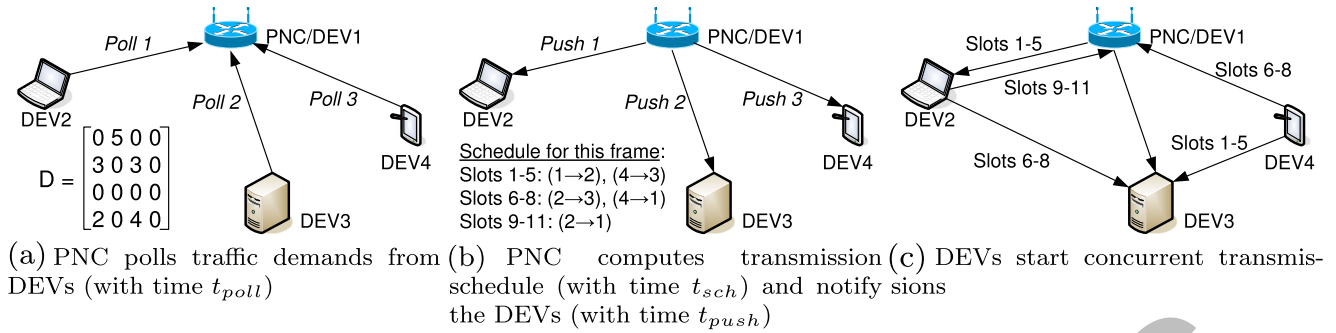


Fig. 1 Operation of the frame-based scheduling directional MAC protocol (FDMAC)

time t_{poll} for the PNC to collect all the d_i 's and assemble the traffic demand matrix D , as shown in Fig. 1a.

Then, the PNC will compute a transmission schedule to serve the traffic demand D (i.e., to clear the backlog received in the previous frame at all the nodes), which takes time t_{sch} . The schedule consists of multiple elements, each pairing the nodes for a certain number of time slots. As shown in Fig. 1b, the schedule is (i) DEV1 transmits to DEV2, and DEV4 to DEV3 for five time slots, (ii) DEV2 transmits to DEV3, and DEV4 to DEV1 for the next three time slots, and (iii) DEV2 transmits to DEV1 for the last three time slots. Note that this schedule is sufficient for clearing the backlogs at all the nodes, i.e., $D \prec S$. Written in the matrix form, the transmission schedule S is

$$D \prec S = 5 \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} + 3 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} + 3 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \quad (1)$$

The transmission schedules are then transmitted to all the nodes, which takes time t_{push} . In the following transmission phase, the nodes pair with each other following the transmission schedule, and start directional, concurrent transmissions, as shown in Fig. 1c. Overall, it takes 11 time slots to serve the traffic demand D , a considerable reduction comparing to the conventional sequential approach that transmits one packet at a time via the PNC (which takes 24 time slots).²

During the current frame, new packet arrivals to the nodes are backlogged in their virtual queues, and will be scheduled for transmission in the next frame.

²With the conventional approach, at most one transmission is made in each time slot. When the PNC is the target receiver or source, each packet will be transmitted once using one time slot. When the PNC is neither the source nor the receiver, each packet will be relayed by the PNC and take two time slots. The amount of time slots for clearing out the traffic matrix D is: $5 + 3 + 2 + 2 \times 3 + 2 \times 4 = 24$ time slots.

3.3 Properties

Definition 4 (Schedule graph G^t) In each time slot t , the connectivity of the small cell, as indicated in the transmission schedule, can be represented by a schedule graph $G^t(V, E^t)$, a digraph where V is the set of nodes and E^t is the set of directional links as given by the schedule in time slot t .

Definition 5 (Schedule adjacency matrix A^t) The schedule adjacent matrix A^t is the $n \times n$ adjacency matrix of digraph G^t [25].

In each time slot t , the topology, as formed by the corresponding transmission schedule, can be represented by the schedule graph G^t or its adjacency matrix A^t . Within a frame, the topology evolves from time slot to time slot as different pairings of the nodes are formed.

Proposition 1 The digraph of the mmWave small cell in time slot t , $G^t(V, E^t)$, is a matching.

Proof A matched graph has a set of edges, which are not adjacent to any other edges [25]. If vertex v is connected to vertex u , vertices u and v should not have any other edges except for edge (u, v) . Note that the links are point-to-point connections and each node can have at most one connection with one neighbor. Thus, any edge element in E^t cannot have adjacent edges. Therefore G^t is always a matching. \square

Corollary 01 The maximum number of concurrent transmissions in the schedule graph $G^t(V, E^t)$ is $\lfloor \frac{n}{2} \rfloor$, where $n = |V|$ is the number of nodes.

Therefore, the capacity of the small cell is upper bounded. The small cell capacity is upper bounded by $\lfloor \frac{n}{2} \rfloor \geq 100\%$, for $n \geq 4$.

3.4 Problem statement and reformulation

To avoid frequent beamforming/steering, we assume that each link can be activated at most once in a schedule. When a directional link (i, j) is formed, it should be sustained for at least d_{ij} time slots in order to serve all the backlogged packets in the j th virtual queue at node i . Then we have the following proposition.

Proposition 2 *The minimum number of required time slots for $G^t(V, E^t)$ is*

$$\delta^t = \max \left\{ d_{ij} | a_{ij}^t = 1, i, j \in V \right\}, \quad (2)$$

where a_{ij}^t is the (i, j) th element of adjacent matrix A^t .

The scheduling algorithm decomposes the traffic demand matrix D into K matrices, each being an adjacency matrix A^k describing a topology, i.e., a pairing of the nodes with directional links, and a duration δ^k describing how long the topology will last. A feasible schedule should satisfy

$$D \prec S = \delta^1 A^1 + \delta^2 A^2 + \delta^3 A^3 + \dots + \delta^K A^K. \quad (3)$$

The number of time slots for the schedule is $\sum_{k=1}^K \delta^k$.

For a given demand matrix D , there are many feasible schedules. An optimal schedule should clear the backlog D with the minimum number of time slots, which implies maximum parallelism of transmissions. We formulate the frame-based scheduling problem as

$$\text{minimize: } \sum_{k=1}^K \delta^k \quad (4)$$

subject to:

$$\sum_{k=1}^K a_{ij}^k = \begin{cases} 1, & \text{if } d_{ij} > 0 \\ 0, & \text{otherwise} \end{cases} \text{ for all } i, j \quad (5)$$

$$a_{ij}^k \in \begin{cases} \{0, 1\}, & \text{if } d_{ij} > 0 \\ \{0\}, & \text{otherwise} \end{cases} \text{ for all } i, j, k \quad (6)$$

$$\sum_{k=1}^K (\delta^k \cdot a_{ij}^k) \begin{cases} \geq d_{ij}, & \text{if } d_{ij} > 0 \\ = 0, & \text{otherwise} \end{cases} \text{ for all } i, j \quad (7)$$

$$\sum_{j=1}^n (a_{ij}^k + a_{ji}^k) \leq 1 \text{ for all } i, k. \quad (8)$$

This is an MINLP problem, which is generally NP-hard. Constraint in Eqs. 5 and 6 indicate that each directional link should be scheduled once. Constraint in Eq. 7 specifies that the number of time slots for links should be equal to or larger than the backlog. Constraint in Eq. 8 indicates that each schedule graph should be a matching, as given in Proposition 1.³

³We focus on the scheduling problem in this paper, assuming ‘pseudo-wired’ mmWave links [9]. If in some cases such pseudo-wired assumption is not exactly true, the concept of *exclusive region* [26] can be incorporated in the problem formulation, as additional linear

The formulated problem has a nonlinear constraint in Eq. 7. We apply a relaxation technique, the *Reformulation-Linearization Technique* (RLT), to obtain a linear relaxation [27]. Specifically, we define a substitution variable $\mu_{ij}^k = \delta^k \cdot a_{ij}^k$. According to Proposition 2, δ^k is bounded as $0 \leq \delta^k \leq \bar{d}$, where $\bar{d} = \max \{d_{ij} | i, j \in V\}$. Also a_{ij}^k is bounded as $0 \leq a_{ij}^k \leq 1$ due to constraint Eq. 6. We can obtain the following *RLT bound-factor product constraints* for μ_{ij}^k

$$\begin{cases} \mu_{ij}^k \geq 0 \\ \delta^k - \mu_{ij}^k \geq 0 \\ \bar{d} \cdot a_{ij}^k - \mu_{ij}^k \geq 0 \\ -\delta^k - \bar{d} \cdot a_{ij}^k + \mu_{ij}^k \geq -\bar{d} \end{cases} \text{ for all } i, j, k. \quad (9)$$

Substituting μ_{ij}^k into Eq. 7, we obtain a mixed integer linear programming (MILP) relaxation as

$$\text{minimize: } \sum_{k=1}^K \delta^k \quad (10)$$

subject to:

$$\sum_{k=1}^K \mu_{ij}^k \begin{cases} \geq d_{ij}, & \text{if } d_{ij} > 0 \\ = 0, & \text{otherwise} \end{cases} \text{ for all } i, j \quad (11)$$

Constraints Eqs. 5, 6, 8, and 9.

4 Greedy coloring algorithm

4.1 Greedy coloring algorithm

The core of FDMAC is a scheduling algorithm that can solve Problem Eq. 4 in real-time (i.e., with small t_{sch}). The GC algorithm computes a near-optimal solution for a given demand D . We show that the scheduling problem can be modeled as an edge coloring problem. The proof is omitted due to lack of space.

Proposition 3 *For a given traffic demand matrix D , let $S = \delta^1 A^1 + \delta^2 A^2 + \dots + \delta^K A^K$ be a feasible schedule. Then S can be modeled as a K -edge-colorable graph.*

It follows Proposition 3 that solving the optimal scheduling problem is equivalent to obtaining the K -edge coloring of a directed and weighted multigraph $G(V, E)$ in order to minimize $\sum_{k=1}^K \delta^k$, where δ^k is the highest weight in the k th edge coloring. Edge coloring is a well studied problem, while the conventional edge coloring problem is to find the *edge chromatic number* ($\chi'(G)$) of a given graph G , which

constraints to exclude such interfering links being scheduled simultaneously. Alternatively, a more sophisticated link model as in [29] can be used.

is the minimum number of colors needed to color all edges in G . Our objective is to minimize the total transmission time $\sum_{k=1}^K \delta^k$, while the number of colors is not a concern. It is acceptable to use additional colors over the edge chromatic number if the total number of time slots can be further reduced.

Algorithm 1: Greedy Coloring Algorithm

Input: traffic demand D , set of n distinct vertices V , set of edges ordered by non-increasing weights E

Output: $G^1(V, E^1), \delta^1, G^2(V, E^2), \delta^2, \dots$

```

1 Obtain a directed and weighted multigraph  $G(V, E)$ ;
2  $t = 0$ ;
3 while  $|E| > 0$  do
4    $t = t + 1$ ;
5   Set  $G^t(V^t, E^t)$  with  $V^t = \emptyset$  and  $E^t = \emptyset$ ;
6   while there is unvisited edge in  $E$  and
      $|E^t| \leq \lfloor \frac{n}{2} \rfloor$  do
7     Get the unvisited edge with the max. weight
        $e_{ij} \in E$ ;
8     if  $i \notin V^t$  and  $j \notin V^t$  then
9        $E^t = E^t \cup \{e_{ij}\}$ ;
10       $V^t = V^t \cup \{i, j\}$ ;
11       $E = E - \{e_{ij}\}$ ;
12      if  $|E^t| == 1$  then
13         $\delta^t = \omega(e_{ij})$ ;
14      end
15    end
16  end
17  Output  $G^t(V, E^t)$  and  $\delta^t$ ;
18 end

```

The pseudo-code of the GC algorithm is given in Algorithm 1. It has the worst case complexity of $O(|E|^2)$. The GC algorithm first obtains a directed and weighted multigraph $G(V, E)$, where V is the set of n vertices representing the nodes, and E is a set of directed and weighted edges sorted in non-increasing order according to edge weights. GC next iteratively makes a matching until all elements are properly colored, where each graph with the same color is a matching. Line 6 in Algorithm 1 examines the termination conditions for each coloring; the algorithm stops when all possible edges are examined or when the number of colored edges reaches the maximum number as in Corollary 01. Line 8 is a matching condition to guarantee a set of pairwise nonadjacent edges.

4.2 Performance bound and enhancement

The GC algorithm selects a set of same-color edges in each iteration (i.e., Lines 4–17). Since the coloring procedure in each iteration stops when there is no more additional edge to be colored, each colored matching is *maximal*, meaning that no more edge can be incorporated. Let Λ^G be the *connectivity matrix* of multigraph $G(V, E)$ corresponding to traffic demand matrix D . Each element in Λ^G has a binary value:

$\Lambda_{ij}^G = 1$ if there is a directional connection from vertex i to vertex j ; otherwise, $\Lambda_{ij}^G = 0$.

We have the following performance bounds for the GC algorithm. The proof is omitted due to lack of space.

Proposition 4 *The number of colors K achieved by the GC algorithm is bounded as*

$$\Delta_v \leq K \leq \Delta_p \leq 4n - 6, \quad (12)$$

where Δ_v is the maximum node degree determined by $\max\{\Delta_i | i \in V\}$ and $\Delta_i = \sum_{j=1}^n (\Lambda_{ij}^G + \Lambda_{ji}^G)$, Δ_p is the maximum number of incident edges on a pair defined as $\max\{\Delta_{ij} | i, j \in V\}$ with $\Delta_{ij} = \Lambda_{ij}^G \cdot (\Delta_i + \Delta_j - \mu_{ij})$, and μ_{ij} is the multiplicity between vertices i and j .

The GC algorithm can be enhanced by allowing transmitting packets that arrive during the current frame whenever possible. Such an enhancement can further improve both the delay and throughput performance of FDMAC, especially in the congestion region. In our simulations, we find the enhanced FDMAC achieves an 8.6 % gain over FDMAC under incipient congestion when the offered load is 445 %.

5 Simulation studies

5.1 Simulation setup

We evaluate FDMAC and compare it with two existing schemes, i.e., MRDMAC [22] and MDMAC [23]. All the protocols are implemented in MATLAB.⁴ MRDMAC enables relaying traffic for two nodes with no direct connection, while most of the transmissions will be done in a single hop. We tune MRDMAC for TDM systems to represent a centralized PNC-based single hop MAC scheme. For MDMAC, we set P_{TI} , the probability of changing from the transmit mode to idle mode, to 0.01 and T_{ESR} , the threshold of explicit state reset, to 90 % as given in [23].

We adopt the same simulation parameters given in Table II of [22], with a 2 Gbps data rate. We assume fixed data packet size of 1,000 bytes. The duration of a time slot is

$$t_{slot} = t_{pkt} + t_{SIFS} + t_{ACK} \approx 5 \mu s, \quad (13)$$

where t_{pkt} is the packet transmission time, t_{SIFS} is the SIFS interval, and t_{ACK} is the transmission time of an ACK packet. Both t_{pkt} and t_{ACK} consider packet header, payload size, and propagation delay due to the nanosecond scale transmissions. The maximum duration of a single TXOP (i.e., transmission opportunity) is 100 μs , and thus the maximum number of time slots for a pairing is bounded by 20.

⁴The FDMAC execution times presented in Section 5.2.2 are measured with a C implementation.

In FDMAC, we need to calculate t_{ph}^{sch} , consisting of t_{poll} , t_{sch} , and t_{push} . Both t_{poll} and t_{push} are the required time for the PNC to sequentially exchange small amount of control data with the DEVs. The number of DEVs the PNC can access in a time slot can be calculated as $\left\lfloor \frac{t_{slot}}{t_{ShFr} + 2 \cdot t_{SIFS} + t_{ACK}} \right\rfloor$, where t_{ShFr} is the time to transmit a small control message. The total number of required time slots for t_{poll} and t_{push} can be estimated for a given network size n , which determines how many control messages should be transmitted during the scheduling phase. The transmission time t_{ph}^{tr} depends on the efficiency of the scheduling algorithm.

We compare the three mmWave MAC schemes with respect to average delay, network throughput, and fairness performance under *on-off bursty traffic*, where packets are generated following an on-off Markov chain model with geometrically distributed on and off periods. We have similar results for self-similar traffic models, which are omitted due to lack of space. When a packet is generated at a source node, its destination is randomly chosen from other nodes in the small cell. We consider both the *uniform* and *non-uniform* traffic patterns. In the uniform case, the destination of a packet is uniformly distributed among the other nodes. For the non-uniform traffic pattern, node i 's neighbors are divided into a heavily loaded subset \mathcal{N}_H^i and a lightly loaded subset \mathcal{N}_L^i . When a new packet arrives at node i , it is forwarded to a node in \mathcal{N}_H^i with probability $\alpha/|\mathcal{N}_H^i|$, or a node in \mathcal{N}_L^i with probability $(1 - \alpha)/|\mathcal{N}_L^i|$, where $\alpha > 0.5$.

5.2 Performance of FDMAC

5.2.1 Bounds on the number of colors

In this simulation, we assume that the traffic demand in each element of D has a random value upper bounded by 20. GC

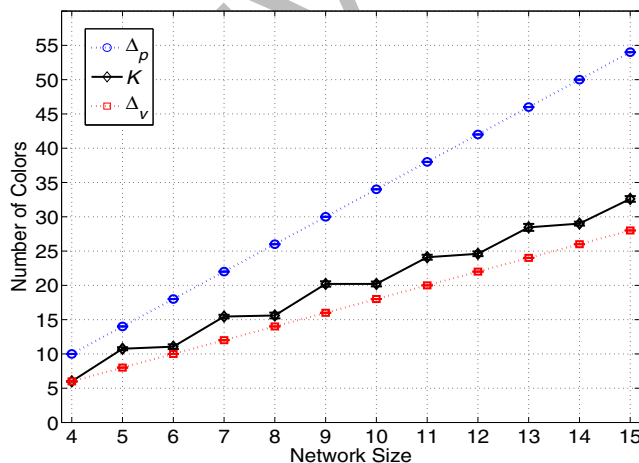


Fig. 2 Number of colors versus network size

Table 1 Execution Time of The GC Algorithm (μs)

Ntw. Size	Exe. Time	Ntw. Size	Exe. Time
4	2.862	10	17.343
5	4.252	11	22.013
6	5.237	12	25.213
7	8.605	13	35.488
8	10.673	14	43.426
9	14.809	15	52.765

is then applied to such randomly generated D matrices to compute feasible schedules.

In Fig. 2, we plot the upper and lower bounds on the number of colors given in Proposition 4, as well as that achieved by GC, for increased network sizes. Each point on the curves is the average of 20 randomly generated traffic demand matrices, with 95 % confidence intervals plotted as error bars, which are negligible. We find the GC results very close to the lower bound, i.e., the maximum node degree Δ_v , and the gap between the two is not susceptible to network size.

5.2.2 Computational complexity

As discussed, the GC algorithm can achieve near-optimal schedules with very low computational complexity. To evaluate execution time, we implement GC in C and test it with various network sizes. To accurately measure the execution time in microseconds, we used the *timespec* structure and the POSIX monotonic clock. The GC code is executed in a desktop computer with an Intel Core Duo™ CPU 2.2 GHz and 2GB RAM.

The execution times for different network sizes are presented in Table 1. To consider the heaviest traffic request, each traffic demand matrix D has the maximum number (i.e., $n \cdot (n - 1)$) of non-zero elements. The computation time is reduced from hundreds of seconds (with *lp_solve*) to tens of μs . Obviously a larger network takes more time. Let $t(n)$ be the GC execution time for a small cell of size n . The number of required time slots for the GC algorithm is $\left\lceil \frac{t(n)}{t_{slot}} \right\rceil$. Generally it takes only a few time slots for computing a near-optimal schedule with our C implementation, which can be effectively mitigated with the proposed algorithm.

5.3 Comparison with existing schemes

We next compare FDMAC with two state-of-the-art directional mmWave MAC protocols with a 10-node mmWave small cell. Initially, each DEV has a randomly generated

small amount of packets in the virtual queues. We consider on-off bursty traffic models, as well as uniform and non-uniform traffic patterns.

In the figures, the x -axis is the *Normalized Offered Load*. For example, an offered load 2 means the overall traffic load to the system, i.e., the sum of traffic rates at all the nodes, is twice of the capacity of a directional link. When the traffic pattern is non-uniform, we select three neighbor nodes to receive 40 % of the traffic load. We set the delay threshold to 10^4 , and discard packets with a larger delay than the threshold. Each protocol is run for 10^6 time slots. Each point in the figures is the average of 10 simulation runs, with 95 % confidence intervals plotted as error bars (most of which are negligible in the figures).

5.3.1 Delay performance

We first evaluate the delay performance of the three MAC protocols. In Figs. 3 and 4, we plot the average delay of received packets for different traffic loads. MRDMAC does not consider concurrent transmissions. As a result, its maximum throughput is under 100 %. In both plots, the average delay of MRDMAC diverges when the offered load is between 0.55 and 0.6.

MDMAC demonstrates similar results regardless of traffic patterns. The MDMAC delay diverges when the offered load is 3.05 for uniform traffic and 3.0 for non-uniform traffic. In both plots, FDMAC can significantly reduce the average delay and support a heavier traffic load. For example, the FDMAC delay diverges when the offered load exceeds 4.5 for uniform bursty traffic, which is very close to the upper bound as given in Corollary 01, and 4.0 for non-uniform bursty traffic, which is still 33.3 % higher than that of MDMAC.

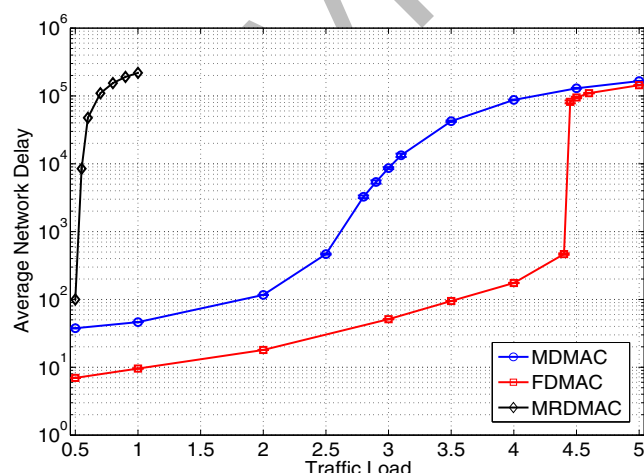


Fig. 3 Average delays of the three MAC protocols under uniform on-off bursty traffic

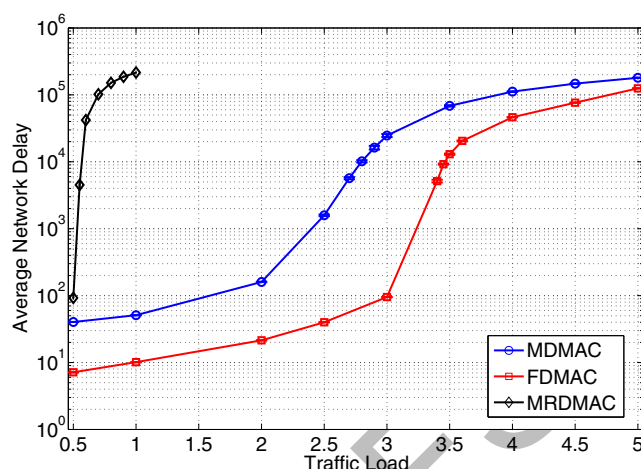


Fig. 4 Average delays of the three MAC protocols under non-uniform on-off bursty traffic

5.3.2 Throughput performance

The total number of successful transmissions achieved by the three MAC protocols are plotted in Figs. 5 and 6. The throughput performance is consistent with the delay performance observed earlier. MRDMAC has the lowest throughput since it does not support concurrent transmissions. FDMAC achieves a throughput close to 450 % under the uniform traffic pattern, while the MDMAC throughput is close to 340 %. FDMAC achieves a throughput close to 380 % under the non-uniform traffic pattern, while the MDMAC throughput is close to 330 % in this case. The normalized improvement ratios are 32.4 % and 15.2 %, respectively.

In Fig. 5, the FDMAC curves drop abruptly when the offered load exceeds 4.5. This is due to the fact that we equally increase each traffic demand element in D to

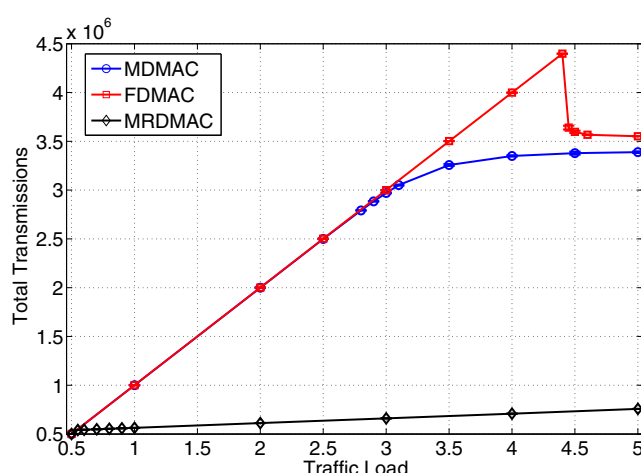


Fig. 5 Total number of successful packet transmissions of the three MAC protocols under uniform on-off bursty traffic

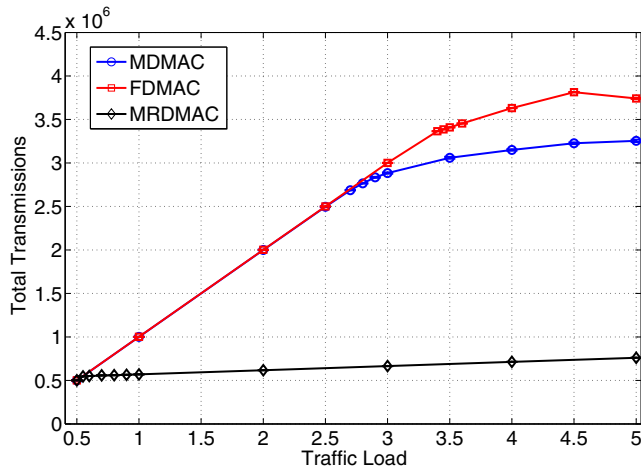


Fig. 6 Total number of successful packet transmissions of the three MAC protocols under non-uniform on-off bursty traffic

achieve heavier traffic loads. Thus, the backlogs at each DEV will be increased simultaneously. When the offer load is beyond 4.5, all the nodes start to suffer from congestion simultaneously, leading to an abrupt drop in the network throughput curve. In the non-uniform traffic pattern case, some heavily loaded nodes become congested earlier than other lightly loaded nodes. Therefore we do not see the abrupt drop in the network throughput curves.

5.3.3 Fairness performance

Finally, we examine the fairness performance of the three MAC protocols. We adopt Jain's fairness index as $f(w_1, w_2, \dots, w_n) = \frac{(\sum_{i=1}^n w_i)^2}{n \sum_{i=1}^n w_i^2}$, where w_i is the average delay at node i . The fairness index varies from 0 to 1. When all the nodes have the same average delay, the fairness index achieves its maximum value 1; when one node has a large

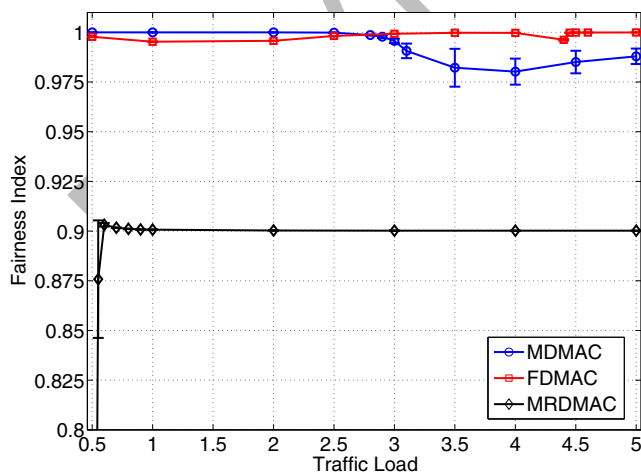


Fig. 7 Fairness indices of the three MAC protocols under uniform on-off bursty traffic

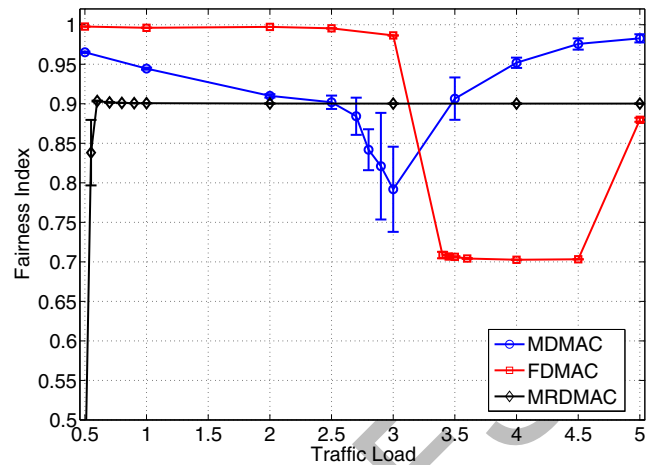


Fig. 8 Fairness indices of the three MAC protocols under uniform on-off bursty traffic

delay that dominates all other nodes, the fairness index is close to $1/n$.

The simulation results are presented in Figs. 7 and 8. We find both FDMAC and MDMAC achieve higher fairness indices than MRDMAC under the uniform traffic pattern, since MRDMAC is already in the heavily congested region for the range of loads shown in the plots. The fairness curves of FDMAC falls slightly when load exceeds 3.0 due to congestion. In the heavy-load region, the concurrent transmissions of MDMAC will be terminated not due to empty buffers, but by a state transition (as given by the state transition probability P_{TI}) or by an explicit state reset (as given by T_{EST}). For the non-uniform traffic cases, the MDMAC performance is affected by the biased traffic flows.

The FDMAC protocol provides comparable fairness performance as MDMAC under uniform traffic pattern, even when the offered load is increased beyond the congestion point. However, the fairness indices will drop to about 0.7 under non-uniform traffic pattern, when the offered load is increased beyond the congestion point. This is because those heavily loaded nodes keep on requesting the maximum link capacity and transmit for long period of time, while the other lightly loaded nodes request fewer time slots.

6 Conclusion

In this paper, we presented FDMAC for mmWave small cells. The main idea was to exploit concurrent transmissions to greatly improve network capacity, as enabled by the highly directional communications. Furthermore, the proposed FDMAC incorporated a frame-based scheduling algorithm to amortize the control overhead over a long sequence of concurrent packet transmissions. The proposed FDMAC was analyzed with respect to performance bounds and evaluated with simulations.

Acknowledgments This work is supported in part by the US National Science Foundation (NSF) through grants CNS-1320664 and CNS-1320472. This work was presented in part at IEEE INFOCOM 2012, Orlando, FL, Mar. 2012 [1].

References

1. Son IK, Mao S, Gong MX, Li Y (2012) On frame-based scheduling for directional mmWave WPANs In: Proceedings of the IEEE INFOCOM 2012, pp 2149–2157, Orlando
2. Qualcomm (2014) The 1000x mobile data challenge: More small cells, more spectrum, higher efficiency Technical report. Qualcomm, San Diego
3. He Z, Mao S (2014) Adaptive multiple description coding and transmission of uncompressed video over 60ghz networks. *ACM Mobile Computing and Communications Review (MC2R)* 18(1):14–24
4. (2008) ECMA TC48, ECMA standard 387, High rate 60 GHz PHY, MAC and HDMI PAL
5. (2009) IEEE 802.15.3 Working Group, Part 15.3: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs), IEEE Unapproved Draft Std P802.15.3c/D10
6. (2010) Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications C Amendment 5: Enhancements for Very High Throughput in the 60 GHz Band. IEEE P802.11ad/D1.0
7. Rappaport T, Gutierrez F, Ben-Dor E, Murdock JN, Qiao Y, Tamir JI (2013) Broadband millimeterwave propagation measurements and models using adaptive beam antennas for outdoor urban cellular communications. *IEEE Trans Ant Prop* 61(4):1850–1859
8. Geng S, Kivinen J, Zhao X, Vainikainen P (2009) Millimeter-wave propagation channel characterization for short-range wireless communications. *IEEE Trans Veh Tech* 58(1):3–13
9. Mudumbai R, Singh S, Madhow U (2009) Medium access control for 60 GHz outdoor mesh networks with highly directional links. In: Proceedings of the IEEE INFOCOM 2009 (Mini Conf.), pp 2871–2875, Rio de Janeiro, Brazil
10. Rappaport TS, Murdock JN, Gutierrez F (2011) State of the art in 60-GHz integrated circuits and systems for wireless communications. *Proc IEEE* 99(8):1390–1436
11. Daniels RC, Murdock JN, Rappaport TS, Heath RW (2010) 60-GHz wireless: Up close and personal. *IEEE Microw Mag* 11(7):44–50
12. An X (2010) Medium Access Control and Network Layer Design for 60 GHz Wireless Personal Area Networks. PhD thesis, Technische Universiteit Delft
13. Ning J, Kim TS, Krishnamurthy SV, Cordeiro C (2011) Directional neighbor discovery in 60 GHz indoor wireless networks. *Elsevier Performance Evaluation* 68(9):897–915
14. Ge X, Huang K, Wang C-X, Hong X, Yang X (2011) Capacity analysis of a multi-cell multi-antenna cooperative cellular network with co-channel interference. *IEEE Trans Wireless Commun* 10(10):3298–3309
15. Xiang L, Ge X, Wang C-X, Li FY, Reichert F (2013) Energy efficiency evaluation of cellular networks based on spatial distributions of traffic load and power consumption. *IEEE Trans Wireless Commun* 12(3):961–973
16. Ge X, Han T, Zhang Y, Mao G, Wang C-X, Zhang J, Yang B, Pan S (2014) Spectrum and energy efficiency evaluation of two-tier femtocell networks with partially open channels. *IEEE Trans Vehi Technol* 63(3)
17. Ge X, Hu J, Wang C-X, Youn C-H, Zhang J, Yang X (2014) Energy efficiency analysis of MISO-OFDM communication systems considering power and capacity constraints. *ACM/Springer Mobile Netw Appl* 17(1):29–35
18. Korakis T, Jakllari G, Tassioulas L (2008) CDR-MAC: A protocol for full exploitation of directional antennas in ad hoc wireless networks. *IEEE Trans Mobile Comp* 7(2):145–155
19. Gong MX, Stacey RJ, Akhmetov D, Mao S (2010) Performance analysis of a directional CSMA/CA protocol for mmWave wireless PANs. In: Proceedings of the IEEE WCNC'10, Sydney, Australia
20. An X, Hekmat R (2008) Directional MAC protocol for millimeter wave based wireless personal area networks. In Proceedings of the IEEE VTC-Spring'08, pp 1636–1640, Singapore
21. Pyo C-W, Kojima F, Wang J, Harada H, Kato S (2009) MAC enhancement for high speed communications in the 802.15.3c mmWave WPAN. *Springer Wireless Pers Commun* 51(4):825–841
22. Singh S, Ziliotto F, Madhow U, Belding EM, Rodwell M (2009) Blockage and directivity in 60 GHz wireless personal area networks: From cross-layer model to multihop MAC design, vol 27
23. Singh S, Mudumbai R, Madhow U (2010) Distributed coordination with deaf neighbors: Efficient medium access for 60 GHz mesh networks. In: Proceedings of the IEEE INFOCOM, pp 1–9, San Diego
24. Shihab E, Cai L, Pan J (2009) A distributed asynchronous directional-to-directional MAC protocol for wireless ad hoc networks. *IEEE Trans Veh Tech* 58(9):5124–5134
25. Gross JL, Yellen J (1999) Graph Theory and Its Applications. CRC Press
26. Cai LX, Cai L, Shen X, Mark JW (2010) REX: a Randomized EXclusive region based scheduling scheme for mmWave WPANs with directional antenna. *IEEE Trans Wireless Commun* 9(1):113–121
27. Kompella S, Mao S, Hou YT, Sherali HD (2009) On path selection and rate allocation for video in wireless mesh networks. *IEEE Trans Networking* 17(1):212–224
28. Takagi H, Kleinrock L (1985) A tutorial on the analysis of polling systems. Computer Science Department, UCLA, Tech. Rep. No. 850005
29. He Z, Mao S, Rappaport TS (2015) Minimum time length link scheduling under blockage and interference in 60GHz networks. In: Proceedings of IEEE WCNC 2015, New Orleans, LA, pp 1–6
30. Rappaport TS, Sun S, Mayzus R, Zhao H, Azar Y, Wang K, Wong GN, Schulz JK, Samimi M, Gutierrez F (2013) Millimeter wave mobile communications for 5G cellular: It will work! *IEEE Access J* 1(1):335–349
31. Sun S, Rappaport TS, Heath RW, Nix A, Rangan S (2014) MIMO for millimeter-wave wireless communications: beamforming, spatial multiplexing, or both? *IEEE Commun Mag* 52(12):110–121