Efficient and Effective Proactive Scheduling for mmWave WLANs

Ang Deng
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia, USA
adeng3@gatech.edu

Douglas M. Blough
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia, USA
doug.blough@ece.gatech.edu

ABSTRACT

To cope with growing wireless bandwidth demand, millimeter wave (mmWave) communication has been identified as a promising technology to deliver Gbps throughput. However, due to the susceptibility of mmWave signals to blockage, applications can experience significant performance variability as users move around due to rapid and significant variation in channel conditions. In this context, proactive schedulers that make use of future data rate prediction have potential to bring a significant performance improvement as compared to traditional schedulers. In this work, we propose an efficient proactive algorithm that prioritizes the scheduling of scarce resources to achieve better performance than traditional schedulers. The results show that our scheduler can increase average data rate by up to 20% compared to non-proactive scheduling and achieves from 60% to 75% of the performance gain of an optimal proactive scheduler.

CCS CONCEPTS

• **Networks** → **Packet scheduling**; **Wireless local area networks**.

KEYWORDS

mmWave, proactive scheduling, WLAN

ACM Reference Format:
Ang Deng and Douglas M. Blough. 2023. Efficient and Effective Proactive Scheduling for mmWave WLANs. In *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM '23), October 30-November 3, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3616388.3617537

1 INTRODUCTION

For next-generation wireless networks, both in the cellular and wireless LAN domains, mmWave communications are considered a key technology due to the plentiful bandwidth available in mmWave bands. However, a major technical challenge with deploying mmWave technology in practical settings is the susceptibility of mmWave signals to blockage due to their short wavelengths. It is well established that mmWave signals are completely blocked by relatively small obstacles including even the human body [13, 14, 19].

MSWiM '23, October 30-November 3, 2023, Montreal, QC, Canada

ACM ISBN 979-8-4007-0366-9/23/10.
https://doi.org/10.1145/3616388.3617537

As a result of the blockage issue, the link quality of a mmWave link can vary rapidly and widely as a user moves around within a space containing obstacles. Meanwhile, many next-generation applications such as ultra-high-definition real-time video and wireless virtual reality require continuous high-quality wireless links. One potential approach to help deal with this issue is to schedule packet transmissions when channel conditions are good and high date rates can be achieved. Such an approach has been referred to as *proactive scheduling*.

Prior work has considered proactive scheduling by making use of blockage predictions on the order of milliseconds by detecting when link quality is beginning to degrade[9]. However, when blockages are primarily due to static obstacles such as furniture items, it should be possible to predict blockages farther into the future if knowledge of the environment can be combined with user mobility prediction, thereby opening up greater potential performance improvements with proactive scheduling.

Our prior work formally defined a proactive scheduling problem with fairness constraints and designed an optimal scheduling algorithm to solve it [7]. Although the optimal scheduler was impractical for real-time scheduling since it involves solving an integer linear programming (ILP) problem, this allowed us to evaluate the upper limits of what could be possible with proactive scheduling. Results showed that very large performance gains could be achieved if blockages can be predicted several seconds ahead of time.

In this paper, we provide a first step toward making proactive scheduling on a longer time horizon practical in the mmWave WLAN context. We develop an efficient greedy heuristic proactive scheduler that can operate in real time. We then evaluate the performance of both the heuristic and optimal proactive schedulers using a realistic hot-spot-based mobility model and with both a very simple mobility prediction scheme and a perfect mobility prediction scheme. Results show that the heuristic scheduler can achieve from 60% to 75% of the performance gain of the optimal proactive scheduler with a running time that is several orders of magnitude faster. The results also show that proactive scheduling outperforms non-proactive scheduling even with very simple mobility prediction, but to realize the full potential of proactive scheduling will require good state-of-the-art mobility prediction schemes to be employed.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 presents the system model and problem formulation. Section 4 describes our proposed efficient heuristic proactive scheduling approach. Section 5 presents numerical evaluation results of the proposed scheduler and the analysis. Lastly, Section 6 discusses key findings and future work, concluding the paper.

2 RELATED WORK

The idea of proactive scheduling is not new to wireless networking. Previous works have studied proactive scheduling in non-mmWave wireless networking [3, 15, 18]. These works leverage the predictability of user channel state, and each develop their own extended form of the proportional fair scheduler which utilizes this future knowledge. In [3], the exact methods of predicting future data rate is not introduced but assumed, whereas [15, 18] presents their own method for estimating the user data rate by leveraging the channel fading effect. Their results consistently agree that proactive scheduling can provide significant throughput gains while maintaining fairness levels. However, the channel state information (CSI) estimation techniques in these works would not apply in mmWave scenarios, due to mmWave's high susceptibility to blockage compared to sub-mmWave bands.

In recent years, several works have explored the use of blockage prediction to cope specifically with mmWave's highly dynamic channel conditions [1, 4, 8, 10, 20, 21]. These works employ different techniques to achieve blockage prediction, and are designed for different network goals. [4] considers the optimization of handover performance in dense cellular networks. Blockage occurrence and duration are predicted using peripherals and geometry, enabling elimination of unnecessary handovers caused by transient blockage as well as early handover to avoid long blockage. In [8], a proactive path selection mechanism is proposed to improve resilience of multihop mmWave networks to blockages. In [20], camera images are used to predict blockages caused by human mobiity. Learning based techniques are also used to predict blockage for mmWave transmission [1, 10, 21]. In [21], the blockage prediction decision is made by a trained classifier. On the other hand, [1, 10] both utilize deep learning. The difference is that [1] uses in-band information to make the prediction, whereas [10] uses out-of-band information.

There are some works that focus particularly on blockage-aware scheduling for mmWave networks [7, 9, 10]. Similar to aforementioned proactive schedulers, [9] also proposes a variation of the proportional fair scheduler. In this work, the scheduler reverses the order of time slot scheduling (going from the furthest future to the nearest future) in order to allocate time slots to users before the predicted blockage happens. This approach successfully boosts the link performance for users who experience the most severe blockage, however it slightly decreases the overall throughput of the system. In [10], a DNN scheduler is proposed, which combines blockage prediction with scheduling and beamforming. This work aims to maximize users' achievable rates but does not take fairness into consideration. Lastly, [7] explores the upper bound for proactive scheduling performance through formulating the scheduling problem as an integer linear programming problem.

3 SYSTEM MODEL AND PROBLEM FORMULATION

We model the indoor wireless LAN environment such that there is a room where a number of obstacles are scattered on the floor. In the room, there are n_u users roaming and one fixed ceiling mounted access point (AP). The AP transmits an infinitely backlogged queue of downlink data to a number of users using the orthogonal frequency-division multiple access (OFDMA) scheme. The total duration of the transmission is T seconds, consisting of time slots of length Δt. The total duration is divided into scheduling sessions each containing n_{ts} time slots. All sub-carriers of a time slot are assigned to the same user, where the aggregate bandwidth is b.

The path loss model is divided into two cases. The first case is the simple line-of-sight (LoS) case, where the user received power is calculated by the standard distance-based path loss equation:

$$p_{rx} = p_{tx} \cdot g_{tx} \cdot g_{rx} \cdot l_0 \cdot d_u^{-v} , \tag{1}$$

where p_{tx} is the transmit power, g_{tx} and g_{rx} are the transmit and receive antenna gain, l_0 is the free space path loss at reference distance of 1 meter, d_u is the AP-UE distance, and v is the attenuation exponent.

The user data rate is then calculated with Shannon's capacity formula:

$$r = b \cdot log_2 \left(1 + \frac{p_{rx}}{p_n}\right) . \tag{2}$$

The second case is the non-line-of-sight (NLoS) case. In this case, the received rate has been shown to be highly variable, ranging from near zero to something close to the LoS rate, depending on the blockage conditions and the reflectivity properties of the surrounding materials [12]. To model this behavior, we assume that the NLoS rate is uniformly distributed between zero and the LoS rate with an extra 10 dB of loss.

We consider a scheduling problem where the scheduler knows the predicted data rate for all users at every time slot of the next scheduling session. For fairness considerations, the algorithm also takes as input an allotment scheme, which dictates how many time slots are to be allocated to each user. These two pieces of information are all our proposed efficient heuristic scheduler needs to produce a schedule. The output schedule is represented by a matrix X of size n_u by n_{ts}. In this matrix, each row represents a user and each column represents a time slot, and a value of 1 means the user is assigned for the time slot and 0 otherwise. The problem constraints are that each time slot (column) must have exactly one user assigned and, for every user i, the sum of the values in row i must equal user i's allotment. Subject to these constraints, the goal is to maximize the aggregate data rate over the scheduling session under consideration.

In previous work, we formulated this scheduling problem as a binary integer linear programming problem (BILP). Solving this BILP problem results in an optimal proactive schedule under the given constraints. However, since BILP is an NP-hard problem, it is computationally very expensive to solve. In fact, the scheduling time with a BILP solution substantially exceeds the scheduling session duration, making it only a theoretical solution and not practical for use in real networks. However, in Section 5, we use the optimal result generated by the BILP scheduler as an upper bound to compare the performance of our proposed heuristic scheduler.

4 PROACTIVE SCHEDULING APPROACH

As discussed earlier, previous work showed that proactive schedulers, which make use of link quality prediction, have the potential to bring significant performance improvements. However, prior work has only evaluated an optimal proactive scheduler, which is computationally expensive and, therefore, not suitable for real-time scheduling in practical network environments. In this section, we

introduce a practical proactive scheduling approach which is capable of performing scheduling in real time. Later, evaluations will show that the performance of this efficient scheduler is much better than the classic proportional fair (non-proactive) scheduler and is not far from the performance of the optimal proactive scheduler.

4.1 Overall Process

The proactive scheduling process is assumed to take place in a room with known geometry, known obstacle sizes and locations, users moving around in an unknown fashion, and a mmWave access point at a known location. A schedule is to be determined for the next scheduling session of a certain length broken into a number of time slots of fixed duration. Given the current location of each node along with its current direction and speed, a mobility predictor predicts the path of the node during the next scheduling session.

The next step is to use the predicted path of a node to generate a predicted data rate for the node at each time slot of the next scheduling session. One approach to this is using an RF map [2, 12, 16], which is a heat map of SNR across a region of interest. If an RF map is available for the given space, the predicted data rate is a straightforward mapping of location to SNR to data rate. If no RF map is available, a predicted data rate can be calculated for a given location by determining the line-of-sight status between the location and the access point, applying the appropriate path loss model (either line-of-sight or non-line-of-sight), and performing an SNR to data rate conversion. The predicted data rate vectors for each user, denoted by $rate_u$ for user u, contain the user's predicted data rate at each time slot and are primary inputs to the proactive scheduler.

The other input to the scheduler is the number of slots that each user should be assigned in the next scheduling session. These allocations can be determined in many ways. An obvious one, corresponding to the notion of time fairness, is to assign each user the same number of time slots. For the simulations presented in Section 5, we assume the time slot allocation resulting from the proportional fair (PF) scheduler in order to compare the performance of proactive schedulers to the classic PF scheduler under the same fairness conditions.

The only basic assumption of our scheduling approach is that there is a predicted data rate for every location within the region of movement of the users. In the simulation results presented later, we generate these values based on the geometry of the room, including sizes and locations of obstacles, and path loss models for LoS and NLoS conditions. However, as mentioned earlier, this information can also come from an RF map of the space, which could be learned from measurements taken as users move around. With such a measurement-based RP map approach, *there is no need to know the room geometry and obstacle information to generate the necessary predicted data rates.* Note that the naive mobility predictor used in the simulation results does *not* use obstacle information for its prediction. While obstacle information could be used to improve mobility prediction and the resulting scheduling performance, the results show that even simple mobility prediction without obstacle information performs significantly better than classic proportional fair scheduling.

4.2 Heuristic Scheduler

With future rate knowledge at hand, the optimal scheduling solution requires solving a binary integer programming problem (BILP), which is both computationally demanding and time consuming. The time required to solve the BILP will exceed the scheduling interval in any practical network context and achieving the optimal schedule is, therefore, unrealistic in real time. In order to reach near-optimal data rate performance within realistic computation time, we propose an efficient greedy heuristic algorithm that can schedule time slots to users in a fair allotment scheme. Greedy schedulers have been shown to perform remarkably well in many wireless contexts [5, 6, 17]. The main issue to consider with a greedy scheduler is in what order to consider the time slots and the users within each time slot, which is discussed next.

This algorithm utilizes knowledge of future low data rate occurrences to prioritize the scheduling of scarce resources, thus securing good channel conditions for users that are the most likely to be affected by adverse channel conditions. This prioritization is achieved through ranking of the time slots by how many low-rate users they contain and ranking of users by a combination of how many low-rate slots they have and the number of time slots they have been allocated. Intuitively, the user with the highest number of low-rate slots plus allocated slots is the most difficult to schedule. Although we do not present the results herein, we tried several other orderings of time slots and users and this method performed the best.

The detailed scheduling algorithm is shown in Algorithm 1. The future adverse transmission condition information (such as blockage) is represented by a matrix $status_u$ of size n_u by n_{ts}. Each row of the matrix represents a user and each column represents a time slot. The value of 1 means the respective user is predicted to experience a low data rate for the particular time slot and 0 otherwise. The predicted data rate is likewise represented by $rate_u$ of size n_u by n_{ts}. Lastly, the fair allotment scheme is represented by matrix $allot_u$, where the n_u elements each represent the designated number of total time slots each user is to be scheduled in for the coming scheduling interval. We refer to this as the allotment scheme.

Algorithm 1 can be divided into two steps. In the first step, the algorithm sums the number of low rate slots for each user and for each time slot, in preparation of the ranking (Lines 1-2). The result is two vectors low_rate_u of length n_u and low_rate_{ts} of length n_{ts}. Then, the time slots are ranked from most to least low-rate slots (Line 3) so that time slots with the most adverse channel condition are prioritized. In the second step, the algorithm sequentially walks through all time slots in the scheduling period according to the ranking in an outer for loop, updating the user low-rate slots along the way (Lines 4-27). The *assigned* flag keeps track of whether the current time slot has been assigned or not and is initially set to 0 at the beginning of each iteration (Line 5). Also, for each iteration, the $metric$ is calculated by summing low_rate_u and $allot_u$, then the users are ranked by $metric$, from most to least, to prioritize users that are predicted to experience the worst channel conditions and that need to be assigned the most time slots (Lines 6-7). Once a slot is assigned, the allocation for the assigned user is reduced by one and every user that has a low rate in the assigned slot has their low rate slot count reduced by one (Lines 11-12 and 22-23).

Algorithm 1 Heuristic scheduler

Input: $rate_u$ (predicted user data rates at every time slot), $status_u$ (predicted user low-rate status at each timeslot, 1 = $rate_u \leq$ threshold, 0 = $rate_u >$ threshold), $allot_u$ (number of time slots to be assigned to each user)

Output: X (user assignment)

1: $low\_rate_u[user] = \sum_{ts=1}^{n_{ts}} status_u[user][ts]$, $1 \leq user \leq n_u$
2: $low\_rate_{ts}[ts] = \sum_{user=1}^{n_u} status_u[user][ts]$, $1 \leq ts \leq n_{ts}$
3: $rank_{ts} = low\_rate_{ts}$ sorted from largest to smallest
4: **for each** $ts \in rank_{ts}$ **do**
5: $assigned = 0$
6: $metric_u = low\_rate_u + allot_u, \forall u$
7: $rank_u$ = user IDs sorted from largest to smallest value of $metric_u, \forall u$ with $allot_u > 0$
8: **for each** $user \in rank_u$ **do**
9: **if** $status_u[user][ts] == 0$ **and** $allot_u[user] > 0$ **then**
10: $X[user][ts] = 1$
11: $allot_u[user] = allot_u[user] - 1$
12: $low\_rate_u[v] = low\_rate_u[v] -$
13: $status_u[v][ts],\ 1 \leq v \leq n_u$
14: $assigned = 1$
15: **break**
16: **end if**
17: **end for**
18: **if** $assigned == 0$ **then**
19: **for each** $user \in rank_u$ **do**
20: **if** $allot_u[user] > 0$ **then**
21: $X[user][ts] = 1$
22: $allot_u[user] = allot_u[user] - 1$
23: $low\_rate_u[v] = low\_rate_u[v] -$
24: $status_u[v][ts],\ 1 \leq v \leq n_u$
25: **break**
26: **end if**
27: **end for**
28: **end if**
29: **end for**
30: **return** X

Within the outer loop there are two inner for loops. The first inner loop walks through users according to most to least *metric* and assigns the current time slot to the first user which has both high data-rate and available allotment (Lines 8-16). Once a user is assigned, the available allotment for this user $allot_u[user]$ is decremented. Each element of the low_rate_u vector is then updated according to the $status_u$ of each user in this ts, so that the value reflects the amount of low rate slots for each user in the remaining unassigned time slots (Line 12). The *assigned* flag is set to 1. In the case where the first inner for loop does not assign the current time slot due to no user of both high data rate and available time slot allotment, *assigned* is not set to 1. In this case, a second for loop goes through the users again to assign this time slot to the highest ranked user that still has available allotment (Lines 17-26).

The time complexity of this heuristic algorithm is bounded by $O(n_{ts} * n_u * log(n_u))$, where the n_{ts} accounts for the outer loop and $n_u * log(n_u))$ accounts for the sorting which happens for every iteration of the outer loop. Therefore, the overall complexity of the scheduling algorithm is polynomial time for any problem size, which is efficient as well as scalable.

5 PERFORMANCE STUDY

This simulation study is set in rooms of size 20 meters by 20 meters where obstacles are randomly placed. The users' mobility is modeled with a hot spot mobility model with a number of randomly located hot spots where users pause for a certain duration of time, and then move to a different hot spot location. Users take the shortest path towards the next hot spot, except that they skirt around the edges of any obstacles encountered along the way.

For data rate prediction, we use a naive mobility predictor, which only predicts the user to continue at the current speed in the current direction. The predicted data rate is then calculated using the predicted location, the known room and obstacle geometries, and standard line-of-sight and non-light-of-sight path loss models. Although this results in perfect data rate prediction if the exact user location is known, the mobility prediction produces location errors that result in errors in data rate prediction.

Due to its very basic nature, the naive mobility predictor can be considered to yield worst case results for proactive scheduling. As a reference, we also run all the experiments with a perfect predictor to show the best case performance for our heuristic proactive scheduler. If state-of-the-art mobility prediction techniques are employed, results will fall somewhere in between these two values.

In this section we mainly compare the performance of 3 schedulers: the traditional proportional fair (PF) scheduler as a baseline, our proposed scheduler (referred to as EEP scheduler in the following sections), and the BILP scheduler from [7] which represents the optimal proactive scheduling performance. To provide an equitable comparison, all experiments conducted hereafter use the same allotment scheme, which is the one produced by the PF scheduler.

5.1 Simulation Settings

In this section, we study the influence of several variables on the performance of the algorithm, including: scheduling session length, user pause time, obstacle density and hot spot density, where for each of these factors we picked 3 values representing low, medium and high values. The full set of values used in the experiments are listed in Table 1. The default setting for experiments uses the medium values for all parameters, which is the base line case discussed in Section 5.2. The fixed variables are given by Table 2.

Table 1: Variable Simulation parameters

Parameter	Low	Medium	High
Scheduling session length (s)	1	3	5
User pause time (s)	2	4	6
Number of obstacles	50	65	80
Number of hot spots	4	6	8

All experiments performed in the following subsections simulate a 120 second period in which 20 users are concurrently moving in the same 20m by 20m by 3m room. The ceiling mounted access point is located at the center of the room. The obstacles

Table 2: Fixed Simulation parameters

Bandwidth	b	2GHz
Noise power	p_n	-71.99dBm
Scheduling time slot length	Δt	62.5μs
Transmit power	p_{tx}	20dBm
Transmitter gain	g_{tx}	3.16dBi
Receiver gain	g_{rx}	0dBi
Path loss reference	l_0	63.4dB
Attenuation exponent	v	1.72

in the room are randomly allocated following a Poisson distribution. The obstacles are assumed to be cuboids sitting on the floor and their dimensions are given by the following distributions: $W \sim \mathcal{N}(0.56, 0.08, 0.25, 1.25)$, $L \sim \mathcal{N}(1.08, 0.18, 0.5, 1.75)$, and $H \sim \mathcal{N}(1.85, 0.2, 1.25, 2.4)$. The obstacles are randomly aligned to be parallel to one of the room walls. Finally, when a user arrives at a hot spot, they choose a new hot spot equiprobably at random from all other hot spots to be the next location.

The mobility model was implemented in the ns-3 enhanced mmWave LAN simulator [11]. This simulator includes obstacle modeling and code to determine the LoS/NLoS status of a user device. This information was then fed to an off-line module that calculated predicted data rates according to the LoS/NLoS models presented in Section 3.

5.2 Baseline Average User Rate and Fairness Comparison

We first analyze a baseline case performance, which is the case used for comparison in later subsections that vary one parameter at a time. In this baseline case, the parameter values correspond to the middle column of Table 1.

Let $rate_{PF}$, $rate_{BILP}$, and $rate_{EEP}$ represent the average data rates of the PF, BILP, and EEP schedulers, respectively. We define Q to be the increase over PF as in the following equation:

$$Q = \frac{rate_{BILP} - rate_{PF}}{rate_{PF}} \quad \text{or} \quad \frac{rate_{EEP} - rate_{PF}}{rate_{PF}} \tag{3}$$

Thus, Q takes $rate_{PF}$ as the baseline non-proactive scheduling performance and calculates the improvement of proactive scheduling, via either BILP or EEP. Also, we use ΔR to denote the data rate difference between one of the proactive schedulers and the PF scheduler, i.e. the numerator of Q, and P to denote the ratio between ΔR_{EEP} and ΔR_{BILP}. Thus, P represents how much of the gap between the non-proactive baseline (PF) and the proactive upper bound (BILP), the heuristic proactive scheduler EEP has closed. In the following results, we present the Q and P ratios as percentages.

In Table 3 we show how the proactive schedulers compare to PF scheduler in terms of average data rate performance. The rows labeled "accurate "show the performance of the proactive schedulers with perfect mobility prediction and the rows labeled "predicted" are the results with the naive mobility predictor. We first note that the heuristic EEP scheduler does fairly well compared to the optimal but impractical BILP scheduler, closing about 72% or 62% of the gap between the non-proactive and optimal proactive scheduling results. We also see that the improvement of proactive scheduling over non-proactive is significant, from 15% to 21% higher average rate with perfect mobility prediction. Improvement with the naive mobility predictor is somewhat smaller, being from 6% to 10% higher than non-proactive. This demonstrates that the quality of mobility prediction is critical to realizing the full benefits of proactive scheduling.

Table 3: Baseline case rate percentage improvement against PF

Scheduler	ΔR	Q	P
EEP (accurate)	73.35 Mbps	15.45%	72.36%
BILP (accurate)	101.4 Mbps	21.35%	N/A
EEP (predicted)	29.73 Mbps	6.26%	62.38%
BILP (predicted)	47.65 Mbps	10.04%	N/A

We note that the improved performance of the proactive schedulers is achieved without any loss of fairness, because we use the user time slot allotment produced the PF scheduler as an input to the proactive schedulers. To demonstrate this, We also calculated the proportional fairness by calculating the sum of log rates values for all three schedulers. The results are presented in Table 4, and show that all three schedulers have nearly identical values of the proportional fairness metric.

Table 4: Sum of log rates for PF, EEP, and BILP schedulers

Scheduler	Log Fairness (accu)	Log Fairness (pred)
PF	173.467	N/A
EEP	174.736	173.997
BILP	175.178	174.304

5.3 Impact of Parameters on Mobility Prediction Accuracy

As we have seen that the quality of mobility prediction has a significant impact on proactive scheduling performance, we now evaluate the accuracy of the naive mobility prediction algorithm. Accuracy is defined as the number of time slots where the user location is correctly predicted divided by the total number of time slots. Although this result is not directly part of the scheduler performance, it is still an important factor that affects the performance of the scheduler like other parameters. Understanding its performance can help explain how this and other factors affect the overall system performance.

Fig. 1 depicts the impact of the pause time, scheduling session duration, number of hot spots, and number of obstacles on mobility prediction accuracy. The trends in the results agree with intuition. First, the longer the pause time, the better the accuracy, since the prediction is more likely to be correct when the user is not moving. Next, the scheduling session duration affects the prediction accuracy in an adverse fashion, with a slope that is steeper than the other variables. This is a result of it being harder to correctly predict location the longer into the future we attempt to predict. The number of obstacles also adversely affects prediction accuracy because more obstacles result in more path obstruction and thus more path

diversions, which the naive mobility predictor does not attempt to predict. Lastly, the number of hot spots does not significantly affect prediction accuracy.

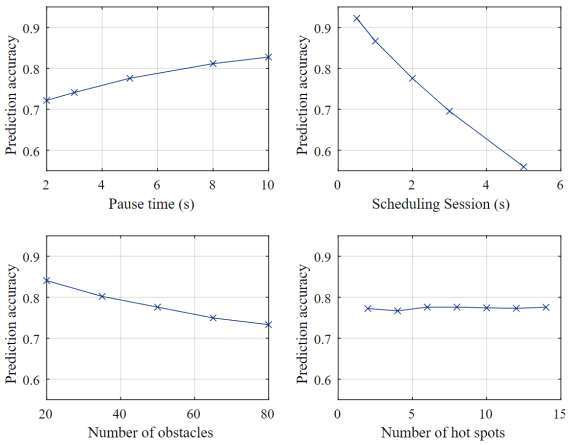

Figure 1: Prediction accuracy vs. different parameters.

5.4 Impact of Pause Time

In this subsection, we study how different pause time durations affect the scheduler performance. Figure 2 plots the average user data rate vs. pause time. As the pause time increases, we see that the performance of the proactive schedulers drop somewhat, despite the mobility prediction accuracy getting higher (see Figure 1). This is due to the pause time becoming longer than the scheduling session duration, which means that if a user is paused at a hot spot with low data rate, the scheduler cannot assign them to higher performing slots within the scheduling session. Despite this, the proactive schedulers significantly outperfrom the non-proactive scheduler in all cases, particularly when the mobility prediction is perfectly accurate.

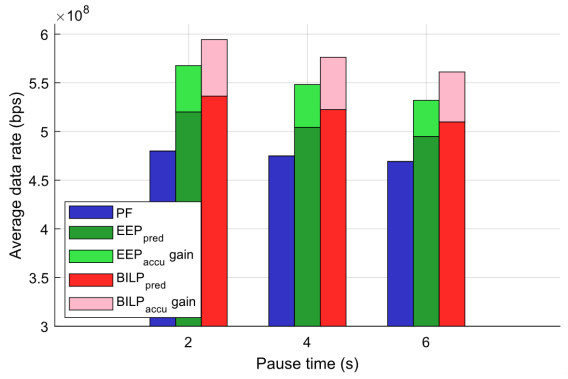

Figure 2: Average user data rate vs. pause time.

Table 5 shows more detailed data on scheduling performance. We again see that the heuristic scheduler does quite well compared to the optimal proactive scheduler, recovering between 62% and 77% of the benefit of the optimal scheduler. The data also confirm the results of Fig. 2 in that the benefits of proactive scheduling are reduced for longer pause times. Finally, the importance of accurate mobility prediction is clearly shown – proactive performance improvement ranges from 13% to 24% with perfect prediction while the improvement is only between 5% and 12% with naive prediction.

Table 5: Proactive scheduler improvement against PF by pause time

Pause (s)	ΔR_{BILP}	Q_{BILP}	ΔR_{EEP}	Q_{EEP}	P
2 (accu)	114 Mbps	23.9%	87.7 Mbps	18.27%	76.63%
4 (accu)	101 Mbps	21.4%	73.3 Mbps	15.45%	72.36%
6 (accu)	91.9 Mbps	19.6%	62.7 Mbps	13.38%	68.28%
2 (naive)	56.3 Mbps	11.8%	39.5 Mbps	8.24%	70.16%
4 (naive)	47.6 Mbps	10.0%	29.7 Mbps	6.26%	62.38%
6 (naive)	40.6 Mbps	8.66%	25.5 Mbps	5.44%	62.84%

5.5 Impact of Scheduling Session Duration

Figure 3 shows the average user data rate plotted against different scheduling session lengths. We can see from the plot that with accurate mobility prediction, the performance of both proactive schedulers significantly increases with longer scheduling sessions. This agrees with the results of [7], which showed that proactive scheduling could have potentially very large gains if accurate predictions could be made over time intervals of several seconds or longer.

However, note that, with the naive predictor, the proactive gain is much smaller. This is explained by Figure 1, where the slope for the scheduling session subplot is significantly steeper than the other parameters, meaning that longer scheduling sessions most heavily degrade prediction accuracy with such a simple predictor. This highlights that, for proactive scheduling to reach its full potential, it will be necessary to combine it with state-of-the-art mobility prediction schemes.

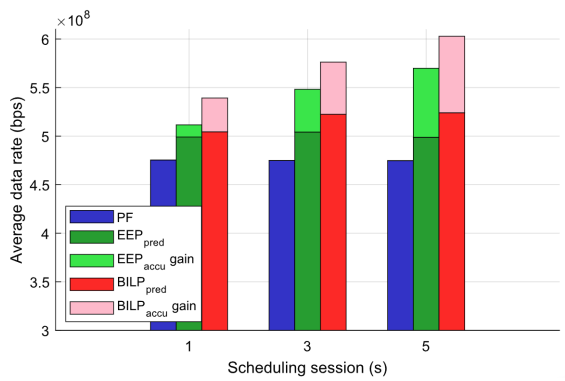

Figure 3: Average user data rate vs. scheduling session duration.

Table 6 shows the detailed data for different scheduling session lengths. With accurate prediction and a 5 second scheduling session,

the heuristic EEP scheduler achieves 20% higher data rate than the PF scheduler and is within 74% of the optimal proactive scheduling gain. However, with naive prediction, its performance drops to only 5% better than PF and 49% away from optimal.

Table 6: Proactive scheduler improvement against PF by scheduling session duration

Sched (s)	ΔR_{BILP}	Q_{BILP}	ΔR_{EEP}	Q_{EEP}	P
1 (accu)	63.9 Mbps	13.5%	36.3 Mbps	7.64%	56.81%
3 (accu)	101 Mbps	21.4%	73.4 Mbps	15.45%	72.36%
5 (accu)	128 Mbps	27.0%	95.1 Mbps	20.02%	74.20%
1 (naive)	29.1 Mbps	6.12%	23.9 Mbps	5.03%	82.20%
3 (naive)	47.7 Mbps	10.0%	29.7 Mbps	6.26%	62.38%
5 (naive)	49.3 Mbps	10.4%	24.2 Mbps	5.09%	49.03%

5.6 Impact of Obstacle and Hot Spot Density

Figure 4 shows the average user data rate vs. the number of obstacles in the room. It can be observed that the benefit of proactive scheduling becomes greater as obstacle density increases. This is due to the users being more likely to experience blockages when more obstacles are present. While the proactive schedulers are almost able to maintain their performance with increasing obstacle density and accurate prediction, the performance of the non-proactive PF scheduler decreases fairly significantly. However, the benefits do decrease when the naive mobility predictor is used. With more obstacles, users tend to detour more often around them, which is not accounted for in the naive prediction scheme. While we could have significantly improved the naive predictor by predicting these detours, this would have made the predictor quite close to the actual mobility model. We chose not to do this, because we wanted to evaluate the range of proactive scheduling performance from a very basic (and not too accurate) prediction scheme to a perfect one.

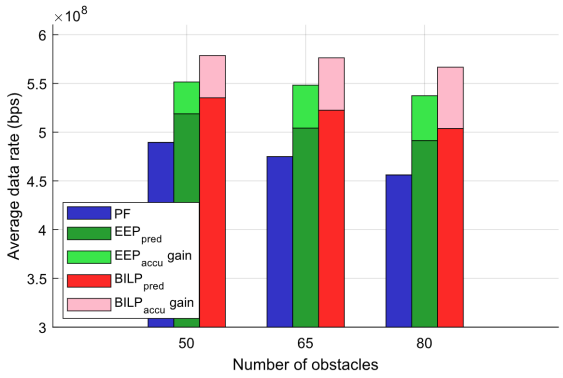

Figure 4: Average user data rate vs. number of obstacles.

From column P in Table 7 we see, once again, that the heuristic EEP scheduler does quite well compared to the optimal BILP scheduler, closing between 62% and 74% of the performance gap between PF and the optimal proactive result.

Table 7: Proactive scheduler improvement against PF by obstacle number

Obs #	ΔR_{BILP}	Q_{BILP}	ΔR_{EEP}	Q_{EEP}	P
50 (accu)	89.1 Mbps	18.2%	62.0 Mbps	12.7%	69.52%
65 (accu)	101 Mbps	21.4%	73.4 Mbps	15.5%	72.36%
80 (accu)	111 Mbps	24.3%	81.5 Mbps	17.9%	73.67%
50 (naive)	45.7 Mbps	9.4%	29.4 Mbps	6.0%	64.23%
65 (naive)	47.6 Mbps	10.0%	29.7 Mbps	6.3%	62.38%
80 (naive)	47.7 Mbps	10.5%	35.2 Mbps	7.7%	73.90%

As the number of hot spots increases, there is a more even distribution of user locations in the room, which helps with users getting better performance on average across all three schedulers. With more hot spots, the average distance between hot spots also decreases, thereby shortening the average user movement time. Therefore, as shown in Figure 5, the performance of all three schedulers increases with the number of hot spots. However, the increase between 6 and 8 hot spots is smaller than between 4 and 6 hot spots, indicating that the performance is likely to converge to a peak value instead of continuing to increase with the number of hot spots.

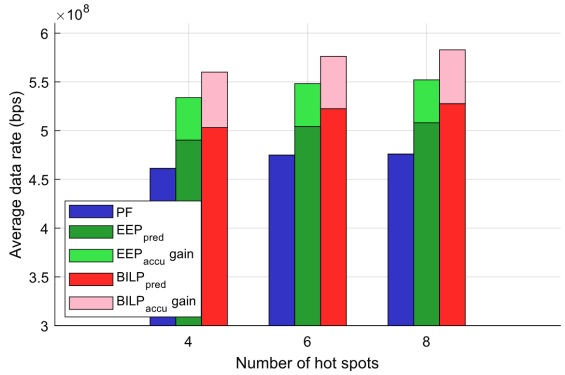

Figure 5: Average user data rate vs. number of hot spots.

As far as EEP is concerned, as the hot spots get denser, EEP's surplus compared to PF also slightly grows, as shown in column ΔR_{EEP} of Table 8. With perfect prediction, EEP can achieve more than 70% of the optimal proactive scheduling gain, whereas with naive prediction, the gain drops to 60% of the optimal.

Table 8: Proactive scheduler improvement against PF by hot spot number

Obs #	ΔR_{BILP}	Q_{BILP}	ΔR_{EEP}	Q_{EEP}	P
4 (accu)	98.8 Mbps	21.41%	72.5 Mbps	15.73%	73.44%
6 (accu)	101 Mbps	21.35%	73.4 Mbps	15.45%	72.36%
8 (accu)	107 Mbps	22.47%	76.3 Mbps	16.03%	71.38%
4 (naive)	41.8 Mbps	9.07%	29.6 Mbps	6.42%	70.82%
6 (naive)	47.6 Mbps	10.04%	29.7 Mbps	6.26%	62.38%
8 (naive)	51.7 Mbps	10.87%	32.4 Mbps	6.81%	62.66%

5.7 Running Time Analysis and Discussion of Alternative Methods

Linear programming (LP) approximation is a common method to obtain near-optimal ILP results. Therefore, for comparison, we tried an alternative method for solving the scheduling problem, which is to apply the same constraints for the BILP formulation into the CPLEX LP solver, and then round the non-integer variables into binary values. How we performed the rounding is by taking the LP result matrix X, walking through it time-slot by time-slot, and selecting the user with the largest non-integer value across that time-slot to be scheduled in that slot. During this process, we also kept track of each user's allotment, and dropped users from the selection process when they reached their allotment.

In Table 9, we show the average running time for the 4 schedulers running one scheduling session of the baseline case in Section 5.2. The running times were evaluated on a machine with 8 cores, 1.8 Ghz clock frequency, and 64 GB of memory. Both the BILP and LP schedulers take several orders of magnitude longer time than the PF and EEP schedulers for a 3 second scheduling session. The reason the LP relaxation method does not significantly speed up compared to the ILP version is that the problem size blows up with the number of time slots to be scheduled and this is the dominant factor in the running time rather than the algorithm complexity. Even though the heuristic EEP scheduler is slower than the PF scheduler, it is clearly fast enough for real-time scheduling purposes even with 48,000 time slots to be scheduled over the 3 second scheduling interval.

Table 9: Average run time for PF, EEP, BILP and LP schedulers

PF	EEP	BILP	LP
0.00556s	0.02126s	7.8745s	6.6091s

6 CONCLUSIONS

In this paper, we presented an efficient heuristic proactive scheduler for mmWave LANs and we compared its performance against a classic non-proactive scheduler and an optimal but impractical proactive scheduler. Two results stood out from these evaluations. First, the heuristic scheduler was able to capture from 60% to 75% of the potential performance gain of proactive scheduling despite a running time that is several orders of magnitude faster than the optimal proactive scheduler. Second, the performance gains of the proactive schedulers were highly dependent on the accuracy of mobility prediction. While performance gains were still achieved with proactive scheduling and a very simple naive mobility predictor, much larger gains are possible if more accurate mobility prediction can be achieved. In future work, we plan to evaluate proactive scheduling based on actual user mobility traces rather than a synthetic mobility model. This will allow us to test out different state-of-the-art mobility prediction schemes to see how close proactive scheduling can come to its theoretical upper bound.

ACKNOWLEDGMENTS

This research was supported by the National Science Foundation under Grants CNS–1813242 and CCRI-2016381.

REFERENCES

[1] Muhammad Alrabeiah and Ahmed Alkhateeb. 2020. Deep Learning for mmWave Beam and Blockage Prediction Using Sub-6 GHz Channels. *IEEE Transactions on Communications* 68, 9 (2020), 5504–5518. https://doi.org/10.1109/TCOMM.2020.3003670

[2] Roohollah Amiri, Srinivas Yerramalli, Taesang Yoo, Mohammed Hirzallah, Marwen Zorgui, Rajat Prakash, and Xiaoxia Zhang. 2023. Indoor Environment Learning via RF-Mapping. *IEEE Journal on Selected Areas of Communication* 41 (Jun. 2023), 1859–1872.

[3] Hans Jorgen Bang, Torbjorn Ekman, and David Gesbert. 2008. Channel predictive proportional fair scheduling. *IEEE Transactions on Wireless Communications* 7, 2 (2008), 482–487. https://doi.org/10.1109/TWC.2008.060729

[4] J. Bao, T. Shu, and H. Li. 2018. Handover Prediction Based on Geometry Method in mmWave Communications: A Sensing Approach. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. 1–6.

[5] Gurashish Brar, Douglas M. Blough, and Paolo Santi. 2006. Computationally Efficient Scheduling with the Physical Interference Model for Throughput Improvement in Wireless Mesh Networks. In *Proc. ACM International Conference on Mobile Computing and Networking*. 2–13.

[6] Reena Chackochan, Senthilkumar Dhanasekaran, and Albert Sunny. 2018. Asynchronous Distributed Greedy Link Scheduling in Multihop Wireless Networks. *IEEE Transactions on Vehicular Technology* 67 (Oct. 2018), 10166–10170.

[7] Ang Deng, Yuchen Liu, and Douglas M. Blough. 2022. Exploring Performance Limits on Proactive Fair Scheduling for mmWave WLANs. In *Proc. IEEE International Symposium on Local and Metropolitan Area Networks*. 1–6. https://doi.org/10.1109/LANMAN54755.2022.9820446

[8] Mine Gokce Dogan, Martina Cardone, and Christina Fragouli. 2022. Proactive Resilient Transmission and Scheduling Mechanisms for mmWave Networks. arXiv:2211.09307 [cs.IT]

[9] Fadhil Firyaguna, Andrea Bonfante, Jacek Kibilda, and Nicola Marchetti. 2020. Performance Evaluation of Scheduling in 5G-mmWave Networks under Human Blockage. *CoRR* abs/2007.13112 (2020). arXiv:2007.13112 https://arxiv.org/abs/2007.13112

[10] Fabian Göttsch and Megumi Kaneko. 2020. Deep Learning-based Beamforming and Blockage Prediction for Sub-6-GHz/mmWave Mobile Networks. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. 1–6. https://doi.org/10.1109/GLOBECOM42002.2020.9322404

[11] Yuchen Liu. 2022. Enhanced mmWave LAN ns-3 simulator. https://github.com/yuchen-sh/mmWave-WLAN-802.11ad/tree/master

[12] Yuchen Liu and Douglas M. Blough. 2022. Environment-Aware Link Quality Prediction for Millimeter-Wave Wireless LANs. In *Proc. ACM International Symposium on Mobility Management and Wireless Access*. 1–10.

[13] George R. MacCartney, Sijia Deng, Shu Sun, and Theodore S. Rappaport. 2016. Millimeter-Wave Human Blockage at 73 GHz with a Simple Double Knife-Edge Diffraction Model and Extension for Directional Antennas. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. 1–6. https://doi.org/10.1109/VTCFall.2016.7881087

[14] George R. MacCartney, Theodore S. Rappaport, and Sundeep Rangan. 2017. Rapid Fading Due to Human Blockage in Pedestrian Crowds at 5G Millimeter-Wave Frequencies. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 1–7. https://doi.org/10.1109/GLOCOM.2017.8254900

[15] Robert Margolies, Ashwin Sridharan, Vaneet Aggarwal, Rittwik Jana, N. K. Shankaranarayanan, Vinay A. Vaishampayan, and Gil Zussman. 2016. Exploiting Mobility in Proportional Fair Cellular Scheduling: Measurements and Algorithms. *IEEE/ACM Transactions on Networking* 24, 1 (2016), 355–367. https://doi.org/10.1109/TNET.2014.2362928

[16] Koya Sato, Katsuya Suto, Kei Inage, Koichi Adachi, and Takeo Fujii. 2021. Space-Frequency-Interpolated Radio Map. *IEEE Transactions on Vehicular Technology* 70 (Jan. 2021), 714–725.

[17] Yulin Shao, Qi Cao, Soung Chang Liew, and He Chen. 2022. Partially Observable Minimum-Age Scheduling: The Greedy Policy. *IEEE Transactions on Communications* 70 (Jan. 2022), 404–418.

[18] Linzhi Shen, Tianyu Wang, and Shaowei Wang. 2019. Proactive Proportional Fair: A Novel Scheduling Algorithm Based on Future Channel Information in OFDMA Systems. In *2019 IEEE/CIC International Conference on Communications in China (ICCC)*. 925–930. https://doi.org/10.1109/ICCChina.2019.8855928

[19] Richard J. Weiler, Michael Peter, Wilhelm Keusgen, and Mike Wisotzki. 2014. Measuring the busy urban 60 GHz outdoor access radio channel. In *2014 IEEE International Conference on Ultra-WideBand (ICUWB)*. 166–170. https://doi.org/10.1109/ICUWB.2014.6958971

[20] K. Yamamoto Y. Oguma, T. Nishio and M. Morikura. 2016. Proactive handover based on human blockage prediction using RGB-D cameras for mmWave communications. *IEEE Transactions on Mobile Computing* E99-B, 6 (Oct. 2016), 1734–1744.

[21] Masoud Zarifneshat, Li Xiao, and Jiliang Tang. 2019. Learning-based Blockage Prediction for Robust Links in Dynamic Millimeter Wave Networks. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SAHCN.2019.8824987