AI-Based Interference Pattern Prediction and Fair Scheduling in Full-Duplex Wireless IoT Networks

Hadi Hosseini and Ehsan Aryafar
Department of Computer Science, Portland State University, Portland, OR, USA

Abstract—**Full-Duplex (FD) wireless is an emerging technology that enables a wireless device to transmit and receive at the same time and on the same frequency band. This can significantly increase the total number of IoT devices that a single access point (AP) can serve. However, in networks with half-duplex (HD) clients, FD operation at the AP can introduce strong interference from uplink (UL) clients to downlink (DL) clients, limiting the system scalability. Efficient scheduling (or pairing) of UL and DL clients can mitigate this interference by selecting clients with low mutual interference (i.e., high path loss). However, acquiring instantaneous channel information between clients to make such scheduling decisions can result in substantial overhead, potentially eliminating all FD capacity gains. This paper introduces a system called PF-FD to address these problems and achieve fairness in multi-AP WiFi-based IoT networks. At its core, PF-FD employs (i) machine learning techniques so that each AP can autonomously predict path loss (or interference relationships) between devices with a high accuracy and low overhead; (ii) A linear complexity scheduling algorithm that helps each AP make scheduling decisions based on its path loss estimates, with outputs that are very close to the proportional fair outcome; and (iii) A distributed contention-based medium access control (MAC) and rate adaptation protocol compatible with legacy WiFi. We evaluate PF-FD with Python and NS3 and show that its path loss estimation method has a very high resiliency to missing data and uncertainties from environmental factors, and that it provides up to 21% more throughput than state-of-the-art.**

I. Introduction

The rapid growth in the number and type of IoT devices has intensified the demand for wireless communication solutions capable of supporting large numbers of connected devices. Full-Duplex (FD) wireless has recently emerged as a solution to address this challenge by enabling simultaneous transmission and reception on the same frequency band, thereby improving spectral efficiency and network capacity [1].

A fundamental challenge in FD wireless is mitigating self-interference (SI) caused by a device's own transmission to its receiver. Over the past decade, many designs have developed mechanisms to suppress or completely eliminate SI [1].

Given the cost and complexity of implementing FD radios, it is expected that access points (APs) will be the first adopters of this technology. Nevertheless, in networks where clients remain half-duplex (HD), FD operation at the AP can introduce significant inter-client interference, particularly from uplink (UL) transmissions disrupting downlink (DL) receptions (Fig. 1), which can degrade the system capacity.

One approach to address this problem is to obtain instantaneous channel state information (CSI) between AP and all its associated clients as well as CSI between all clients, and then use a scheduling method at the AP to appropriately pair UL and DL clients with low mutual interference, and select their optimal transmission rates. However, this introduces several challenges. First, the overhead of obtaining such CSI would be pretty large, which can eliminate all FD capacity gains. Second, the scheduling algorithm should maintains fairness across all clients and have a low computational complexity for real-world implementation. Third, the MAC (medium access control) protocol should support existing devices and be compatible with legacy WiFi.

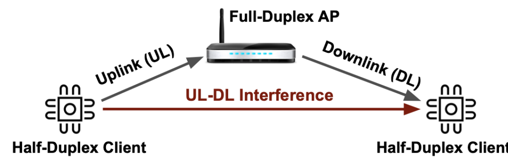

Fig. 1: **AP operates in FD mode. The two IoT clients operate in HD mode.**

To address these challenges, we present the design and evaluation of a distributed system called "PF-FD". At a high level, PF-FD divides time into repetitive cycles, where, in a low overhead manner, coarse-grained path loss[1] information between clients (which also captures mutual interference relationships) is estimated with the help of machine learning (ML) at the beginning of each cycle. This coarse-grained information is then used in a low complexity and approximately proportional fair scheduler at each AP, to make UL-DL pairing (scheduling) decisions during data transmission. The MAC protocol supports legacy wireless devices and is based on 802.11 CSMA/CA (i.e., legacy WiFi) and competition among nodes. Further, the protocol is designed such that instantaneous CSI between an AP and its selected UL-DL pair is obtained. This allows for selection of optimal data rates (i.e., modulation and coding schemes) for both UL and DL data transmissions. Specifically, our key contributions are as follows:

- **Design: Path Loss Estimation:** PF-FD initially uses SVD (Singular Value Decomposition) or KNN (K Nearest Neighbor) algorithms to predict missing AP-client path loss values from a set of sparse AP-client path loss measurements. It then leverages this information and a set of sparse client-client path loss measurements as inputs to a neural network that predicts missing client-client path loss values independently at each AP. **Scheduling:** PF-FD

[1]Path loss refers to the reduction in power density of a wireless signal as it propagates through space from a transmitter to a receiver.

uses a linear complexity UL-DL client pairing algorithm that gets very close to the proportional fair outcome. **MAC:** PF-FD employs a contention based MAC that incorporates the proposed scheduling algorithm at each AP and determines optimal instantaneous UL/DL rates.

- **Evaluation:** We gather two datasets that represent different wireless environments and use them in NS3 for network simulations. In the first dataset, we use commercial wireless devices to collect RSS (Received Signal Strength) and path loss data across multiple topologies, each composed of five APs and ten clients. In the second dataset, we use a standard compatible path loss model with 10 APs and 90 clients to evaluate PF-FD's performance in larger configurations. Our simulation results in Python and NS3 show that **(i) ML Metrics:** SVD outperforms KNN in predicting AP-client path loss values. Further, our neural network has a very high accuracy in predicting missing client-client path loss values, e.g., achieving less than 8% MAPE (Mean Absolute Percentage Error) with 95% of client-client and 80% of AP-client path loss values missing and there is high uncertainty in path loss values due to environmental factors. **(ii) Networking Metrics:** Compared to IC2 (state-of-the-art solution), PF-FD achieves up to 21% higher throughput in saturation region, 22% higher throughput with asymmetric traffic, and 22% higher proportional fairness value.

The paper is organized as follows. We discuss the related work in Section II and the system model in Section III. We present the details of PF-FD in Section IV, our performance evaluation results in Section V, and conclusion in Section VI.

II. Related Work

Our works falls in the category of machine learning for FD wireless IoT networks. Here, we provide a brief summary of related solutions. For a more comprehensive discussion of related work, please refer to [1]–[4].

PHY Solutions: The work in [5] uses a side-channel between clients to cancel UL-DL interference. PF-FD, on the other hand, handles interference by intelligent pairing, and can use any side-channel to increase capacity. Other work [6], has developed a joint power allocation and beamforming scheme for multi-cell FD wireless, which requires channel information between devices creating scalability and overhead issues. IC2 [7] presents a joint PHY-MAC solution for actively canceling UL-DL interference by using power control at the UL client and allocating some of the AP's power to send a canceling signal towards the DL client. The level of canceling signal power depends on the UL-DL client pairing, and IC2 uses a weighted historical-based solution to pair clients. We compare PF-FD against IC2 and show that PF-FD achieves up to 21% higher throughput in saturation region and 22% higher throughput with asymmetric traffic.

MAC Solutions: FD^2 [8] uses directional antennas at APs and clients to reduce UL-DL interference. PF-FD can be employed in FD^2 to reduce the information gathering overhead. The work in [9] proposes a generalized proportional fair scheduling scheme assuming both HD and FD clients. Similarly, theoretical upper bounds on the proportional fair outcome were presented in [10]. PF-FD incorporates ML methods for path loss estimation to reduce overhead and uses those predictions in a linear complexity scheduling algorithm that gets very close to the proportional fair outcome. Client location and RF map of an environment can be used to predict UL-DL interference [11]. However, localization in indoor environments is hard [12], continuous location sharing with APs creates overhead, and RF map changes as objects move.

III. System Model

We assume a multi-cell WiFi network deployed within an infrastructure (e.g., indoor enterprise, smart factory). There are K APs and M IoT clients in the network. All APs are connected to one another through a wired backbone such as Ethernet, which allows them to easily share information.

For ease of explanation[2], we assume that: (i) UL and DL clients are separate. Specifically, there are D DL clients and U uplink clients in the network, and $D + L = M$. (ii) All clients and APs have omni-directional antennas with 0 dBi gain. (iii) Clients and AP are in full buffer mode, i.e., each UL or DL client always has a packet to transmit or receive. (iv) There is no residual self-interference when AP operates in FD mode.

APs are able to operate in both HD and FD mode, while clients are restricted to HD operation. Clients are dispersed across different cells and can only communicate with the AP they are associated with. We refer to the AP and clients associated with it as a BSS (Basic Service Set, which is the terminology adopted in WiFi standards). We assume that different BSSs are tuned to different frequency channels, and hence do not interfere with one another. Fig. 2a illustrates the architecture studied in this paper.

As illustrated in Fig. 2b, we assume that time is divided into two intervals: (i) DGI (Data Gathering Interval), in which all APs and clients share information that will help each AP independently predict the level of path loss between itself and its associated clients as well as the path loss in between its associated clients. (ii) DTI (Data Transmission Interval), during which devices compete for channel access according to the random access MAC protocol discussed in Section IV-E.

During DTI, based on the outcome of the MAC competition, each AP uses our scheduling algorithm (Algorithm 1) to determine *if* and *how* UL and DL clients should be paired to achieve proportional fairness. The scheduling algorithm uses coarse-grained path loss estimates from DGI interval as its input. The MAC protocol is designed in such a ways that lets each AP obtain instantaneous channel (CSI) estimates between the AP and its selected UL-DL pair. This allows the AP to determine the optimal transmission rates for both UL and DL data packets with a low overhead.

[2]Note that the system can be easily modified to remove these assumptions, but we use them here for ease of presentation. For example, the SINR (Signal-to-Interference-plus-Noise Ratio) expressions in Eq. 1 or Eq. 4 can be easily updated to include antenna gains or residual FD self-interference. Similarly, the information collection step in Section IV-B can be updated to include buffer level at each IoT device.

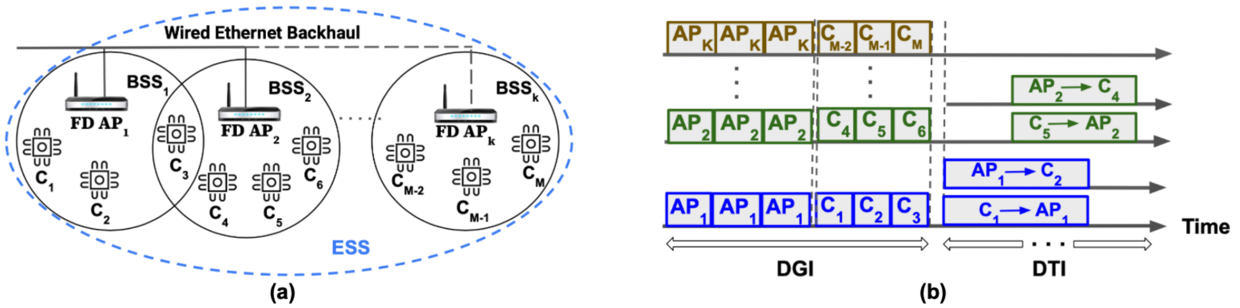

Fig. 2: **(a) Wi-Fi network with K APs and M IoT clients. Clients are associated to APs based on a configuration metric such as load balancing or signal strength. The set of APs connected to the same Ethernet backbone is referred to as ESS (Extended Service Set). (b) PF-FD operation. Time is divided into repetitive cycles. Different colors show different frequency channels. During DGI, clients and APs send messages that will help each AP independently estimate the level of path loss between itself and its associated clients as well as in between its associated clients.**

We refer to the combination of DGI and DTI time intervals as a cycle, which repeats itself. The duration of the cycle as well as the duration of the DGI and DTI time intervals depend on the size of the network and the level of mobility of wireless devices, and are configured by the network operator.

Our approach allows for learning of long-term channel fluctuation relationships at the beginning of each cycle, and then learning and accommodating short-term channel relationships during the actual data transmissions. This eliminates the need to continuously obtain accurate channel (CSI) between all wireless devices in a BSS for every transmission, which incurs a very large overhead and can eliminate all FD capacity gains.

Pairing (scheduling) of clients is based on coarse-grained path loss, SNR (Signal-to-Noise Ratio), or SINR (Signal-to-Interference-plus-Noise Ratio) estimations at the AP. Specifically, we define $SNR_{u,k}$ in dB (decibel) as the SNR at AP k from UL client u, and calculate it as:

$$SNR_{u,k} = {P_u}^{UL} - L_{u,k} - N_k \tag{1}$$

Here, ${P_u}^{UL}$ (in dBm) is the transmission power of UL client u, which we assume to be provided to the AP during client association. Further, $L_{u,k}$ (in dB) is the path loss between UL client u and AP k, and N_k (in dBm) is the AP k noise power. Note that Eq. 1 does not have the antenna gain factors, since we assume all devices have 0 dBi gain antennas in order to simplify our presentation. Further, Eq. 1 holds irrespective of whether AP operates in HD or FD mode, since we assume there is no residual self-interference.

With a known ${P_u}^{UL}$, an AP k can easily derive $L_{u,k}$ from measured received signal strength ($RSS_{u,k}$ in dBm) from client u's UL transmission at AP k. This is because:

$$RSS_{u,k} = {P_u}^{UL} - L_{u,k} = SNR_{u,k} + N_k \tag{2}$$

Note that path loss between a pair of wireless devices is always symmetric, that is $L_{u,k} = L_{k,u}$.

Similarly, if an AP k transmits a message to a client d in HD mode, the client can measure the received signal strength and estimate its path loss from AP k ($L_{k,d}$) with a known AP transmission power level (${P_k}^{DL}$), leveraging:

$$RSS_{k,d} = {P_k}^{DL} - L_{k,d} \tag{3}$$

Now, let's assume that AP k decides to operate in FD mode by pairing UL client u with DL client d. The AP can use a coarse-grained estimate of its DL SINR at client d for such scheduling purposes, leveraging:

$$SINR_{k,d} = {P_k}^{DL} - L_{k,d} - 10\log_{10}(10^{\frac{{P_u}^{UL} - L_{u,d}}{10}} + 10^{\frac{N_d}{10}}) \tag{4}$$

Here, ${P_k}^{DL}$ is the AP transmit power, $L_{k,d}$ is the path loss between AP k and client d, $L_{u,d}$ is the path loss between clients u and d, and N_d is the noise power at client d. The exponentiation terms in Eq. 4 are to correctly add up the impact of interference and noise, as all our parameters are in dBm/dB. Further, the noise power for a communication system with bandwidth BW can be approximated as:

$$-174 + 10log_{10}(BW) \tag{5}$$

Note that scheduling (i.e., pairing of UL and DL clients) at the AP, requires information about UL and DL SNR/SINR values for different combination of UL and DL clients. The AP can easily obtain information about all parameters in Eq. 1 and Eq. 4, except for path loss variables, in particular $L_{u,d}$, which is the path loss between UL client u and DL client d.

A basic approach to solve this problem involves each AP sending a message to its associated clients, which then sequentially report the measured AP-client path loss. During this process, all other clients can measure client-client path loss. After the last client sends its message, clients sequentially report these measured client-client path loss values back to the AP. For M' clients, this approach incurs $O(M'^2)$ transmission bytes (assuming one byte per path loss value). Although this method provides accurate measurements, they remain valid only for a short coherence time, which is negligible compared to a cycle duration. In contrast, PF-FD incurs a significantly lower overhead (only $O(M')$ bytes per AP as we show later in Section IV-B) while providing path loss estimates that maintain comparable accuracy after a coherence time.

IV. PF-FD Design

In this section, we first provide an overview of the PF-FD system. Next, we introduce a novel ML-based technique for AP-client and client-client path loss estimation with low over-the-air transmission overhead. Utilizing this information, we then formulate a problem aimed at achieving proportional fairness. We end the section by discussing our proposed algorithm to solve the pairing of clients as well as a FD MAC, which also determines the optimal data rates for UL and DL packet transmissions based on instantaneous CSI.

A. System Overview

PF-FD is a distributed system tailored for FD infrastructure WiFi networks, aimed at optimizing both DL and UL rates, while approximating proportional fairness. The key components in the design of PF-FD are as follows:

1) **Information Collection (Section IV-B):** During this phase, APs exchange messages with their clients as well as themselves (over the wired backbone).
2) **Path Loss Prediction (Section IV-C):** Each AP uses the information gathered from the previous step and ML-based methods to independently estimate coarse-grained path loss values between itself and its clients as well as between its clients. The network operator can choose the appropriate level of overhead in step 1 to achieve a desired level of accuracy in step 2.
3) **Proportional Fair (PF) Scheduling (Section IV-D):** Each AP uses our algorithm to decide *if* and *how* UL and DL clients should be paired. The pairing is based on coarse-grained path loss estimates.
4) **Data Transmission (Section IV-E):** In each BSS, AP and clients compete for medium access. In our MAC protocol, AP uses the algorithm in step 3 to determine how clients should be paired. Further, the MAC protocol allows for AP to obtain instantaneous CSI between AP and its selected UL-DL pair. This allows the AP to determine optimal UL and DL transmission rates based on instantaneous channel conditions. Finally, each time a successful UL and/or DL transmission happens, the corresponding path loss values are identified. The old (measured or predicted) AP-client and client-client path loss values are then replaced with the up-to-date ones. This allows the scheduling algorithm to use the most recent values in future scheduling (pairing) decisions.

B. Information Collection

During the DGI interval, all APs within the ESS synchronize in time. The following actions are then taken to collect the necessary information.

First, each AP repeatedly sends a message for a number of times (specified by the operator) in its own frequency channel (as depicted in Fig. 2b). Each message contains information about the future order of transmission of associated clients. Each client measures its associated AP's RSS (or path loss, leveraging Eq. 3) and then tunes to one or more randomly selected channels to obtain RSS (or path loss) information of a few other APs. Then, clients tune to their Associated APs' channels, and report their measured RSS (or path loss) values to their APs based on the order specified in each AP's message.

For example, consider the setup specified in Fig. 2b with clients 1-3 associated with AP 1, clients 4-6 associated with AP 2, and clients 7-9 associated with AP 3. Clients 1 and 2 report measured RSS values from AP 1 and other randomly selected APs. Further, during client 1's transmission, clients 2 and 3 measure client 1's RSS. Then, client 2 during its transmission not only sends measured RSS values from APs, but also measured RSS value from client 1. Now, during client 2's transmission, client 3 randomly switches to a different channel and measures RSS from client 5's transmission, which happens at the same time client 2 is sending its message. Then, during its transmission turn, client 3 sends RSS from some APs as well as RSS from clients 1 and 5. At the end of DGI, each AP broadcasts its locally constructed AP-client and client-client matrices over the wired backbone, so that all APs have the same information. APs merge all information to construct sparse AP-client and client-client matrices

Figs. 3a and 3b illustrate the outcomes of this process. Based on the messages exchanged in the DGI interval, each AP is able to build the sparse AP-client matrix in Fig. 3a. Here, each matrix element shows the measured path loss between a client and an AP based on RSS measurements. Further, each AP is also able to build a sparse client-client path loss matrix (Fig. 3b). In Section IV-C, we use ML models to populate these sparse AP-client and client-client matrices.

Overhead Analysis. Each AP broadcasts its messages for a fixed number of times (say x). If M' is the number of clients per AP, the number of messages sent at each BSS will be $x+M'$. Further, the number of client-client samples is constant (and as we show later through simulations can be very low). Thus, total number of bytes sent is $O(M')$.

C. Path Loss Matrix Population

This section outlines the details of constructing full AP-client and client-client path loss matrices at each AP based on a few measured values in each matrix.

Specifically, based on the information exchanged in the previous section, each AP is able to construct a sparse AP-client path loss matrix as depicted in Fig. 3a as well as a sparse client-client path loss matrix as depicted in Fig. 3b. Note that due to the symmetry in path loss values, the matrix in Fig. 3b is symmetric.

At a high level, our approach to populate these matrices is as follows. First, we use ideas from recommender systems to populate the sparse AP-client matrix as depicted in Fig. 3c. Next, we use rows of the populated AP-client matrix, as input features to a fully connected neural network (Fig. 3d) to train and later predict the unknown client-client path loss values and populate the corresponding matrix as depicted in Fig. 3e.

AP-Client Path Loss Matrix Population. We employ and compare two prominent ML techniques that are also used in recommender systems: k-Nearest Neighbors (KNN) and Singular Value Decomposition (SVD).

KNN is a memory-based (or neighborhood-based) collaborative filtering (CF) method, which directly relies on the user-item interaction data (e.g., ratings) to find similar users or items [13], [14]. The initial step involves representing user-item interactions meaningfully, typically accomplished by

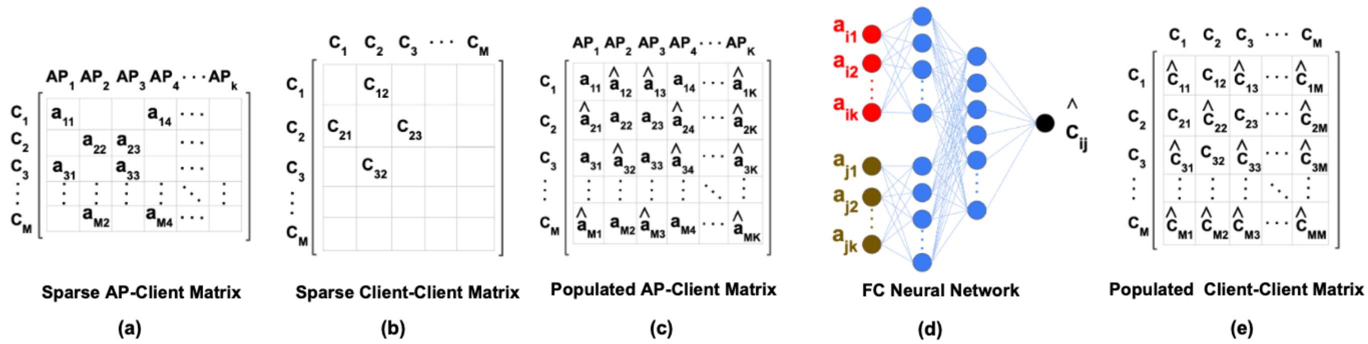

Fig. 3: **(a) Sparse AP-client path loss matrix is generated at each AP based on the information exchanged in DGI, (b) Matrix C: Based on the messages exchanged in DGI, each AP is also able to generate a sparse client-client path loss matrix (which is symmetric), (c) We use KNN or SVD to populate the sparse AP-client matrix. The populated matrix is referred to as matrix A. The predicted values are shown with a ˆ mark, (d) The rows of matrix A serve as input features to a fully connected neural network, enabling the network to learn and predict unknown client-client path loss values, (e) The missing client-client path loss values in matrix C are predicted with the help of the neural network.**

creating a user-item matrix, wherein each row corresponds to a user and each column represents an item. Subsequently, a similarity measure such as the Pearson correlation coefficient quantifies the similarity between users or items. To predict a user's rating or preference for an item, KNN identifies the k most similar users (or items) to the target user (or item) based on the chosen similarity measure. It then aggregates the ratings (or preferences) of these similar users (or items) to make a prediction. Finally, recommendations are generated by predicting ratings for unrated items for a given user and suggesting items with the highest ratings [15]. One formula for KNN prediction of user rating for an unrated item is:

$$\hat{r}_{ui} = \frac{\sum_{v \in S(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in S(u)} \text{sim}(u, v)} \tag{6}$$

Where $\hat{r}_{ui}$ represents the predicted rating of user u for item i, $\text{sim}(u, v)$ denotes the similarity between users u and v, $S(u)$ is the set of K nearest neighbors to user u, and r_{vi} is the rating given by user v to item i.

In our scheme, we use KNN as the first method to populate the AP-client path loss matrix. We do this, by replacing users with clients, items with APs, and path loss as the metric.

Latent factor and matrix factorization models are classified as model-based CF. We use them as a second method to populate the sparse AP-client matrix.

In this method, matrix factorization techniques decompose the user-item interaction matrix into low-rank matrices to uncover latent features that represent user preferences and item characteristics. SVD is a commonly used matrix factorization method. In SVD, we predict $\hat{r}_{ui}$ as follows [16]:

$$\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{q_i}^T \mathbf{p_u} \tag{7}$$

Each item i corresponds to a vector $\mathbf{q_i} \in \mathbb{R}^f$ where the elements of $\mathbf{q_i}$ quantify the degree to which the item embodies latent factors. Similarly, each user u is linked to a vector $\mathbf{p_u} \in \mathbb{R}^f$. The size of $\mathbf{q_i}$ and $\mathbf{p_u}$ is determined by the dimensionality f, which represents the number of latent factors in the model. This dimensionality f is typically much smaller than the dimensions of the user-item matrix and can vary depending on the complexity of the data and the desired accuracy of the model. Within this framework, the components of $\mathbf{p_u}$ gauge the user's level of interest in items that exhibit prominence in the respective factors. The resultant dot product, $\mathbf{q_i}^T \mathbf{p_u}$, captures the interaction between the uth user and the ith item. SVD emerges as a viable method for uncovering the latent vectors of $\mathbf{q_i}$ and $\mathbf{p_u}$. The overall average rating is denoted by μ; the parameters b_u and b_i indicate the observed deviations of user u and item i, respectively, from the average. The learning process is then driven by minimizing the squared error function, manifested as [16]:

$$\min_{p*, q*, b*} \sum_{(u,i) \in S} (r_{ui} - \hat{r}_{ui})^2 + \lambda({b_i}^2 + {b_u}^2 + \|\mathbf{q_i}\|^2 + \|\mathbf{p_u}\|^2)$$

Where, S is the set of the (u, i) pairs for which r_{ui} is known (the training set), λ controls the extent of regularization and is usually determined by cross-validation.

Client-Client Path Loss Matrix Population. Once the AP-client matrix is populated (Fig. 3c), the next step is for each AP to independently populate the client-client path loss matrix. Note that at the end of the DGI interval, the sparse client-client matrix (Fig. 3b) is known at each AP.

Let **A** denote the populated AP-client path loss matrix and **C** denote the sparse client-client matrix. We take the following steps to populate matrix **C**.

We treat each row of the matrix **A** as a feature vector, resulting in M feature vectors each of dimension K. We next build and train a fully connected artificial neural network (ANN), leveraging these input vectors. Fig. 3d depicts our neural network, which aims at predicting the path loss between two given clients i and j. It does so, by taking rows i and j of matrix **A** as input to the ANN, and making a (regression) prediction denoted as $\hat{c}_{ij}$ (predicted client-client path loss).

Prior to making predictions and populating the client-client matrix, we need to train the ANN, which means determining its model parameters or weights. We train the ANN using supervised learning and leveraging the true values for known client-client path loss variables c_{ij}. Specifically, we train the network by leveraging the following loss formula:

$$\min_{\mathbf{W}} \quad \sum_{i,j} L(c_{ij}, F(\mathbf{W}, \text{features of i \& j})) \tag{8}$$

Here, $\mathbf{W}$ represents the ANN weight matrix, L represents the loss function, F is the output (regression prediction) of ANN, and features of i and j represent the ith and jth rows of matrix $\mathbf{A}$. The summation happens over i and j values for which c_{ij} is known.

To enforce the symmetric nature of the predictions in our training (since the matrix in Fig. 3e should be symmetric), we let the initial weights for top nodes in the first hidden layer in Fig. 3d to be the same as the bottom nodes. Further, during back propagation an average of weights is applied to both parts. This makes sure that if we swap the two input vectors, the prediction remains the same.

The ANN that we used in our implementation has two hidden layers of size 64 and 32. We additionally used ReLU as our activation function, Adam as our optimizer, and mean squared error as our loss function.

ANN vs. SVD/KNN. The reason for choosing different techniques for populating matrix **A** and matrix **C** is that matrix **C** is significantly sparser than matrix **A**. This sparsity makes methods like SVD or KNN less effective, as the training data for estimating missing values is insufficiently dense. Further, the ANN can utilize the values in matrix **A** as part of the training dataset, enriching the available data and improving the accuracy of matrix **C** estimation. While the values in matrix **A** might introduce some bias due to their dependence on prior estimations, our ML approach effectively minimizes this bias by leveraging patterns across the available data. This robustness is reflected in the low MAPE values achieved, as we show later in Section V-B, even with up to 95% missing values in matrix **C**.

D. Proportional Fair Scheduling

In this section, we discuss how an AP can schedule (pair) UL and DL clients to achieve PF. Consider a BSS k with AP k and a set of associated UL and DL client.

Let's define nth TXOP (transmission opportunity) as the nth time any wireless device in the BSS (AP or an UL client) has grabbed the medium for transmission. The key question to answer then is *if* and *how* an AP should pair UL and DL clients to achieve proportional fairness. We refer to this problem as proportional fair scheduling or pairing.

Let $SNR_{u,k}$ and $SINR_{k,d}$ be defined as the uplink SNR (from client u to AP k) and downlink SINR (from AP k to client d) as formulated in Eq. 1 and Eq. 4, respectively. Consider the nth TXOP. Then, we define ${R_{d,n}}^{DL}$ as the coarse-grained[3] data rate for DL client d and ${R_{u,n}}^{UL}$ as the coarse-grained data rate for UL client u as follows:

$${R_{d,n}}^{DL} = BW\ log_2(1 + 10^{\frac{SINR_{k,d}}{10}}) \tag{9}$$

[3]We use coarse-grained since these rates are used for making scheduling decisions and would be different from real data rates (based on instantaneous CSI estimates) for actual data transmissions.

$${R_{u,n}}^{UL} = BW\ log_2(1 + 10^{\frac{SNR_{u,k}}{10}}) \tag{10}$$

Note that SNR/SINR values in Eq. 1 and Eq. 4 are in dB, and therefore they appear as part of the exponent in the above equations. Next, we use the notation from an IIR (Infinite Impulse Response) filter to smooth out variations in data rates, and define the average rate for DL client d and the average rate for UL client u until nth TXOP as:

$$\overline{{R_d}^{DL}}(n) = (1 - \frac{1}{T})\overline{{R_d}^{DL}}(n-1) + \frac{1}{T} I_{d,n} {R_{d,n}}^{DL} \tag{11}$$

$$\overline{{R_u}^{UL}}(n) = (1 - \frac{1}{T})\overline{{R_u}^{UL}}(n-1) + \frac{1}{T} I_{u,n} {R_{u,n}}^{UL} \tag{12}$$

Here, $\frac{1}{T}$ represents the coefficient of the IIR filter, with T denoting the time window size for average rate updating. $\overline{{R_d}^{DL}}(n-1)$ and $\overline{{R_u}^{UL}}(n-1)$ represent the average data rate for the DL client d and the UL client u in the previous TXOP, respectively. We define $I_{d,n} \in \{0, 1\}$ as an assignment index indicating whether DL transmission for client d is scheduled in the current TXOP, and similarly $I_{u,n} \in \{0, 1\}$ is defined for UL transmission scheduling of client u. If $I_{d,n} = 1$ and $I_{u,n} = 1$, the AP decides to operate in FD mode in the current TXOP by pairing clients d and u. If $I_{d,n} = 1$ and $I_{u,n} = 0$, the AP decides to operate in HD DL mode, and if $I_{d,n} = 0$, $I_{u,n} = 1$, the AP decides to operate in HD UL mode.

In our client scheduling problem formulation (problem $\mathcal{P}_1$), we aim to maximize the sum of logarithmic functions of the average rates for both DL and UL transmissions to achieve proportional fairness as follows:

$$\mathcal{P}_1 : \quad \underset{d,u}{\text{MAX}} \quad \sum_{d=1}^{D} log(\overline{{R_d}^{DL}}(n)) + \sum_{u=1}^{U} log(\overline{{R_u}^{UL}}(n)) \tag{13}$$

Our goal in problem $\mathcal{P}_1$ is to find the best pair of DL and UL clients denoted by d and u, respectively, in the current TXOP that maximizes the sum of log of average data rates across all DL and UL clients up to the current TXOP. In our setting, at most one pair of UL and DL clients can be paired at each TXOP. Solving the optimization problem involves selecting the optimal client pair (or a single DL client) when the AP acquires the medium first. Alternatively, when an UL client grabs the medium, the problem reduces to selecting the best DL client or refraining from a DL transmission altogether.

We can solve this problem optimally by doing an exhaustive search and choosing the output that has the highest value in problem $\mathcal{P}_1$. Consider a BSS with M' total clients, D' DL clients and U' UL clients ($D' + U' = M'$). Then, selecting the best pair for FD mode when AP grabs the medium first requires $D' \times U'$ searches. To also consider HD mode, an additional D' searches are needed to find the value of objective function in $\mathcal{P}_1$ for each one of DL clients. Therefore, the total number of searches required is $(D' \times U') + D'$, which results in $O(M'^2)$ computational complexity (e.g., assuming $D' = U' = \frac{M'}{2}$), representing a polynomial time complexity of degree two. On the other hand, if an UL client grabs the medium first, AP needs to search through DL clients only, which will result in $O(M')$ (i.e., linear) complexity.

Algorithm 1 illustrates our proposed pair selection (scheduling) algorithm. Instead of the optimal exhaustive search described earlier (which has $O(M'^2)$) complexity, Algorithm 1 has linear overall complexity. Specifically, when AP grabs the medium, AP always goes through DL packets in a round robin manner. As a result, the DL client is known and the search (pairing) happens only over the UL clients. The reason we use this approach is that if a packet transmission fails at the MAC level, AP would continue to serve packets from the same DL client until a successful transmission goes through. This can sometimes lead to excessive delays with bursty interference at the location of DL client. Serving DL clients in round robin manner eliminates these types of delays, mimics the fair-opportunity principle among UL clients[4], and achieves an overall throughput that is very close to the exhaustive search outcome with lower computational complexity.

Algorithm 1 UL and DL Pair Selection

Input: $R_{d,n}{}^{DL}, R_{u,n}{}^{UL}, \overline{R_d{}^{DL}}(n-1), \overline{R_u{}^{UL}}(n-1)$
Output: d, u, (d,u) {Output: Selected UL or DL client indices in HD mode or UL-DL pair indices in FD mode}

```
Switch
Case 1: AP grabs the medium with a packet for DL client d
for u = 1 to U' do
    Calculate R = log(R_d^DL(n)) + log(R_u^UL(n))
    Append R to the end of R_FD list
end for
if max(R_FD) > log(R_d^DL(n)) then
    Find the corresponding index u for the max(R_FD)
    return (d, u)
else
    return d
end if
Case 2: UL client u grabs the medium
for d = 1 to D' do
    Calculate R = log(R_d^DL(n)) + log(R_u^UL(n))
    Append R to the end of R_FD list
end for
if max(R_FD) > log(R_u^UL(n)) then
    Find the corresponding index d for the max(R_FD)
    return (d, u)
else
    return u
end if
```

E. MAC Protocol

The scheduling algorithm presented in the previous section, uses coarse-grained path loss (and SNR/SINR) values for UL-DL client pairings. However, modulation and coding scheme (MCS) selection (also known as rate selection) based on coarse-grained information or trial and error is known to be inferior compared to MCS selection based on up to date channel information. In this section, we present the details of our MAC protocol and also discuss how AP announces the appropriate rate/MCS values. Our MAC protocol is similar to the MAC protocol specified in IC2 [7]. However, our MAC is designed to support competition among AP and all UL clients. Further, we include modifications through fragmentation to support UL-DL rate diversity.

[4]Consider UL traffic only and 802.11 DCF, where clients randomly compete for channel. This results in similar medium access probability (opportunity) across all clients [17].

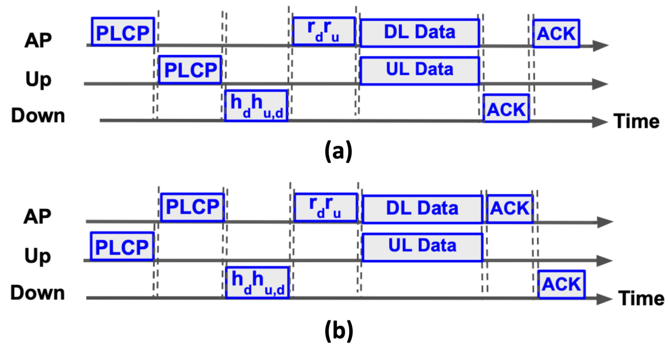

Fig. 4: **(a) AP grabs the medium first, (b) Client wins the competition.**

The key challenge for appropriate rate (MCS) selection is access to instantaneous channel (CSI) information [18]. Thus, we focus on CSI acquisition. For example, an UL client with knowledge of its CSI at the AP (h_u) can choose the optimal MCS for transmission. Similarly, if an AP operates in FD mode, the AP needs knowledge about the CSI between itself and DL client (h_d) as well as channel between UL and DL clients ($h_{u,d}$) to determine its SINR at the DL client and the optimal data rate it should use for transmission.

Our MAC protocol helps with CSI acquisition, does not obstruct the operation of legacy nodes, and is compatible with 802.11. Specifically, similar to 802.11 our MAC is based on competition between all nodes for channel access. If AP grabs the medium first (Fig. 4a) and has decided to operate in FD mode, it would then send a preamble message (PLCP in WiFi), which allows the DL client measure DL CSI (h_d) and also instructs the selected UL client to transmit a PLCP message. The DL client obtains $h_{u,d}$ based on UL client's PLCP transmission and feeds that back along with h_d to the AP. The AP then sends a control message that announces the DL data rate it plans to use as well as the data rate the UL client should use for transmission to AP. AP and UL client then proceed to data transmission. If the UL data packet is smaller than the DL data packet, the UL client ends its transmission earlier. If on the other hand, UL data packet is larger than the DL duration, the UL client would fragment its data and send it over multiple transmission opportunities. The AP only forwards an UL client's packet when all its fragments are received. Either way, after data transmission ends, the DL client sends an ACK packet to acknowledge reception of AP's packet, which is followed by AP ACK transmission to acknowledge UL packet's transmission.

When UL client grabs the medium first (Fig. 4b), we apply a similar method, except that UL client sends the PLCP message first and all other clients measure the channel for possible feedback to AP. The AP then follows with PLCP transmission and identification of specific DL client that should send h_d

as well as $h_{u,d}$. After this, AP announces the rates UL and DL clients should use for transmission. For ACK packets, AP sends an ACK first, which is followed by the DL client ACK.

Updates. Each time a successful UL and/or DL transmission happens, the corresponding path loss values are identified (leveraging Eqs 1 and 4). The old (measured or predicted) values in the AP-client and/or client-client matrices are then replaced with the up-to-date ones. Further, in Eq. 11 and Eq. 12, the actual DL and UL transmission rates replace the predicted ones, and calculations are re-done, so that AP maintains an accurate account of each clients' data rate.

V. Performance Evaluation

In this section, we present the results of our extensive evaluations using numerical simulations with Python and event-driven simulations with NS3. We present both ML and networking performance metrics. We plan to publicly release all our data and code for further outreach to the community.

A. Path Loss Generation Dataset

We create two datasets to model different path loss environments and feed them both to NS3 for network experiments (e.g., experiments concerning throughput and fairness). The two path loss datasets are generated as follows:

Measured Data. We deployed commercial wireless devices in an indoor office building of size 30×18 m^2 to gather RSS data and extract real-world path loss information over a two week time frame. The office environment is composed of cubicles, office rooms, conference room, and corridors. The network setup includes five APs and ten client devices. Some of the client devices have Line-of-Sight (LoS) to their associated APs, while others are blocked (e.g., by cubicles or walls). RSS measurements were taken at various times throughout the day and night using the 802.11ac 5 GHz channel 149 with 40 MHz bandwidth. We created 20 different topologies by randomly changing the location of APs and clients in our indoor environment. In each location, we took 10 different measurements at different hours to generate a total of 200 different snapshots in time of path loss across all devices.

Simulated Data. We used the log-distance path loss model. Specifically, for a given topology (with a given location of APs and clients), this model uses their distance and frequency to generate the path loss. We generated 100 different network scenarios, by randomly placing 10 APs and 90 clients in a 100×100 m^2 area. For each topology, we chose different path loss exponents, setting it between 1.6 and 4 to model different environments or sections of the same environment. We additionally included a shadowing random variable (SRV) in the path loss model to capture fading that can be caused by obstacles in the environment such as walls and other objects that can block or reflect the signal. The SRV is zero mean with standard deviation σ measured in dB. For a given topology and SRV σ, we created 10 instances of the path loss matrices. We experimented with three σ values of 0, 3, and 6 and considered different σ values for client-client and AP-client links.

B. Path Loss Estimation Accuracy

In this section, we evaluate the impact of missing data (i.e., level of sparsity in AP-client and client-client path loss matrices) as well as the standard deviation (σ) of the shadowing random variable in the simulated dataset. We used Python Keras and TensorFlow libraries to build the neural network architecture we discussed in Section IV-C.

Fig. 5a illustrates the impact of the missing path loss data on the average mean absolute percentage error (MAPE) for the *real-world dataset* we gathered in our indoor enterprise environment. The figure shows both the accuracy of KNN and SVD methods in predicting missing values in the AP-client matrix as well as the accuracy of the ANN in predicting missing values in the client-client matrix. Note that the ANN takes the populated AP-client matrix as input.

To generate the missing values, we first looked at each matrix (AP-client or client-client) and identified the missing values[5]. Then we increased the number of missing values to a desired level (e.g., 60%), by randomly removing some of the data points. We then show the average accuracy across all predicted points (and over different topology realizations). For the KNN method, the range of neighboring values considered varied from $K = 1$ to $K = 40$. We observe that as the percentage of missing values increases from 60% to 80%, the MAPE increased from 2% to 3% for the SVD and from 3% to 5% for the KNN. Generally, SVD outperforms KNN in terms of average MAPE by 1-2%. As a result, our ANN method used the AP-client matrix that was populated by the SVD method only. From Fig. 5a, we observe that as the percentage of missing values in the client-client matrix increases from 60% to 95%, the MAPE of the ANN increases from 3% to 6%, implying a very small data gathering overhead is sufficient for coarse-grained path loss estimation.

Fig. 5b illustrates the impact of the missing data in AP-client path loss matrix for the *simulated dataset*. Recall that the simulated dataset uses a shadowing random variable to model impact of obstacles such as walls and furniture. We consider three different values for the standard deviation of the shadowing random variable: zero, three, and six, and present the results for both KNN and SVD as the underlying models to predict the missing values. Each data point is an average across all generated topologies as we discussed in Section V-A. In each topology, we randomly removed a given number of data points to reach the desired missing percentage point in the AP-client matrix. From Fig. 5b, we observe that SVD consistently outperforms KNN in populating the AP-client matrix. Further its MAPE is less than 10% for the σ value of 6, (i.e., when there is large random deviation from the path loss model) and when 80% of data in the AP-client matrix is missing. When σ is zero (i.e., there is no shadowing) and 80% of data is missing, SVD's MAPE is only 4%.

Fig. 5c illustrates the effect of missing data on client-client path loss estimation accuracy. Here, the x-axis shows the

[5]This is because some nodes may not be within each others' transmission range and would not have the RSS and hence, path loss values.

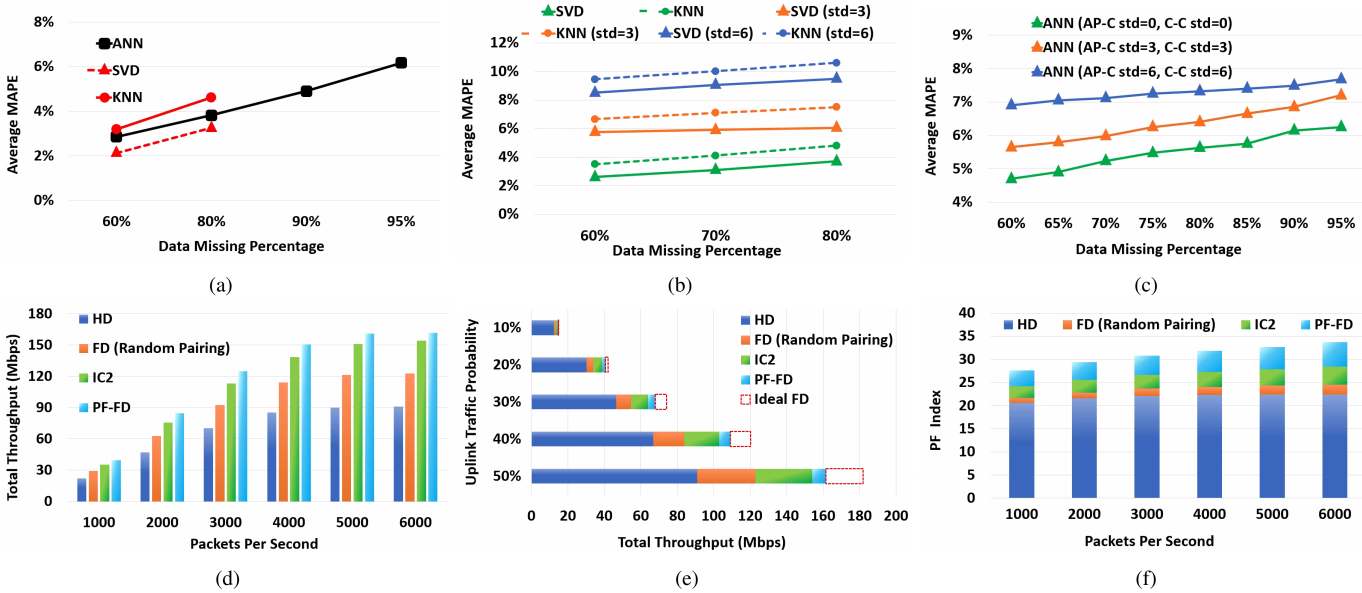

Fig. 5: **(a) Average MAPE in estimating the missing path loss values using KNN, SVD, and ANN (measured dataset), (b) Average MAPE in estimating the missing path loss values in AP-client matrix using KNN and SVD (simulated data). Shadowing variable standard deviation is shown as *std*, (c) Average MAPE in estimating the missing path loss values in the client-client matrix using ANN (simulated data), (d) Total throughput of different schemes as a function of traffic arrival rate, (e) Impact of unbalanced traffic on total throughput assuming 6000 packets per second, (f) Final proportional fair value across different schemes as a function of traffic arrival rate.**

percentage of missing entry points in the client-client matrix.

The ANN predicts missing path loss values using two sparse client-client and AP-client matrices. The AP-client matrix is populated with the SVD method, where 80% of the matrix entries are missing.

Each data point in Fig. 5c is an average across all topologies and predicted data points. We let both the AP-client and client-client path loss values to have a shadowing random variable with standard deviation σ. By increasing the value of σ from zero to six, we increase the noise in the path loss values due to the environmental factors. In the worst-case scenario, with 95% missing data in the client-client matrix and a shadowing variable σ of six in the AP-client and client-client path loss models, the ANN achieves a MAPE of less than 8%.

In summary, PF-FD effectively predicts missing client-client path loss values, even with sparse data and significant environmental uncertainties. This is achieved by leveraging both AP-client and client-client path loss values through a combination of SVD and ANN, enhancing accuracy while minimizing overhead.

C. Networking Baselines for Comparison

We updated NS3 with FD operation and conducted extensive event-driven simulations to characterize the networking performance in terms of throughput and fairness. Specifically, we implemented the following methods:

HD: APs and clients use 802.11 basic access (without RTS/CTS). No device has FD functionality. Transmission rate is the optimal data rate based on the instantaneous channel.

FD with Random Pairing: We use the MAC protocol employed in PF-FD, but pair UL and DL clients randomly.

IC2: We use the active cancellation method and historical-based pairing proposed in [7]. We update the MAC protocol to support competition by both AP and UL clients, and use logarithm as the utility to achieve proportional fairness.

PF-FD: This is our system with 80% of data points in AP-client and 90% of data points in client-client matrices missing.

Ideal FD: In this scheme, we tune UL and DL to two different channels, and on each channel we implemented HD WiFi independent of the other channel. This scheme, still incorporates the impact of MAC level collisions. But, the impact of client-client interference is completely eliminated.

In all methods, we first generate the topology and path loss values according to the measured dataset we explained in Section V-A. We then generate new topologies with varying number of APs and clients, leveraging the path loss generation methodology described in our simulated dataset (Section V-A). In these topologies, the shadowing random variable has a σ value of 6. In all of our simulations, we use 1000 byte data packets, 10 second simulation duration, and data rates according to 802.11a (from 6 Mbps to 54 Mbps).

D. Networking Performance

Throughput: Fig. 5d depicts the average total throughput (DL + UL) for the measured data (path loss values) as a function of the traffic arrival rate (packets per second). There are 5 UL and 5 DL clients.

At low traffic arrival rates, the network operates in the unsaturated region, where throughput increases proportionally with traffic. As the traffic arrival rate increases to 6000 packets/s and approaches the network's capacity, the system enters the saturated region, where the throughput plateaus at its maximum achievable value. At traffic saturation, the total throughput for FD with random client pairing is approximately $1.35\times$ that of HD mode. In comparison, PF-FD achieves a higher average gain of up to $1.77\times$ the HD throughput, while IC2 offers a gain of about $1.69\times$ the HD throughput.

For the simulated data (graphs omitted due to page limitations), we observed a similar trend in the results. The average throughput gain of PF-FD over HD mode reaches $1.72\times$ in the saturated region when the number of DL and UL clients increases to 20 each, totaling 40 clients. PF-FD's higher throughput compared to IC2 is because IC2 spends part of the AP's transmission power to cancel UL-DL client interference, whereas PF-FD uses the entire power for signal transmission. This, coupled with differences in client pairings results in a higher aggregate throughput in PF-FD.

Traffic Asymmetry: DL traffic demand typically exceeds that of UL traffic in many wireless networks. To evaluate system performance under various unbalanced traffic loads, we examined scenarios with different DL/UL traffic ratios. Fig. 5e illustrates the impact of unbalanced traffic on the total throughput at the traffic saturation threshold (6000 packets/s) for the measured data. Half of the clients are UL and the other half DL. Each DL client consistently has packets, while the likelihood of an UL client having a packet varies.

With UL-DL traffic asymmetry, the maximum throughput gain of ideal FD over HD decreases from $2\times$ to $1.2\times$ as the UL traffic probability drops from 50% to 10%. This reduction occurs because FD transmission is only viable when both paired UL and DL clients have packets to transmit. Additionally, PF-FD outperforms the other two FD schemes, achieving a 4% to 7% higher gain compared to IC2 as the UL traffic probability decreases from 50% to 10%. For the simulated data (graphs omitted due to page limitations), we observed similar trend achieving a gain of approximately 22% over IC2 at a 50% uplink traffic probability.

Fairness. We next evaluate the value of proportional fairness index (calculated as sum of log of data rates across all clients).

Fig. 5f shows this impact as a function of traffic arrival rate over measured data. We observe that the median PF index for PF-FD can reach up to $1.1\times$ that of IC2 and $1.3\times$ that of FD with random client pairing. At the traffic saturation threshold, the PF index for PF-FD reaches up to $1.04\times$ that of IC2 and $1.31\times$ that of FD with random client pairing. This is because of explicitly accounting for fairness in client pairing in PF-FD as well as its higher throughput. For the simulated data (graphs omitted due to page limitations), as the number of clients increases up to 40, we observe that the median PF index for PF-FD can reach up to $1.22\times$ that of IC2 showing the effectiveness of PF-FD in achieving improved fairness compared to other schemes under saturated conditions.

VI. Conclusion

Future wireless networks are expected to support a large number of IoT devices. FD wireless is an important technology to address the capacity demand. However, FD operation by APs with HD client (IoT) devices would create significant client-client interference, which if not treated can eliminate all FD capacity gains. This paper introduced PF-FD, an AI-based interference pattern prediction and scheduling method to address this problem in FD IoT networks. We evaluated PF-FD across two datasets and showed that compared to the state-of-the-art, PF-FD achieves up to 21% higher throughput in saturation and 22% higher proportional fairness value.

VII. Acknowledgements

This research was supported in part by the NSF (CNS-1942305) and by the David E. Wedge Vision Professorship award. We also thank the anonymous reviewers for their feedback, which helped improve the quality of this paper.

References

[1] B. Smida, A. Sabharwal, G. Fodor, G. C. Alexandropoulos, H. A. Suraweera, and C. B. Chae, "Full-Duplex Wireless for 6G: Progress Brings New Opportunities and Challenges," in *IEEE Journal on Selected Areas in Communications*, 2023.

[2] F. Meyer and V. Turau, "QMA: A Resource-efficient, Q-learning-based Multiple Access Scheme for the IIoT," in *Proceedings of IEEE ICDCS*, 2021.

[3] T. Oyedare, V. K. Shah, D. J. Jakubisin, and J. H. Reed, "Interference Suppression Using Deep Learning: Current Approaches and Open Challenges," in *IEEE Access*, 2022.

[4] S. Szott, K. Kosek-Szott, P. Gawłowicz, J. T. Gómez, B. Bellalta, A. Zubow, and F. Dressler, "Wi-Fi Meets ML: A Survey on Improving IEEE 802.11 Performance with Machine Learning," in *IEEE Communications Surveys & Tutorials*, 2022.

[5] J. Bai and A. Sabharwal, "Distributed Full-Duplex via Wireless Side-Channels: Bounds and Protocols," in *IEEE Transactions on Wireless Communications*, 2013.

[6] H. Luo, N. Garg, M. Holm, and T. Ratnarajah, "Joint Power Allocation and Beamforming for In-band Full-duplex Multi-cell Multi-user Networks," in *IEEE Transactions on Vehicular Technology*, 2024.

[7] K. C-J. Lin, K-C. Hsu, and H. Wei, "Inter-Client Interference Cancellation for Full-Duplex Networks With Half-Duplex Clients," in *IEEE/ACM Transactions on Networking*, 2019.

[8] E. Aryafar and A. Keshavarz-Haddad, "FD2: A Directional Full Duplex Communication System for Indoor Wireless Networks," in *Proceedings of IEEE INFOCOM*, 2015.

[9] T. Ohto, K. Yamamoto, K. Haneda, T. Nishio, and M. Morikura, "Generalized PF Scheduling for Bidirectional and User-Multiplexing Unidirectional Full-duplex Links," in *Proceedings of APCC*, 2015.

[10] J-T. Lim, J. Bae, Y. Choi, and Y. Han, "PF Scheduling in Hybrid Full Duplex System: Upper Bound and Simple Algorithms," in *Proceedings of IEEE WCNC*, 2017.

[11] M. Duarte, A. Feki, and S. Valentin, "Inter-User Interference Coordination in Full-Duplex Systems Based on Geographical Context Information," in *Proceedings of IEEE ICC*, 2016.

[12] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," in *IEEE Communications Surveys & Tutorials*, 2019.

[13] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in *Recommender Systems Handbook*. Springer, 2010.

[14] R. Widayanti, M.H.R. Chakim, C. Lukita, U. Rahardja, and N. Lutfiani, "Improving Recommender Systems using Hybrid Techniques of Collaborative Filtering and Content-Based Filtering," in *Journal of Applied Data Sciences*, 2023.

[15] Y. Koren, "Factor in the Neighbors: Scalable and Accurate Collaborative Filtering," in *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2010.

[16] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," in *IEEE Computer*, 2009.

[17] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," in *IEEE Journal on Selected Areas in Communications*, 2000.

[18] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 Packet Delivery from Wireless Channel Measurements," in *Proceedings of ACM SIGCOMM*, 2010.