On Frame-based Scheduling for Directional mmWave WPANs

In Keun Son[†] Shiwen Mao[‡] Michelle X. Gong[§] Yihan Li[‡]
[†]Defense Acquisition Program Administration, Seoul, South Korea
[‡]Dept. Electrical & Computer Engineering, Auburn Univ., Auburn, AL 36849-5201, USA
[§]Intel Labs, Intel Corporation, Santa Clara, CA 95054-1549, USA

***Abstract*—Millimeter wave (mmWave) communications in the 60 GHz band can provide multi-gigabit rates for emerging bandwidth-intensive applications, and has thus gained considerable interest recently. In this paper, we investigate the problem of efficient scheduling in mmWave wireless personal area networks (WPAN). 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 WPANs. 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. The proposed FDMAC is analyzed and evaluated under various traffic models and patterns. Its superior performance is validated with extensive simulations.**

I. Introduction

In recent years, millimeter wave (mmWave) communications in the 60 GHz band has gained considerable interest from academia, industry, and standards bodies. This is due to the huge unlicensed bandwidth (i.e., up to 7 GHz) that is available in this band in most parts of world. With this massive unlicensed bandwidth, many bandwidth-demanding new applications can be easily supported in mmWave wireless networks. Several standards have been or are being defined to achieve multi-gigabit rates for 60 GHz networks, such as ECMA-387 [1] and IEEE 802.15.3c [2].

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 of 5 GHz signals, and atmospheric absorption for 60 GHz signals ranges from 15 to 30 dB/km [3]. As a result, the range of an mmWave link is limited. The 60 GHz systems have been shown viable for short range communications [4], [5], such as wireless personal area networks (WPAN) that may require high data rate transmissions over short distances (e.g., HDTV or mass storage synchronization). 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 WPANs.

As a result of directional listening and transmission, the signal strength is usually very low at third party nodes that are not involved in the current transmission, making it difficult to perform carrier sense. This is often referred to as the deafness problem [6]. 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 [7], the links in an mmWave network can be regarded as *pseudo-wired*, i.e., with negligible interference to each other. New scheduling algorithms and directional MAC protocols are needed to ensure communicating nodes meet each other for scheduled transmissions, and to exploit concurrent wired-like transmissions for maximizing spatial reuse.

In this paper, we investigate efficient scheduling in mmWave WPANs. We consider an mmWave WPAN consisting of a piconet coordinator (PNC) and multiple devices (DEVs) [2]. Assuming some bootstrapping mechanism [8], 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 DEVs and computes schedules to enable concurrent directional transmissions in the WPAN. We consider *frame-based scheduling* in this paper and term the proposed protocol *frame-based scheduling directional MAC* (FDMAC). The objective is to leverage collision-free concurrent transmissions to fully exploit spatial reuse in mmWave WPANs.

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 and 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

978-1-4673-0775-8/12/$31.00 ©2012 IEEE

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 evaluate FDMAC with extensive simulations, considering various traffic models and uniform and non-uniform traffic patterns. The simulation results validate the superior performance of FDMAC, which is shown to outperform two state-of-the-art mmWave MAC schemes with regard to both delay and throughput performance, while achieving comparable fairness performance.

The remainder of this paper is organized as follows. We discuss related work in Section II. The system model and problem statement are given in Section III. We present the GC algorithm and its enhancement in Section IV. Simulation results are given in Section V. Section VI concludes this paper.

II. Related Work

There have been considerable work on directional MAC protocols in the literature. For example, see [6], [9], [10]. Conventional directional MAC protocols may not be directly used in mmWave WPANs, since the unique channel properties in the mmWave band have not been fully considered in these schemes [4], [11]. Some proposals require high control overhead or have high complexity, while others rely on omnidirectional communications for control messages, which may not be feasible for mmWave systems that operates in the multi-gigabit domain with highly direction transmissions [4].

Both standards for 60 GHz WPANs, i.e., ECMA-387 [1] and IEEE 802.15.3c [2], adopt Time Division Multiple Access (TDMA) for data communications. Two existing MAC protocols recently proposed for 60 GHz networks [12], [13] are also based on TDMA. Due to the bursty nature of data traffic, the required medium time is often highly unpredictable. A TDMA-based MAC 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 [11]. 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 [14]. Most data transmissions are via the PNC with a sequential polling policy. When the direct link toward the PNC is not available due to obstacles, the PNC selects an intermediate node to relay the traffic. MRDMAC solves the deafness problem by PNC's weighted round robin scheduling and overcomes obstructed 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) [15] and directional-to-directional MAC (DtDMAC) [16] 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.

III. System Model and Problem Statement

A. System Model

Consider an mmWave WPAN consisting of a PNC and multiple DEVs [2]. 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 WPAN to accommodate traffic requests from the DEVs.

Let there be n DEVs in the WPAN and one DEV, denoted as DEV1, is selected to serve as the PNC. We assume a bootstrapping scheme in place (e.g., see [8]) 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 (e.g., in the scheduling phase, see Section III-B), all the DEVs point their beams to the PNC. The PNC polls traffic requests from the DEVs and pairs them for concurrent 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.

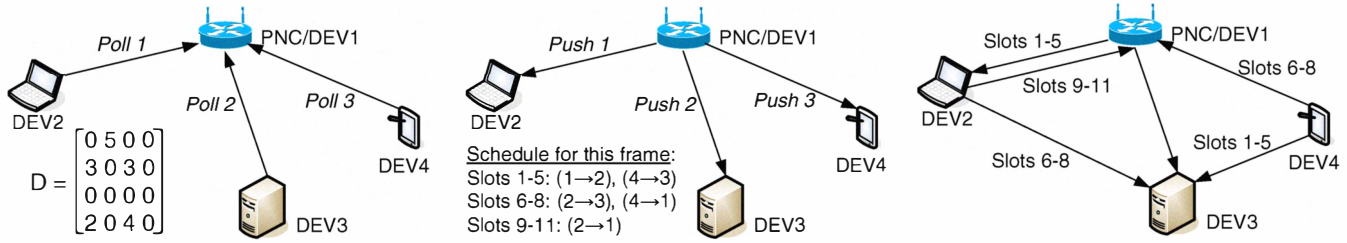
The directional connectivity formed in each time slot can be modeled as a directed graph (or, *digraph*). Let $G^t(V, E^t)$ be such a connectivity graph in time slot t, where V is the set of nodes and E^t is the set of directional links. If node i is scheduled to transmit to node j in time slot t, we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

B. Frame-based Scheduling and Matching

The MAC protocol should leverage collision-free concurrent transmissions to fully exploit spatial reuse in directional mmWave WPANs. We focus on *frame-based scheduling* in this paper. 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 the PNC and DEVs start concurrent transmissions following the schedule.

In the n-node WPAN, every node maintains $n-1$ *virtual queues*, each storing the packets destined to a distinct neighbor.



(a) PNC polls traffic demands from the DEVs to construct the traffic demand matrix D

(b) PNC computes a transmission schedule for demand D and pushes the schedule to the DEVs

(c) DEVs start concurrent transmissions following the schedule received from the PNC

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

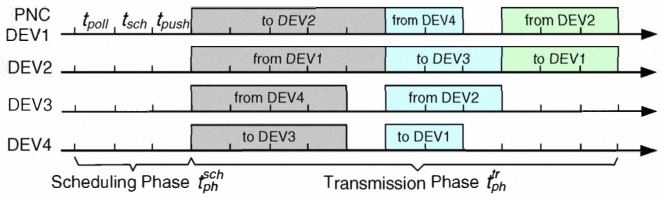


Fig. 2. Time-line illustration of FDMAC operation with the traffic demand D and schedule given in Fig. 1.

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) For the WPAN, the traffic demand matrix D is an $n \times n$ matrix, where the ith row is the traffic demand vector d_i, for all i.

The operation of FDMAC is illustrated in Figs. 1 and 2. *In FDMAC, traffic received in a frame interval is first cumulated in the virtual queues, and then gets transmitted in the next frame.* As shown in Fig. 2, 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. 1(a), 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 time t_{poll} for the PNC to collect all the d_i's and assemble the traffic demand matrix D, as shown in Fig. 1(a).

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. 1(b), 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. 1(c). 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 29 time slots).

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. The time-line operation of FDMAC is illustrated in Fig. 2 for the example in Fig. 1.

C. Properties

Definition 4: (*Schedule graph* G^t) In each time slot t, the connectivity of the WPAN, 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 [17].

In each time slot t, the WPAN 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 WPAN topology evolves from time slot to time slot as different pairings of the nodes are formed.

Proposition 1: The digraph of the mmWave WPAN 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 [17]. 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. ■

Corollary 0.1: 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 WPAN is upper bounded. Due to the concurrent transmissions, the WPAN capacity is upper bounded by $\lfloor \frac{n}{2} \rfloor \geq 100\%$, for $n \geq 4$.

D. 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 jth virtual queue at node i. Then we have

Proposition 2: The minimum number of required time slots for $G^t(V, E^t)$ is $\delta^t = \max\left\{d_{ij} | a^t_{ij} = 1, i, j \in V\right\}$, where a^t_{ij} 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

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

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

For a given traffic demand matrix D, there are many feasible schedules satisfying (2). An optimal schedule should clear the backlog D with minimum number of time slots, which implies maximum parallelism of transmissions. We formulate the optimal frame-based scheduling problem as:

$$\text{minimize:} \quad \sum_{k=1}^{K} \delta^k \quad (3)$$

subject to:

$$\sum_{k=1}^{K} a^k_{ij} = \begin{cases} 1, & \text{if } d_{ij} > 0 \\ 0, & \text{otherwise} \end{cases} \quad \text{for all } i, j \quad (4)$$

$$a^k_{ij} \in \begin{cases} \{0, 1\}, & \text{if } d_{ij} > 0 \\ \{0\}, & \text{otherwise} \end{cases} \quad \text{for all } i, j, k \quad (5)$$

$$\sum_{k=1}^{K} (\delta^k \cdot a^k_{ij}) \begin{cases} \geq d_{ij}, & \text{if } d_{ij} > 0 \\ = 0, & \text{otherwise} \end{cases} \quad \text{for all } i, j \quad (6)$$

$$\sum_{j=1}^{n} (a^k_{ij} + a^k_{ji}) \leq 1 \quad \text{for all } i, k. \quad (7)$$

This is an MINLP problem, which is generally NP-hard. Constraints (4) and (5) indicate that each directional link should be scheduled once. Constraint (6) specifies that the number of time slots for links should be equal to or larger than the backlog. Constraint (7) indicates that each schedule graph should be a matching, as given in Theorem 1.

In a recent work [7], the authors prove by analysis and simulations that interference can be essentially ignored in mmWave MAC design, due to the highly directional antennas required at both the transmit and receive nodes; "the links in the network can be thought of as 'pseudo-wired'" [7]. We thus focus on the scheduling problem in this paper. If in some cases such pseudo-wired assumption is not exactly true (or when there is reflection [18]), the concept of *exclusive region* introduced in [19] can be incorporated in the problem formulation, as additional linear constraints in Problem (3) to exclude such interfering links from being scheduled simultaneously.

Problem (3) has a nonlinear constraint (6). We apply a relaxation technique, the *Reformuation-Linearization Technique* (RLT), to obtain a linear relaxation [20]. Specifically, we define a substitution variable $\mu^k_{ij} = \delta^k \cdot a^k_{ij}$. 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^k_{ij} is bounded as $0 \leq a^k_{ij} \leq 1$ due to constraint (5). We can obtain the following *RLT bound-factor product constraints* for μ^k_{ij}:

$$\begin{cases} \mu^k_{ij} \geq 0 \\ \delta^k - \mu^k_{ij} \geq 0 \\ \bar{d} \cdot a^k_{ij} - \mu^k_{ij} \geq 0 \\ -\delta^k - \bar{d} \cdot a^k_{ij} + \mu^k_{ij} \geq -\bar{d} \end{cases} \quad \text{for all } i, j, k. \quad (8)$$

Substituting μ^k_{ij} into (6), we obtain a mixed integer linear programming (MILP) relaxation as:

$$\text{minimize:} \quad \sum_{k=1}^{K} \delta^k \quad (9)$$

subject to:

$$\sum_{k=1}^{K} \mu^k_{ij} \begin{cases} \geq d_{ij}, & \text{if } d_{ij} > 0 \\ = 0, & \text{otherwise} \end{cases} \quad \text{for all } i, j \quad (10)$$

Constraints (4), (5), and (7)

RLT bound-factor product constraints (8).

E. Example

As an example, consider a 5-node mmWave WPAN. At the beginning of a frame, the traffic demand matrix is:

$$D = \begin{bmatrix} 0 & 4 & 0 & 9 & 0 \\ 7 & 0 & 5 & 0 & 0 \\ 0 & 8 & 0 & 0 & 6 \\ 0 & 1 & 4 & 0 & 0 \\ 10 & 0 & 0 & 3 & 0 \end{bmatrix}. \quad (11)$$

We can schedule at most two concurrent transmissions in any time slot according to Corollary 0.1. Each matrix A^t can have at most two 1-valued elements, and the network capacity is upper bounded by 200% (i.e., twice the nominal link capacity). We solve problem (9) using an open-source MILP solver, *lp_solve* ver 5.5.2.0 [21]. The solution is S_1 as given in (12). The schedule consists of six matrices. To clear the backlog D, $\sum_{k=1}^{6} \delta^k = 34$ time slot are needed with this schedule.

The drawback of using the optimization software, however, is the significantly long computation time associated with solving the MILP problem. To obtain an optimal solution for the 5-node network, it takes about 410 s, which is not practical for mmWave WPANs where time slot durations are only a few μs. In the next section, we develop a computationally efficient greedy algorithm, termed *Greedy Coloring (GC) Algorithm*, for our FDMAC protocol that can compute near-optimal schedules.

IV. Greedy Coloring Algorithm

As discussed, a frame in FDMAC consists of a scheduling phase (with duration t^{sch}_{ph}) and a transmission phase (with variable duration t^{tr}_{ph}). The scheduling phase represents the overhead of FDMAC, while in the transmission phase, there is

$$D \prec S_1 = 4\begin{bmatrix} 0&1&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&0&0 \end{bmatrix} + 9\begin{bmatrix} 0&0&0&1&0\\ 0&0&1&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} + 7\begin{bmatrix} 0&0&0&0&0\\ 1&0&0&0&0\\ 0&0&0&0&1\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{bmatrix} + 10\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&1&0&0&0\\ 0&0&0&0&0\\ 1&0&0&0&0 \end{bmatrix} + 1\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&1&0&0&0\\ 0&0&0&0&0 \end{bmatrix} + 3\begin{bmatrix} 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&1&0 \end{bmatrix}. \quad (12)$$

no need for additional overhead such as RTS/CTS handshakes. If the frame is sufficiently long, the t_{ph}^{sch} overhead will be amortized over a long sequence of successive data transmissions, thus having the potential to achieve higher throughput performance over existing approaches, a similar idea to the design of PSMAC in [22]. The core of FDMAC is a scheduling algorithm that can solve Problem (3) in real-time (i.e., with small t_{sch}), which is presented in this section.

A. Greedy Coloring Algorithm

The GC algorithm computes a near-optimal solution for a given traffic demand matrix D. We show that the scheduling problem can be modeled as an edge coloring problem.

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

Proof: The traffic demand matrix D can be modeled as a directed and weighted multigraph, $G(V, E)$, where V is the set of distinct nodes and E is a set of directed edges. For each edge $e_{ij} \in E$, there is a non-zero traffic demand from its initial vertex to its terminal vertex, and the edge weight $\omega(e_{ij})$ is set to the corresponding traffic demand d_{ij}. In addition, graph G has 2 multiplicity due to half-duplex communications.

Recall that each A^t, $1 \le t \le K$, in S represents the adjacent matrix of the schedule graph G^t, and δ^t is the highest weight among all the edges in G^t (see Proposition 2). If we assign K distinct colors to each graph G^t, all the edges in the same graph have identical color. According to (4), each edge can be scheduled only once, i.e., each edge can have only one of the K colors. Furthermore, each graph G^t is a matching according to Proposition 1. Thus each vertex cannot have multiple colors on its edges. Therefore, the edges of a vertex must have distinct colors. We conclude that *proper edge coloring* is possible if there is a feasible solution. ■

According to Proposition 3, 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 $\sum_{k=1}^{K} \theta^k$, where θ^k is the highest weight in the kth 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 is the minimum number of colors needed to color all edges in G. Our objective is to minimize the total transmission time $\sum_{k=1}^{K} \theta^k$, 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.

The pseudo-code of the GC algorithm is given in Table I. It has the worst case complexity of $O(|E|^2)$. The GC algorithm first obtains a directed and weighted multigraph $G(V, E)$,

TABLE I
GREEDY COLORING ALGORITHM

```
1:  Obtain a directed and weighted multigraph G(V, E):
2:    - Input: traffic demand matrix D
3:    - V: set of n distinct vertices
4:    - E: set of edges ordered by non-increasing weights
5:  t = 0;
6:  WHILE (|E| > 0)
7:    t = t + 1;
8:    Set G^t(V^t, E^t) with V^t = ∅ and E^t = ∅;
9:    WHILE (there is unvisited edge in E and |E^t| ≤ ⌊n/2⌋)
10:     Get the unvisited edge with the largest weight, e_ij ∈ E;
11:     IF (i ∉ V^t and j ∉ V^t)
12:       E^t = E^t ∪ {e_ij};
13:       V^t = V^t ∪ {i, j};
14:       E = E − {e_ij};
15:       IF (|E^t| == 1) θ^t = ω(e_ij); END IF
16:     END IF
17:   END WHILE
18:   Output G^t(V, E^t) and θ^t;
19: END WHILE
```

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 9 in Table I 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 given in Corollary 0.1. Line 11 is a matching condition to guarantee a set of pairwise nonadjacent edges.

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, $\Lambda_{ij}^G = 0$. It can be shown that the number of colors K achieved by the GC algorithm is bounded as

$$\Delta_v \le K \le \Delta_p \le 4n - 6, \quad (13)$$

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 proof is omitted due to lack of space.

B. Enhanced FDMAC

In the proposed FDMAC, packets arriving in the current frame are stored in the virtual queues first, and then will be scheduled and transmitted in the next frame. This is analogous to the *gated* service policy in polling systems [22]. In a feasible schedule, δ^k is chosen to be the largest traffic demand among the scheduled links, as given in Proposition 2.

The GC algorithm can be enhanced by allowing transmitting packets that arrive during the current frame whenever possible. For instance, see the example in Section III-B (Figs. 1 and 2), where five time slots are allocated to the transmission from DEV4 to DEV3. Since the current traffic demand is $d_{43} = 4$, a packet that arrives at DEV4 before the 5th time slot and destined for DEV3, can be transmitted using the remaining time slot, such that it does not need to wait in the virtual queue to be transmitted in the next frame. Such an enhancement can further improve both the delay and throughput performance of FDMAC, especially in the congestion region. We evaluate the potential gain achieved by the enhancement in Section V.

C. Example

We apply the GC algorithm to the same traffic demand matrix D given in (11). It took GC 4.3 μs to compute a schedule S_2 as given in (14). S_2 is a 6-edge-coloring solution for D, requiring 36 time slots in total. According to (13), the number of colors is bounded as $5 \leq K \leq 8$. Comparing to the optimal solution given in Section III-E, the GC schedule has the same number of topology permutation (i.e., $K = 6$), but uses two additional time slots for the same traffic demand D. The computation complexity, however, is greatly reduced, from hundreds of seconds to 4.3 μs, making GC a practical scheme for mmWave WPANs.

V. Simulation Studies

A. Simulation Setup

We evaluate FDMAC and compare it with two existing schemes, i.e., MRDMAC [14] and MDMAC [15]. All the protocols are implemented in MATLAB, except that the FDMAC execution times presented in Section V-B1 are measured with a C implementation. 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 to idle mode, to 0.01 and T_{ESR}, the threshold of explicit state reset, to 90% as given in [15].

We adopt the same simulation parameters given in Table II of [14], with a 2 Gbps data rate. We assume fixed data packet size of 1,000 bytes. The duration of a time slot is

$$t_{slot} = t_{pkt} + t_{SIFS} + t_{ACK} \approx 5 \ \mu\text{s},$$

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.

In FDMAC, we need to calculate t_{ph}^{sch}, consisting of t_{poll}, t_{sch}, and t_{push} as shown in Fig. 2. 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.

TABLE II
Execution Time of the GC Algorithm

Network Size	*Exe. Time* (μs)	*Network Size*	*Exe. Time* (μs)
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

We compare the three mmWave WPAN MAC schemes with respect to average delay, network throughput, and fairness performance under the following two traffic types:

- *i.i.d. Bernoulli traffic*: a packet arrives in each time slot with a predetermined probability.
- *On-off bursty traffic*: packets are generated following an on-off Markov chain model with geometrically distributed on and off periods.

When a packet is generated at a source node, its destination is randomly chosen from other nodes in the WPAN. We consider both *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 $\alpha > 0.5$.

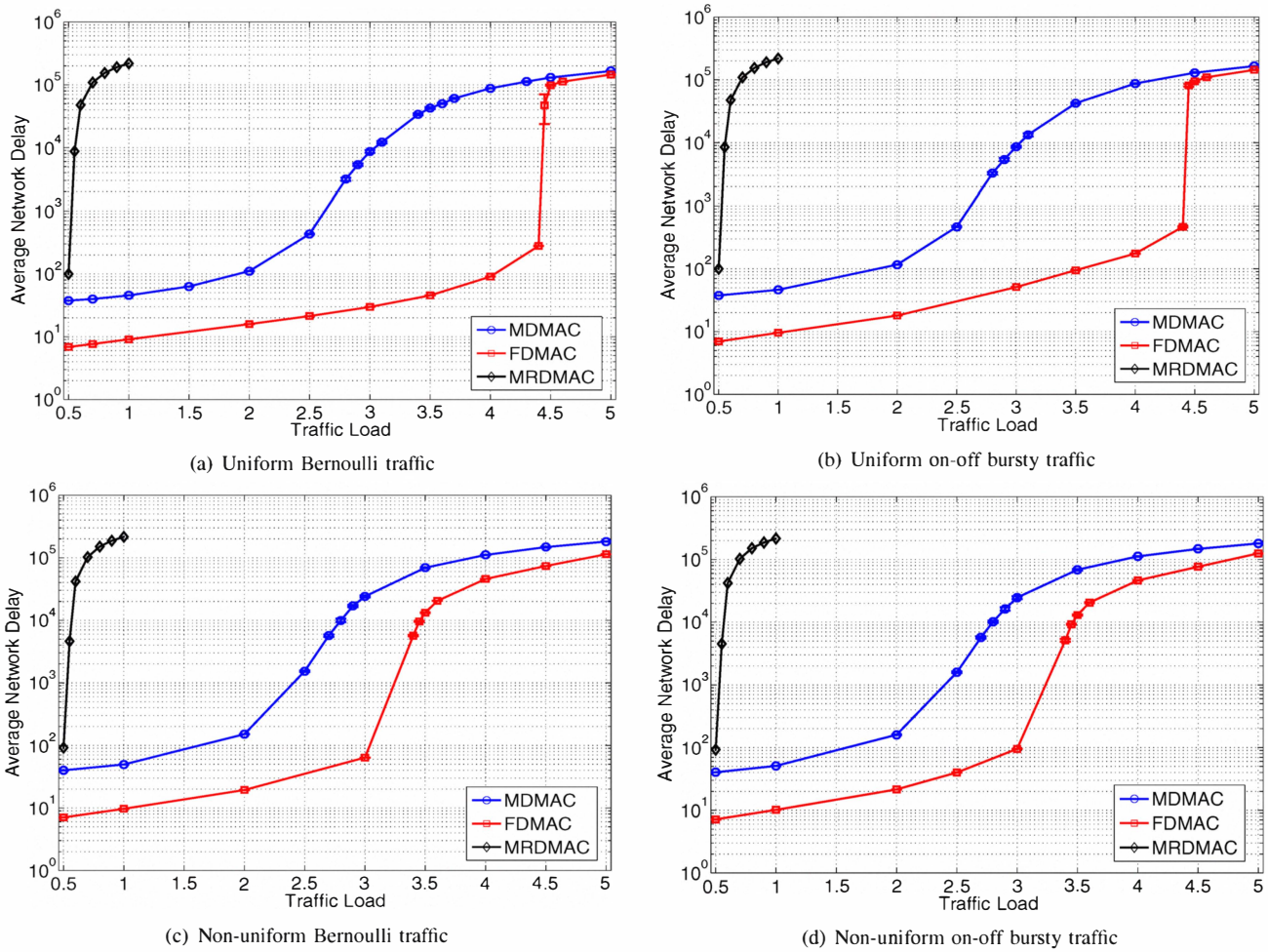
B. Performance of FDMAC

1) 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 II. 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 WPAN 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.

2) Enhanced FDMAC: We also implemented the enhanced version of FDMAC that schedules packets arriving in the current frame whenever possible, as discussed in Section IV-B.

$$D \prec S_2 = 10 \begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&0&0&0 \\ 1&0&0&0&0 \end{bmatrix} + 9 \begin{bmatrix} 0&0&0&1&0 \\ 0&0&0&0&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \end{bmatrix} + 7 \begin{bmatrix} 0&0&0&0&0 \\ 1&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&0&0 \end{bmatrix} + 5 \begin{bmatrix} 0&0&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&1&0 \end{bmatrix} + 4 \begin{bmatrix} 0&1&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \end{bmatrix} + 1 \begin{bmatrix} 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&1&0&0&0 \\ 0&0&0&0&0 \end{bmatrix}. \quad (14)$$



(a) Uniform Bernoulli traffic

(b) Uniform on-off bursty traffic

(c) Non-uniform Bernoulli traffic

(d) Non-uniform on-off bursty traffic

Fig. 4. Average delays of the three MAC protocols under uniform and non-uniform traffic patterns.

The simulation results are presented in Fig. 3, where *Throughput Ratio* is defined to be the total amount of received traffic over the total amount of transmitted traffic. We find FDMAC and enhanced FDMAC achieve similar throughput in the low-load region and in the heavily congested region. However, the enhanced FDMAC achieves an 8.6% gain over FDMAC under incipient congestion when the offered load is 445%.

C. Comparison with Existing Schemes

We next compare FDMAC with two state-of-the-art directional MAC protocols for mmWave WPANS with a 10-node mmWave WPAN. Initially, each DEV has a randomly generated small amount of packets in the virtual queues.

In Figs. 4 and 5, 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 α =40% of the offered load. We set the delay threshold to 10^4, and discard packets with a delay larger 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. We find most of the error bars are negligible in the figures.

1) Delay Performance: We first evaluate the delay performance of the three MAC protocols. In Fig. 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 all the four plots in Fig. 4, the average delay of MRDMAC diverges when the offered load is between 0.55 and 0.6.

MDMAC demonstrates similar results regardless of traffic models and traffic patterns. The MDMAC delay diverges when the offered load is 3.05 for uniform Bernoulli/bursty

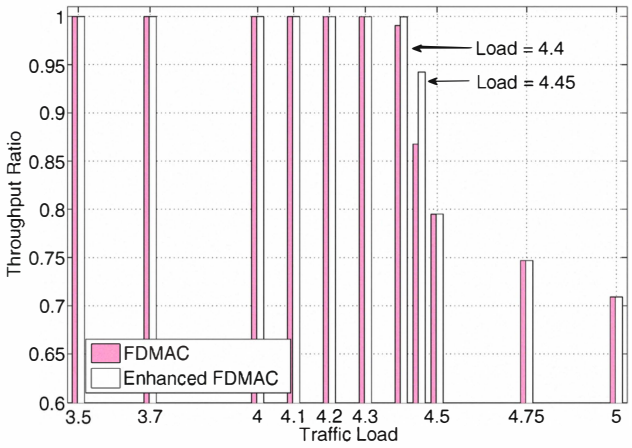


Fig. 3. FDMAC versus the enhanced FDMAC.

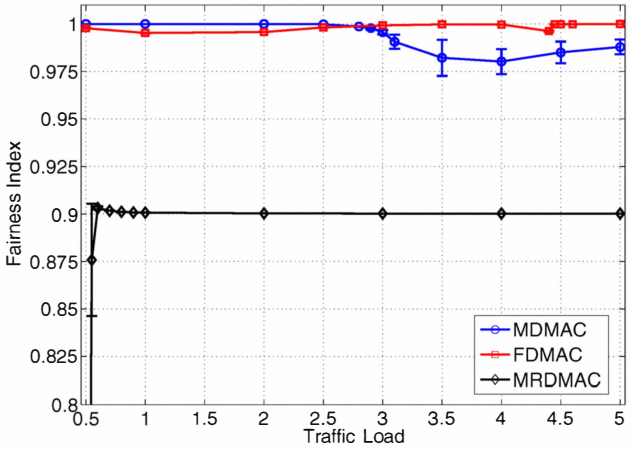


Fig. 6. Fairness performance under uniform on-off bursty traffic.

traffic and 2.8 for non-uniform Bernoulli/bursty traffic. In all the four 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 Bernoulli/bursty traffic, which is very close to the upper bound given in Corollary 0.1, and 3.45 for non-uniform Bernoulli/bursty traffic, which is still 65% higher than that achieved by MDMAC.

2) Throughput Performance: The total number of successful transmissions achieved by the three MAC protocols are plotted in Fig. 5. 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 300%. FDMAC achieves a throughput close to 345% under the non-uniform traffic pattern, while the MDMAC throughput is close to 280% in this case. The normalized improvement ratios are 50% and 23.2%, respectively.

Under uniform traffic, 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 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.

3) Fairness Performance: Finally, we examine the fairness performance of the three MAC protocols. We adopt the fairness index defined in [23], as

$$f(w_1, w_2, ..., w_n) = \left(\sum_{i=1}^{n} w_i\right)^2 / \left(n \sum_{i=1}^{n} w_i^2\right), \qquad (15)$$

where w_i is the average delay at node i. The fairness index varies from 0 (worst) to 1 (best).

Due to lack of space, we only show the fairness results for the uniform on-off bursty traffic case in Fig. 6. 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 plot.

VI. Conclusion

In this paper, we presented FDMAC for mmWave WPANs. The main idea is to exploit concurrent transmissions to greatly improve network capacity, as enabled by the highly directional transmissions in the 60 GHz band. The proposed FDMAC is a frame-based MAC protocol that incorporates a GC algorithm to amortize the control overhead over a long sequence of concurrent packet transmissions. We analyzed the FDMAC protocol and evaluated its performance with extensive simulations. It is shown to outperform two state-of-the-art schemes with regard to delay and throughput performance, while achieving comparable fairness performance.

Acknowledgment

Part of this work was conducted when Dr. In Keun Son was pursuing a Ph.D. degree at Auburn University. This work is supported in part by the US National Science Foundation (NSF) under Grants CNS-0953513, ECCS-0802113, CNS-1145446, IIP-1127952, and DUE-1044021, and through the NSF Wireless Internet Center for Advanced Technology. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Foundation.

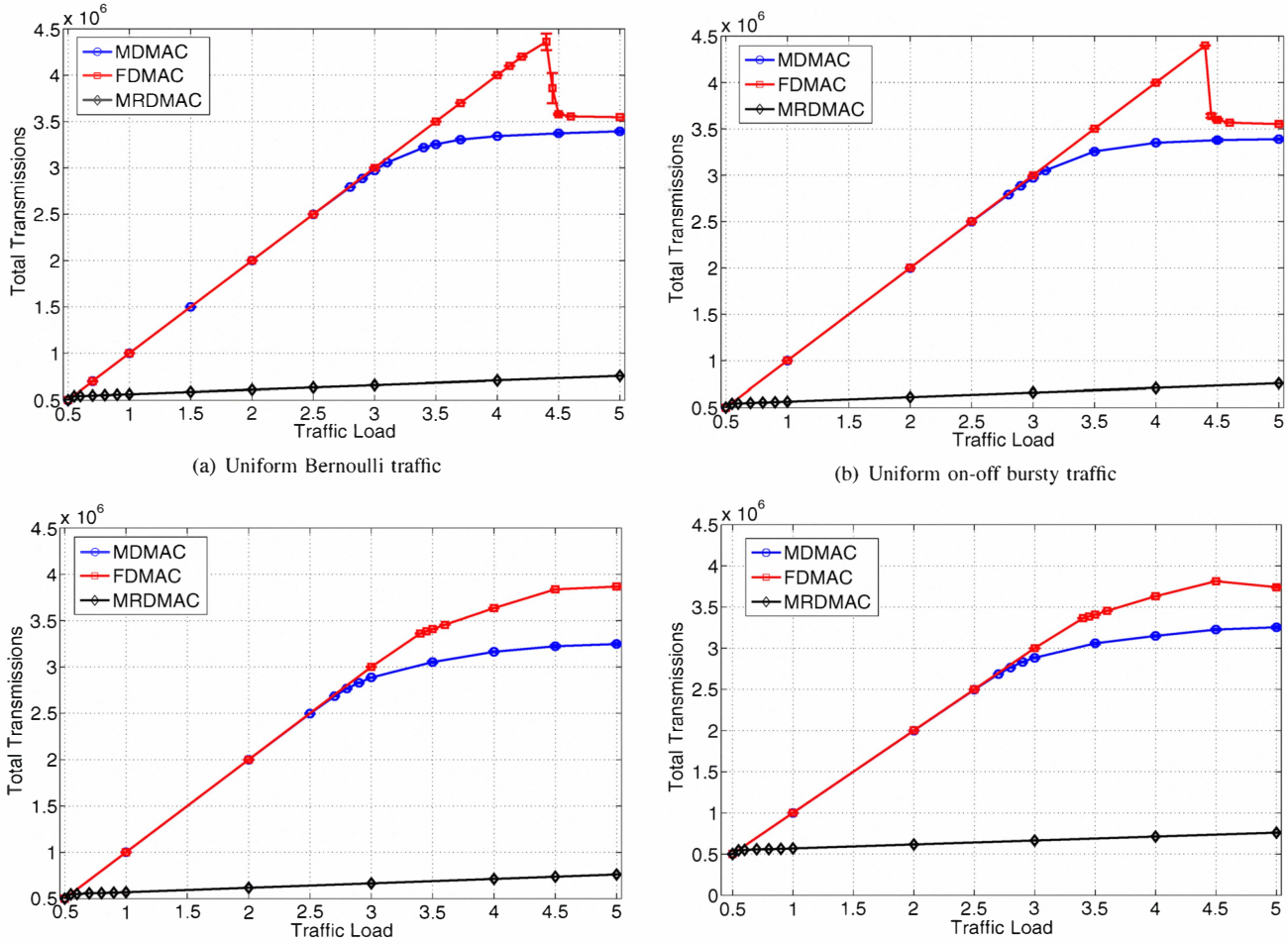
References

[1] ECMC TC48, ECMA standard 387, "High rate 60 GHz PHY, MAC and HDMI PAL," Dec. 2008.

[2] 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, Jun. 2009.

[3] R. C. Daniels and R. W. Heath, "60 GHz wireless communications: Emerging requirements and design recommendations," *IEEE Veh. Tech. Mag.*, vol. 2, no. 3, pp. 41–50, Sept 2007.

[4] S. Geng, J. Kivinen, X. Zhao, and P. Vainikainen, "Millimeter-wave propagation channel characterization for short-range wireless communications," *IEEE Trans. Veh. Tech.*, vol. 58, no. 1, pp. 3–13, Jan. 2009.



(a) Uniform Bernoulli traffic

(b) Uniform on-off bursty traffic

(c) Non-uniform Bernoulli traffic

(d) Non-uniform on-off bursty traffic

Fig. 5. Total number of successful packet transmissions of the three MAC protocols under uniform and non-uniform traffic patterns.

[5] M. Park, C. Cordeiro, E. Perahia, and L. L. Yang, "Millimeter-wave multi-gigabit WLAN: Challenges and feasibility," in *Proc. IEEE PIMRC'08*, Cannes, France, Sept. 2008, pp. 1–5.

[6] Y.-B. Ko, V. Shankarkumar, and N. H. Vaidya, "Medium access control protocols using directional antennas in ad hoc networks," in *Proc. IEEE INFOCOM*, New York, NY, 1999, pp. 13–21.

[7] R. Mudumbai, S. Singh, and U. Madhow, "Medium access control for 60 GHz outdoor mesh networks with highly directional links," in *Proc. IEEE INFOCOM 2009 (Mini Conference)*, Rio de Janeiro, Brazil, Apr. 2009, pp. 2871–2875.

[8] J. Ning, T.-S. Kim, S. V. Krishnamurthy, and C. Cordeiro, "Directional neighbor discovery in 60 GHz indoor wireless networks," in *Proc. ACM MSWiM '09*, Tenerife, Canary Islands, Spain, 2009, pp. 365–373.

[9] M. Takai, J. Martin, A. Ren, and R. Bagrodia, "Directional virtual carrier sensing for directional antennas in mobile ad hoc networks," in *Proc. ACM MobiHoc*, Lausanne, Switzerland, 2002, pp. 183–193.

[10] T. Korakis, G. Jakllari, and L. Tassiulas, "CDR-MAC: A protocol for full exploitation of directional antennas in ad hoc wireless networks," *IEEE Trans. Mobile Comp.*, vol. 7, no. 2, pp. 145–155, Feb. 2008.

[11] M. X. Gong, R. J. Stacey, D. Akhmetov, and S. Mao, "Performance analysis of a directional CSMA/CA protocol for mmWave wireless PANs," in *Proc. IEEE WCNC'10*, Sydney, Australia, Apr. 2010.

[12] X. An and R. Hekmat, "Directional MAC protocol for millimeter wave based wireless personal area networks," in *Proc. IEEE VTC-Spring'08*, Singapore, May 2008, pp. 1636–1640.

[13] C.-W. Pyo, F. Kojima, J. Wang, H. Harada, and S. Kato, "MAC enhancement for high speed communications in the 802.15.3c mmWave WPAN," *Springer Wireless Pers. Commun.*, vol. 51, no. 4, pp. 825–841, July 2009.

[14] S. Singh, F. Ziliotto, U. Madhow, E. M. 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, Oct. 2009.

[15] S. Singh, R. Mudumbai, and U. Madhow, "Distributed coordination with deaf neighbors: Efficient medium access for 60 GHz mesh networks," in *Proc. IEEE INFOCOM*, San Diego, CA, Mar. 2010, pp. 1–9.

[16] E. Shihab, L. Cai, and J. Pan, "A distributed asynchronous directional–to–directional MAC protocol for wireless ad hoc networks," *IEEE Trans. Veh. Tech.*, vol. 58, no. 9, pp. 5124–5134, Nov. 2009.

[17] J. L. Gross and J. Yellen, *Graph Theory and Its Applications*. CRC Press, 1999.

[18] E. Ben-Dor, T. S. Rappaport, Y. Qiao, and S. J. Lauffenburger, "Millimeter-wave 60 Ghz outdoor and vehicle AOA propagation measurements using a broadband channel sounder," in *Proc. IEEE GLOBECOM'11*, Houston, TX, Dec. 2011.

[19] L. X. Cai, L. Cai, X. Shen, and J. W. Mark, "REX: a Randomized EXclusive region based scheduling scheme for mmWave WPANs with directional antenna," *IEEE Trans. Wireless Commun*, vol. 9, no. 1, pp. 113–121, Jan. 2010.

[20] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, "On path selection and rate allocation for video in wireless mesh networks," *IEEE Trans. Networking*, vol. 17, no. 1, pp. 212–224, Feb. 2009.

[21] *lp_solve* Reference Guide Menu [Online]. Available: http://lpsolve.sourceforge.net.

[22] Y. Li, S. Mao, S. S. Panwar, and S. F. Midkiff, "On the performance of distributed polling service-based wireless MAC protocols," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4635–4645, Nov. 2008.

[23] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," Feb. 1999, ATM Forum/99-0045.