# Wi-Fi Implementation Using LabVIEW: Midterm Report

Nathan Schimpf  
University of Louisville  
ntschi01@louisville.edu

## Abstract

One of the predominant schemes for wireless communication, Wi-Fi enables the majority of all internet traffic.As internet use increases, the need for Wi-Fi systems to provide higher throughput rates has persisted since its inception, leading to its use of Quadrature-Amplitude Modulation (QAM) and Orthogonal Frequency Division Multiplexing (OFDM) first seen in the 802.11a standard. To gain a better understanding of the core Wi-Fi physical layer (PHY), these core modulation and demodulation techniques were developed in LabView.

## Introduction

Modern physical-layer Wi-Fi implementations incorporate a number of modulation and coding techniques to maximize throughput and robustness. Figure 1.1 illustrates the overall architecture.

A screenshot of a cell phone

Description automatically generated

Figure 1.1: General Physical-Layer Architecture for Wi-Fi Systems

At its core, OFDM is a specialized technique of grouping multiple subcarriers (52, for this project’s scope) into a single channel. Each subcarrier sends individual data streams on a specified carrier frequency, in such a way that they are both tightly-packed and non-interfering.While most of the subcarriers present are data-bearing, there are four pilot subcarriers which provide a fixed, repeated signal for estimation processes. The layout of these subcarriers can be seen in figure 1.2.

It should be noted that the orthogonality of these subcarriers can be affected by changes in the physical space of transmission, whether that be the medium of the channel or movement of the devices; in both cases, the interference caused by this loss of orthogonality is known as *fading*, and is addressed via Forward-Error Correction coding and estimations of phase and frequency offsets of each subcarrier.

The transmission of bits of data by each subcarrier leverages the broad modulation scheme of Quadrature-Amplitude Modulation (QAM). In order to utilize this modulation technique, the transmitter and receiver must be synchronized, expecting a character to be transmitted at a fixed interval of time based on the period of the lowest-frequency subcarrier – the *symbol rate* of the channel. Synchronization is accomplished using a guard-time prefix, extending the first symbol prior to transmission.

A close up of a logo

Description automatically generated

Figure 1.2: Subcarrier Layout of 52-subcarrier Wi-Fi system

Given this synchronization, in-phase and quadrature (90-degree lagging) components of the subcarrier are used to determine bit values. Atransmitter may increase the complexity and bitrate of its subcarriers by using a larger range of component values.A graphical representation (1.3) of the supported modulation techniques is provided to better illustrate this.

A screenshot of a cell phone

Description automatically generatedFigure 1.3: Illustration of Quadrature-Amplitude Modulation (QAM) techniques in Wi-Fi systems

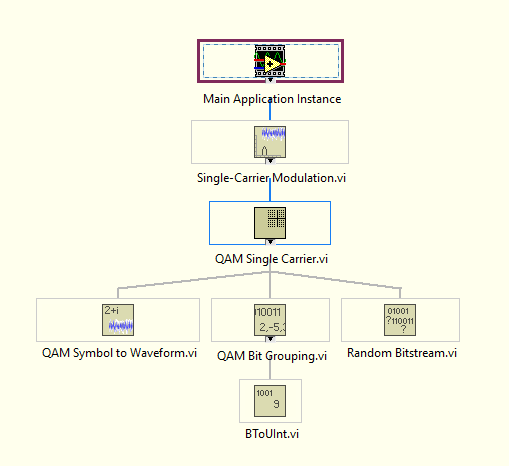
By developing the core QAM and OFDM modulation and demodulation techniques, this project aims to provide a minimal-feature simulation environment for students to examine and locally experiment with typical wireless systems. The exact features this project aimed to incorporate are described below.

The project will:

* Capabilities for all subcarrier QAM modulation schemes in figure 1.3.
  + Two implementations will be provided: one based around the RF Communications Toolbox, and one custom implementation using LabView’s base components.
  + These will be selectable, based on a front-panel input.
  + This will be visible via a front-panel constellation diagram.
* OFDM on a 52-subcarrier system.
  + This will be illustrated with an FFT waveform chart.
* A guard time for the beginning of each transmission.

## Midterm Status

At the time of publication, the implementation includes a *Top VI* which allows for signal simulated noise and channel fading between the transmitter and receiver; *Top VI* implements the high-level QAM modulator (*QAM Mod*), OFDM transmitter (*OFDM TX*), OFDM receiver (*OFDM RX*), QAM demodulator (*QAM Demod*), and various helper functions, with VI hierarchy illustrated in figure X.X. At a cursory view, *Top VI* allows the user to specify QAM parameters (such as samples per symbol and complexity), while also providing basic analysis tools such as a frequency-domain waveform chart and constellation graph of the receiver; validation indicators are provided as well, referencing the number of correctly-received subcarriers and correctly-decoded bits. Finally, *Top VI* provides a control for selecting which QAM de/modulator is used: the Labview-provided de/modulation tools in the *RF Communications Toolbox (MT Modulate)*, or the custom-written de/modulation tools. It should be noted that Labview does not provide an OFDM implementation, which therefore must use the custom Vis.

Figure 1.4: Current VI Hierarchy

Supporting the core transmitter/receiver pair is *Symbol to DataLen*, determining the number of bits need to provide 48 data-bearing subcarriers during the QAM modulation. Though the number of bits per QAM symbol should only vary by QAM complexity in accordance with table X.X, some control was altered due to *MT Modulate* requiring a minimum of 4 symbols (2 bits) in it’s symbol map. As a result, the dual-case structure in figure X.X was implemented to scale the number of bits per symbol.

|  |  |  |
| --- | --- | --- |
| QAM System | QAM Symbol Range | Bits required per symbol |
| BPSK | [-1+j0, 1+j0] | 1 (homebrew), 2 (MT Modulate) |
| QPSK | [-1-j1, 1+j1] | 2 |
| 16-QAM | [-2-j2, 2+j2] | 4 |
| 64-QAM | [-4-j4, 4+j4] | 6 |
| 256-QAM | [-8-j8, 8+j8] | 8 |

### QAM Modulation

Handling the QAM modulation is *QAM Mod,* which utilizes several subVIs to craft the individual QAM weights from a randomly-generated bitstream; insert pilot and center frequency weights according to figure X.X; and generate a waveform from weights. The custom implementation illustrates this process rather symbolically in figure X.X.

More complicated is the QAM process using *MT Modulate – QAM*, which requires two key parameters: filtering coefficients, which must be generated using *MT Generate Filter Coefficients*; and QAM parameters, including the samples per symbol and symbol listing. Providing the symbol listing was easily done with constant arrays at first. However, for 256-QAM, it became more practical to design and read from a CSV file than to generate the array in LabView; figure X.X illustrates the *MT Modulate* implementation, including reading the symbol listing from a local CSV file.

Examining *QAM Bit Grouping* illustrates how symbol weights are generated. The VI iterates through all bits in the received bitstream and – using case structures and shift registers, converts grouped bits into unsigned integers, which are then scaled to span the appropriate QAM constellations in figure X.X. This process is exemplified in figure X.X.

Iarray: insertaion index of array

NSC: current total of subcarriers

SSC: samples per subcarrier

Wi: width of inserted subarray

*WSC*: width in terms of subcarriers

Inserting pilot subcarriers and the center frequency are accomplished in similar manner, though with separate Vis. Each symbol fragment is iterated through (for pilot insertion, 48 subcarriers; for center insertion, 52 subcarriers), and each insertion is accomplished based on equations X.X and X.Y. This operation is performed 4 times per OFDM frame for pilot insertion (figure X.X), and once per OFDM frame for center insertion (figure X.X). Because insertion may happen either before or after the QAM waveform has been generated, the equations may scale by *sSC*: setting this as the system Samples-per-Symbol allows for insertion after waveform generation, or setting this as 1 allows for use before generation.

For the custom QAM implementation, *QAM Waveform Generation* is a necessary component. In effect, it elongates the upsamples and duplicates each value from the symbol listing according to the number of samples per symbol; this is verified via modulo operation and illustrated in figure X.X.

#### QAM Mod Execution

When running *QAM Mod*, the custom implementation performs as expected, providing a series of square pulses representing each frequency-domain subcarrier at each complexity. As seen in figure X.X, the pilot and center insertions appear to work properly as well. By all appearances, the data produced is identical to that with *MT Modulate – QAM* (figure X.X).

### OFDM Transmit

Though initially more complicated, *OFDM TX* effectively became an Inverse Fast-Fourier Transform (IFFT) with some scaling and documentation in the waveform cluster output. The orthogonality principle of OFDM is achieved by aligning subcarriers at integer-multiple frequencies of one another; as seen in figure X.X, the only other operation is redefining waveform increments (dt). The increment is redefined by equation X.X, a ratio of the period of the symbol rate *RS* (the slowest frequency in contained by the OFDM frame, and therefore duration of the frame) and total samples in the IFFT of the signal.

initial implementation efforts achieved this by manually generating each subcarrier based on independent QAM waveforms; this approach was quickly abandoned due to the decentralization of subcarrier data (providing no central graph to observe each OFDM frame), as well as runtime inefficiency (generating 48 times the number of intended OFDM frames, which required several minutes and approximately 4 GB of program memory).

After further discussion and research, the formula for an IFFT was recalled as

which effectively sums the received frequency-domain signal as an average of complex sinusoids with frequency , where k is the present sample of a subcarrier, and N is the total number of samples. For simplicity’s sake, this implementation was used; the initial implementation has been preserved under “A-Simple-WiFi-PHY\OFDM\DO NOT USE\,” though will not be further analyzed.

When executed, *OFDM TX* provides an accurate time-domain signal that combines each symbol within the OFDM frame, pictured in the figure X.X.

### OFDM Receive

Given the realizations from the OFDM Transmit VI, *OFDM RX* was implemented as a Fast-Fourier Transform (FFT) with time scaling defined by equation X.X. This implementation is seen in figure X.X.

When executed in the *Top VI* system, the frequency-domain waveform is accurately reproduced with marginal offsets; these offsets are more visible from the output constellation chart. When analyzed on a BPSK transmission (figure X.X), the imaginary and real component are seen to drift by ±1 fV, likely due to differences in execution time.

### QAM Demodulation

*QAM Demod* handles all signal processing to take the frequency-based QAM waveform, extract symbols from each subcarrier in the waveform, and convert these (data-bearing) symbols to a bitstream; the VI provides this final bitstream, as well as the extracted subcarrier symbols for comparison.

When executing *QAM Demod* based on the *RF Communications* Toolbox Vis, the entire process is integrated into *MT Demodulate – QAM*. As in *QAM Mod*, the VI requires a list of filter coefficients, samples per symbol, and symbol listing be provided; this is accomplished by the same methods, depicted in figure X.X. To output only data-bearing bits, an additional VI, *MT Modulate – Data Bit Extraction*, was developed to selectively remove pilot and center-frequency bits from the bitstream; this is implemented as in figure X.X.

The signal chain for providing data-bearing subcarriers for the *RF Communications* Toolbox implementation is almost identical to the custom implementation for bitstream recovery. As seen in figure X.X, the waveform is first decimated (*FFT Downsample,* figure X.X) to one sample per symbol, which is then split into pilot and data subcarriers (*QAM Components*, figure X.X) before being iteratively rounded (*Double to Integer,* figure X.X) and converted to unsigned integers (*Singed to Unsigned,* figure X.X), then finally the output bitstream (*QAM Symbol to BitStream,* figure X.X; *Unsigned to Binary,* figure X.X).

When executing the VI in the context of *Top VI*, some flaws become apparent in the overall operation. Notably, the *MT Modulate* implementation is consistently missing 3 subcarriers each execution. Comparison from the front panel indicates that the subcarrier weights, in both implementations, are not correctly received in comparison to the transmitted weights; this inaccuracy is widened with more complex QAM systems, where up to 22 subcarriers have been incorrect in a 266-QAM execution (figure X.X).

### Conclusion

While the operating principles are well-illustrated by the project, there are significant bugs to still be addressed to be useable as a simulation environment. Significant effort has been made, however, to make sure this project code is readable and well-structured; as such, it still provides a value as an educational tool for understanding how these signal processing techniques function and are implemented.

## License and availability

In order to provide this project to receive feedback and share the project for academic and personal reference, the LabView code has been published to GitHub under a General Public License. It can be accessed at: <https://github.com/schimpfen/A-Simple-WiFi-PHY>

# References

|  |  |
| --- | --- |
| [1] | X. Chen, in *Coding in 802.11 WLANs*, Maynooth, Maynooth: National University of Ireland Maynooth, 2012, pp. 2-10. |
| [2] | Wireless Village, "Wireless Village Hacking Lab: B-Sides DC 2018 Edition," in *B-Sides DC*, Washington D.C., 2018. |
| [3] | Cisco, "Cisco Viual Networking Index: Global Mobile Data Traffic Forecast Update, 2017-2022," Cisco, San Jose, 2017. |
| [4] | S. e. a. Firdose, "An OFDM Transmitter and Receiver using NI USRP with LabVIEW," *International Journal of Engineering Research and Technology,* vol. 6, no. 13, 2018. |
| [5] | B. Bloessl, M. Segata, C. Sommer and F. Dressler, "An IEEE 802.11a/g/p OFDM Receiver for GNU Radio," in *Proceedings of the Second Workshop on Software Radio Implementations Forum*, Hong Kong. |
| [6] | M. Gast, "13. 802.11a and 802.11j: 5-GHz OFDM PHY," in *802.11 Wireless Networks: The Definitive Guide*, Cambridge, O'Reilly Media, 2013, pp. 283-307. |
| [7] | National Instruments, "Lab 15: Principles of OFDM," in *Lab Manual: Communications Principles Using the EMONA Communications board for NI ELVIS III*, National Instruments, 2019. |