AI and Machine Learning for Wi-Fi Optimization

Yousssef Baccouche, Jonas Andre*
*Chair of Network Architectures and Services
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: ge87qet@mytum.de, andre@net.in.tum.de

***Abstract*—Wireless Local Area Networks (WLANs), standardized in IEEE 802.11 and known as Wi-Fi are fundamental in daily internet communication. However, collision and interference may hinder Wi-Fi performance. This paper delves into models applied in layer 1 and 2 to mitigate performance issues. It presents 3 Machine Learning (ML) and Artificial intelligence (AI) based approaches. We exhibit Chen et al.'s [1] model called *Aifi* to remove interference as well as Ali et al.'s [2] approach to reduce collision. Then we proceed with Coronado et al.'s [3] mechanism that utilizes ML to determine the frames' length for each station.**

***Index Terms*—Wi-Fi, WLAN, IEEE 802.11, machine learning, deep learning, artificial intelligence, physical layer, data link layer.**

1. Introduction

The paper illustrates the deployability of AI and ML in Wi-Fi optimization, especially interference mitigation, collision avoidance, and frame aggregation. It demonstrates how each approach mitigates and enhances Wi-Fi performance. Wireless interference significantly impedes Wi-Fi performance, and mitigating its effects remains a critical challenge. Most available solutions to this issue rely on *CSMA/CA* [4]–[6], which lacks efficiency in removing interference. To diminish the issue, Chen et al. in [1] create an interference cancellation technique called *Aifi* that leverages the power of artificial intelligence to estimate and remove the interference. The models estimate the interference based only on the available physical information. *Aifi* shows promising results in reducing load latency. Beyond interference mitigation, channel access represents the most extensively addressed topic regarding the influence of ML on Wi-Fi performance. Ali et al. [2] propose a machine learning-based approach to dynamically choose the optimal contention window value to avoid collisions, while simultaneously accounting for the dynamic Wi-Fi environment. This model [7, Section A.1] shows steady throughput in dense networks. Frame aggregation in IEEE 802.11 is another area of improvement in layer 2. The two approaches provided by the 802.11 standard fail to accommodate the Wi-Fi dynamics. Thus, Coronado et al. [3] use supervised learning techniques to provide the optimal frame size for each station. The authors' model has significantly lowered the retransmission rate. The paper further analyzes in Section 2 Chen et al.'s [1] approach to remove interference. Section 3.1 studies in depth Ali et al.'s [2] model and its outcome whereas Section 3.2 decomposes the strategy adopted by Coronado et al. [3] and elucidates how ML is deployed in Wi-Fi. Then the paper recapitulates and outlines the main challenges that endanger AI and ML deployability in Section 4.

2. AI and ML in Optimizing the Physical Layer

Artificial intelligence and machine learning play a crucial role in mitigating the trade-offs and the issues that we face in the physical layer. In this section, we address interference and specify how ML and AI mitigated it based on the physical layer information.

2.1. Interference and its Mitigation

The approach to mitigate interference is based on *CSMA/CA* [4]–[6] . At first, the sender senses the carrier to determine if another device is sending at that moment. This process is called the *carrier sense (CS)*. If the channel is busy, *collision avoidance (CA)* is triggered, resulting in a delay as the transmission is postponed. However, *CSMA/CA* is limited as it can not fully eliminate the interference. Chen et al. in [1] come up with the idea to analyze the information available in the physical layer instead of probing in the air.

2.2. Background and Motivation

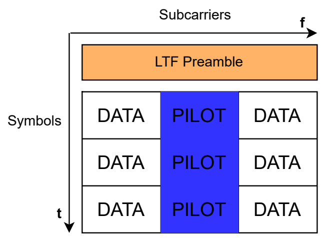

Figure 1: Physical Information in a Wi-Fi Frame [1]

The Figure 1 illustrates how the physical information is encapsulated in a Wi-Fi frame. As shown above, on the x-axis the frequency is divided into subcarriers. There is always an alternation between a block of data that carries the user's data and a pilot block used to observe the difference between the received signal and an ideal one. *Channel State Information (CSI)* [8] and *Pilot Information (PI)* [8] are extracted from the pilot subcarrier. A *Long Training Field (LTF)* preamble is spanned over all the subcarriers to synchronize them. The *LTF* preamble provides detailed specifications about the channel, mainly used for channel estimation. Symbols stored in the subcarriers and the *LTF* preamble are illustrated by the y-axis. *CSI* gives the overall channel response and provides information on how the channel affected the signal. *PI* is a subset that specifically focuses on phase shifts and is used for phase synchronization. Since the pilot subcarrier and the *LTF* preamble are based on a *Binary Phase Shift Keying (BPSK)* [1, Section 2.1], the difference between two consecutive *PI* samples or *CSI* samples is always a *fixed Phase* [1]. *BPSK* is a modulation technique based on two phases that are seperated by 180° and can also be termed *2-PSK*. However, interference may alter the phase variation in *PI* and the frequency domain, represented by *CSI*, from linear to nonlinear. This nonlinearity is explained by the channel estimation [1] as follows:

$$H_I = \frac{Y_I}{X} = \frac{HX + I}{X} = H + \frac{I}{X} \tag{1}$$

In (1) H_I illustrates the channel estimation with the presence of the interference I, whereas H refers to the interference-free channel estimation $H = Y/X$ with Y reflecting the non-interfered signal. Y_I refers to the received *LTF* signal in the presence of interference and X is the predefined *LTF* signal. The fraction I/X characterizes the effect of interference and the origin behind the phase variation's nonlinearity.

2.3. System Overview

Chen et al. in [1] contribute a mechanism called *Aifi* that starts by extracting the interference features from the *PI* and *CSI* information to estimate interference on different data subcarriers with the help of a regression neural network. Regression neural network is a machine learning technique. Depending on the training data, the regression neural network predicts the outcome, i.e., the interference estimation. In this case, the training data is a set of interfered Wi-Fi signals [1, Section 3] that are subject to non-identical interference patterns and channel conditions to guarantee the generality and adaptability of the Neural Networks (NNs). These predictions are then improved via a refinement neural network to acquire more precise interference estimations. *Aifi* conducts an interference removal process on the data subcarriers and corrects the data encoding errors across the subcarriers.

2.3.1. Interference Estimation. *Aifi* extracts the interference features [1, Section 3.1] as follows:

$$I = (H_I - H)X \tag{2}$$

When the transmitting signal X is known, I can be uniquely identified by the channel estimation [9]–[11] without interference H and with interference H_I. For this matter, *Aifi* uses a *Convolutional Neural Network (CNN)*. The idea behind *CNN* is to use the provided training data to figure out what the filters also called kernels should be. In addition, it helps with pattern recognition which in this case refers to the interference patterns. The *CNN* starts by extracting features from the interfered and non-interfered signals and then proceeds with the subtraction as shown in (2). This demonstrates that *Aifi* does not require prior knowledge about patterns of the interfering signal. The training data that serves as an input to our NNs, *Aifi* starts by collecting non-interfered *PI* and *CSI* information from Wi-Fi frames with a minimum *Signal Interference and Noise Ratio (SINR)* of 23 dB. The higher the *SINR*, the better the quality of the signal, and thus the lower the relative impact of interference and noise on the signal. The estimations collected from the interfered and non-interfered signals used in training are stored as pairs. That way *Aifi* can remove the channel estimation triggered by irrelevant factors, e. g. device mobility. If the NNs are well trained to extract the features, *Aifi* can be efficient even though the training does not englobe all the possible interfering signals. After collecting the interference features from *PI*, *Aifi* proceeds with an RNN also known as *Deconvolutional Neural Network (DeNN)*. *DeNN* performs reverse operations of *CNN* and captures the non-linearity of the data subcarriers by focusing on their *generic features* [1, Section 3.1].

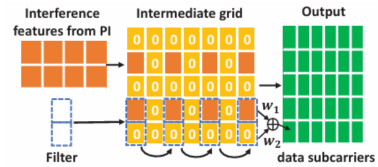

Figure 2: Regression NN [1, Section 3.1]

$$Output = w_1 \cdot x_1 + w_2 \cdot x_2 \tag{3}$$

The interference features extracted from the *PI* are expended into an intermediate grid with 0 padding, as depicted in Figure 2, to match the size of the data subcarriers. A one-Dimensional (1D) filter is applied to the intermediate grid. In a 1D filter, only the rows or the columns contribute to the output. In Figure 2, the filter is applied to the columns. The specificity of the 1D filter is its one-dimensional array kernel. The kernel length is set to 2 to process every pair of adjacent elements. In Figure 2, 1D filter has two trainable weights w_1 and w_2. Those weights are initially initialized randomly and then updated during the training process. The 1D filter slides across the intermediate grid operating as shown in (3). In this operation, x_1 and x_2 illustrate the adjacent elements over which the filter is applied. x_1 is weighted by w_1 likewise to x_2, i.e., weighted by w_2 and their weighted sum provides the output. The size of the 1D filter is chosen in a way to leverage the *continuity* [1, Section 3.1] between every two consecutive subcarriers.

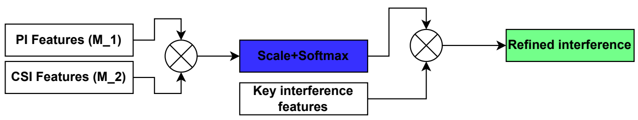

Figure 3: Feature Refinement NN [1, Section 4]

$$W = \text{softmax}\left(\frac{M_1 \cdot M_2}{\sqrt{\text{Scale}}}\right) \tag{4}$$

However, the estimation based only on the interference features captured from the *PI* lacks accuracy. To solve this issue Chen et al. [1] use the interference features from *CSI* to enhance the estimations' accuracy. This is done

 doi: 10.2313/NET-2025-05-1_10

via *stacked refinement NNs* to ensure the highest learning power level. The main purpose of the *refinement NN* is to learn the weight matrix (W) as depicted in (4). (W) reflects the correlation [1, Section 4] between *PI* and the *CSI* interference features. As shown in Figure 3 the *PI* features (M_1) and the *CSI* features (M_2) serve as input to the softmax function. The $\sqrt{\text{Scale}}$ in (4) is a constant that prevents the correlation from growing in magnitude, leading to a smaller gradient. The weight matrix is then applied to the key interference features extracted from either the *PI* or the *CSI*. As a result, a refined interference is obtained.

2.3.2. Interference Removal.

$$X = (X_I - I) \cdot W \tag{5}$$

Aifi uses a fully connected NN to imitate the equalization process in the commodity Wi-Fi that utilizes *Zero-Forcing (ZF)* [9], [12]. *ZF* tackles the signal's interference by applying the inverse of channel estimation to the signal [1, Section 3.2]. The fully connected NN ensures flexibility by computing multiple iterations and learning the optimal weights W. In addition, it emulates the equalization in the feature space offering efficient fine-tuning abilities to remove the interference. As shown above in (5), the (X_I) refers to the received signal that contains the interference, while (I) is the estimated interference from Section 2.3.1. By multiplying the subtraction with the learnable weights, the interference-free signal X is obtained.

2.3.3. Data Payload Correction. *Aifi* starts with residual interference detection and handling. Indeed, when the interference is too strong, the interference-removing process may fail to obliterate it. This is due to the *limited resolution* [1, Section 3.3] of the signal in *PI* and *CSI*. To mitigate this issue, *Aifi* incorporates a supplementary neural network to correct the decoding errors repercussed by the interference. *Aifi* then proceeds to emulate the Wi-Fi encoder. In this stage, *Aifi* utilizes a *Long-Short-Term Memory (LSTM)* network. *LSTM* is a type of a reccurent network used to mitigate the issue of long-term dependency in sequential data. In the recurrent NN the output of the previous input is fed into the node as part of the input. *LSTM* enhances this structure by adding an internal state mechanism. The state comprises three gates: the forget gate, the input gate, and the output gate. Each gate can be assigned a number between 0 and 1. *LSTM* learns the dependency between consecutive symbols and in case of an error, the *LSTM* network attempts to correct it by rebuilding the correct data payload. *Aifi* takes the output of the *LSTM* network and transforms it into the final data payload with the help of a demodulation neural network. This type of neural network tends to mimic the decoding behavior in the commodity Wi-Fi.

2.4. Real-Word Experimentation

The relevance of *Aifi* in boosting the network's performance in real-world applications, especially the loading of web pages is evaluated by Chen et al. [1]. The experiment is performed on a couple of well-known web pages e. g. Google.com, Delta.com, Twitter.com, Twitch.com [1, Section 8.2] and a predefined Wi-Fi transmission of 24Mbps. Before we delve into analyzing the outcome of the experience, we need to explain the *Frame Reception Rate (FRR)*. *FRR* is defined as:

$$\text{FRR} = \frac{\sum \text{successfully received frames}}{\sum \text{transmitted frames}} \tag{6}$$

As shown in (6), low *FRR* reflects a high impact of interference. Interference engenders high packet loss. High packet loss repercusses elevated latency. As depicted in Figure 4, in the case of low *FRR* ($FRR \leq 30\%$) we can see that *Aifi* prevails the most. When the interference is too strong ($FRR = 0\%$), we can remark that Twitter had the highest load latency ($200\,\text{seconds}$). The loading functionality of Twitter was nearly unresponsive. However, with *Aifi* deployed, the load latency plummeted ($80\,\text{seconds}$). The loading functionality of Twitch is now usable. All the web pages (Google.com, Delta.com, Twitter.com) experienced this huge load latency reduction with *Aifi* deployed, when the interference is prevailing. When the *FRR* is medium (between 40% and 70%), the load latency is reduced from around $160\,\text{seconds}$ to around $50\,\text{seconds}$ on average for all the web pages. And even in the case of high *FRR* ($FRR > 70\%$) *Aifi* ensures that the load latency is less than $3\,\text{seconds}$. It offers a good web browsing experience. To sum up, the model can transform the web pages' loading outstandingly.

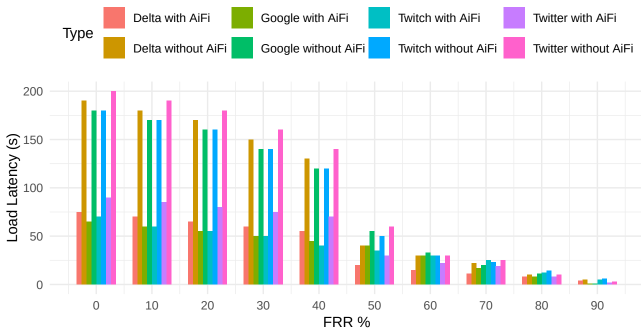

Figure 4: Latency of webpages loading [1, Section 8.2]

3. AI and ML in Optimizing the Data Link Layer

The main strength of AI and ML relies upon their power to gain knowledge rapidly, generalize it, and learn from previous experiences. In this section, we delve into the panoply of approaches that use AI and ML to optimize the data link layer.

3.1. Channel Access

Most applied approaches are tightly linked to the *Distribute Coordination Function (DCF)*. *DCF* is a *CSMA/CA*-based channel access mechanism [13]. Devices randomly select a backoff counter from the *Contention Window (CW)* range. This backoff counter represents the waiting time of this device before accessing the channel to avoid collision. The *CW* range is the following $\{0, 1, 2, \ldots, \min\{2^{c+k-1} - 1, 255\}\}$. (c) is a constant related to the physical configuration and (k) illustrates the number of transmission attempts starting from $k = 1$. One of the commonly known trade-offs is the choice of the *CW* values. Indeed, choosing small *CW* values provide quicker transmissions but increase the collision risk and thus reduce input, whereas large *CW* values minimize the probability of collision but increase the idle time of the channel leading to a reduction in the throughput [7, Section A]. The following sections provide the major findings on how to mitigate this issue with the help of ML.

3.1.1. Collision Reduction. In high-density 802.11ax WLANs, Ali et al. [2] use a combination of *Reinforcement Learning (RL)* [14] and *Intelligent Q-learning-based Resource Allocation (IQRA)*. The purpose of *RL* is to optimize a policy to yield maximum reports. The report in our case refers to the increase of throughput and the reduction of collision probability. An *RL* consists of an agent, a set of states (S), and Actions ($\mathcal{A}$). When the agent acts $a \in \mathcal{A}$, the agent transitions from one state to another. As opposed to *DCF* which resets the *CW* value every time the channel is idle, the *CW* value is now calculated by analyzing the channel collision probabilities that are based on the *Channel Observation-based Scaled Backoff (COSB)* [15] mechanism. Before scaling the *CW*, *COSB* measures how often the channel is busy. It then predicts the likelihood of collisions according to failed transmissions or retransmission attempts while accounting simultaneously for the number of active devices. That way the *IQRA* mechanism can manage between *exploring new actions* [7, Section A.1] like new *CW* adjustment and choosing the *optimal actions* [7, Section A.1] by increasing or reducing the *CW* (according to COSB). *Optimal actions* reflect the safest strategy of leveraging the already available information to be efficient in the current environment. The *IQRA* mechanism always accounts for the unstability of the Wi-Fi environment.

3.1.2. Real-Word Experimentation. Results from the ns-3 network simulator [7, Section A.1] for dense networks (with 50 stations) as shown in Figure 5 illustrate the positive effect of *IQRA* on the network throughput compared to the standard 802.11 protocol. *IQRA* provides a nearly constant network throughput during the whole Simulation time (from 0s to 60s) around $38\,\mathrm{MB\,s^{-1}}$ contrary to the standard protocol that shows relatively low throughput. In the standard protocol, the network throughput decreases from $38\,\mathrm{MB\,s^{-1}}$ to around $27\,\mathrm{MB\,s^{-1}}$.

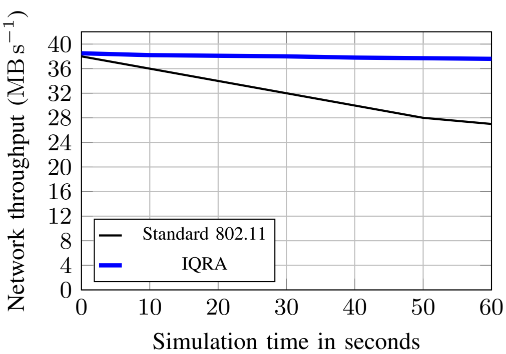

Figure 5: Network Throughput Comparison [7, Section A.1]

3.2. Frame Aggregation

Frame aggregation consists of combining smaller frames into a bigger one to improve efficiency in communication. Larger frames can reduce the overhead, however, they are more prone to transmission errors. One error in one of the subframes can cause the retransmission of the large frame. To tackle this trade-off, the 802.11 standard introduced two basic approaches. The first method is called *Aggregated Mac Service Data Unit (A-MSDU)* [16]. *A-MSDU* is more efficient as it has only one *Frame Check Sequence (FCS)* but is less robust. If one subframe is corrupted the entire frame is discarded. The second method is the *Aggregated Mac Protocol Data Unit (A-MPDU)* [16]. *A-MPDU* offers more robustness because each subframe has its own *FCS* but it introduces more overhead. The optimal approach is to apply those methods simultaneously to harmonize between robustness and efficiency. The 802.11 standard does not account for the *CSI*'s dynamic nature. Thus, to optimally choose the frame size, we leverage the dynamic features of ML. To optimally select the frame size under dynamic conditions, Coronado et al. [3] implement a *low computational complexity* [3, Section 1] technique based on a *Random Forest Regressor (RFR)* for the Modulation and Coding Scheme (MCS) settings and the frame aggregation. The MCS is a metric that reflects several parameters between the Access Point (AP) and the station including data rate, channel width, etc. The model is deployed on the management plane and fed periodically with the control plane's knowledge, e. g. channel conditions and user state. This model is considered to have *low computational complexity* because the type of ML model deployed requires minimal resources, reduced training time, and provides fast outcome predictions. A *Random Forest (RF)* is a ML model that uses an ensemble of decision trees to make its prediction. Each random forest is characterized by three parameters: the size of the nodes, the number of trees, and the number of features.

3.2.1. Model Description. The model utilizes the features like the station details as input [3, Section 4.3.5]. It assigns an *RFR* for each MCS and limits the depth of the tree to 3. By this limitation, the model avoids overfitting and reduces the complexity of the tree. The model then undergoes a tenfold cross-validation. Indeed, the data is split into 10 parts. The model is trained on 9 folds and is evaluated on the fold left. The model is deployed on the management plane. The management plane is responsible for controlling the network. An appropriate frame length is constantly provided for a specific MCS and in the next iteration, the model reconfigures itself by using the feedback from the real obtained goodput. In due course, the model can correct the next prediction with a prediction error factor [3, section 4.3.5].

3.2.2. Real-Word Experimentation. The setup consists of an AP, a Software-Defined Networking (SDN), and the ML model deployed on the management plane. A SDN ensures the configuration and monitoring of an efficient and dynamic network environment. The experiment [3, Section 5.1] consists of moving the stations within a $30\,\mathrm{m}$ radius of the AP while varying simultaneously the mobility and the MCS index (MCS-0, MCS-2, MCS-7). Each MCS index illustrates a unique combination of modulation type, coding rate, and several other parameters. Performance is evaluated on the goodput metric [3, Section 5.2]. The simulation is repeated 10 times with 14 seconds runs. The model illustrated its ability to dynamically improve goodput by about 18.36% compared to static techniques like max aggregation and no aggregation. The model tends to outperform all the other approaches, especially when the bitrate is high ($25\,\mathrm{MB\,s^{-1}}$) and the MCS is equal to 7. The approach has significantly lowered the retransmission rate.

4. Conclusion and Challenges

The impact of ML and AI is undeniable in improving Wi-Fi performance. This paper provides a comprehensive overview of 3 recent ML-based approaches deployed in the first two layers. It starts with the approach of Chen et al. [1] called *Aifi*. This approach shows its efficiency in mitigating interference and reducing 80% bit errors. *Aifi* improves the *FRR* by a factor of 18. However, the applicability to various wireless technologies may pose both an impediment and a challenge for *Aifi*. Indeed, this approach applies to only OFDM-based systems [1, section 9]. Some wireless systems have different physical structures and may provide their physical information differently. Then the paper proceeds by analyzing Ali et al.'s [2] approach. This model ensures steady network throughput by reducing the chances of collision based on a combination of *RL* [14] and *IQRA*. For the frame aggregation, we scrutinize Coronado et al.'s [3] approach. This model provides the optimal frame length for each station and ensures a low retransmission rate. However, ML and AI may raise privacy and security concerns. The pace of AI and ML deployability outran the pace of their regulations. The success of AI and ML depends on future standardizations to avoid data transfer that involves sensitive information used for the training of ML models.

References

[1] K. H. Ruirong Chen and W. Gao, "AiFi: AI-Enabled WiFi Interference Cancellation with Commodity PHY-Layer Information," *ACM Conference on Embedded Networked Sensor Systems (SenSys '22)*, pp. 134–148, 2022.

[2] R. Ali, N. Shahin, Y. B. Zikria, B.-S. Kim, and S. W. Kim, "Deep reinforcement learning paradigm for performance optimization of channel observation—based mac protocols in dense wlans," *IEEE Access*, vol. 7, pp. 3500–3511, 2019.

[3] E. Coronado, A. Thomas, and R. Riggio, "Adaptive ML-based frame length optimisation in enterprise SD-WLANs," *Journal of Network and Systems Management*, vol. 28, no. 4, pp. 850–881, Oct 2020.

[4] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans," in *Proceedings of the 7th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 1996, pp. 392–396.

[5] E. Felemban and E. Ekici, "Single hop ieee 802.11 dcf analysis revisited: Accurate modeling of channel access delay and throughput for saturated and unsaturated traffic cases," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3256–3266, 2011.

[6] X. Wang and K. Kar, "Throughput modelling and fairness issues in csma/ca based ad-hoc networks," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1. IEEE, 2005, pp. 23–34.

[7] S. Szott, K. Kosek-Szott, P. Gawlowicz, J. Torres Gómez, B. Bellalta, A. Zubow, and F. Dressler, "Wi-fi meets ml: A survey on improving ieee 802.11 performance with machine learning," p. 51, 09 2022.

[8] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11-2012, March 2012.

[9] A. Klein, G. K. Kaleh, and P. W. Baier, "Zero forcing and minimum mean-square error equalization for multiuser detection in code-division multiple-access channels," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 2, pp. 276–287, 1996.

[10] Y. Jiang, M. K. Varanasi, and J. Li, "Performance analysis of zf and mmse equalizers for mimo systems: An in-depth study of the high snr regime," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2008–2026, 2011.

[11] P. Schniter, "Low-complexity equalization of ofdm in doubly selective channels," *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 1002–1011, 2004.

[12] Y. Ding, T. N. Davidson, Z.-Q. Luo, and K. M. Wong, "Minimum BER block precoders for zero-forcing equalization," *IEEE Transactions on Signal Processing*, vol. 51, no. 9, pp. 2410–2423, 2003.

[13] N. Shenoy, J. Hamilton, A. Kwasinski, and K. Xiong, "An improved ieee 802.11 csma/ca medium access mechanism through the introduction of random short delays," in *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, 2015, pp. 331–338.

[14] C. K. Lee and S. H. Rhee, "Collision avoidance in IEEE 802.11 DCF using a reinforcement learning method," in *Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2020, pp. 898–901.

[15] R. Ali, N. Shahin, Y.-T. Kim, B.-S. Kim, and S. W. Kim, "Channel observation-based scaled backoff mechanism for high efficiency WLANs," *Electronics Letters*, vol. 54, pp. 663–665, May 2018.

[16] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on ieee 802.11ax high efficiency wlans," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 197–216, first quarter 2019.