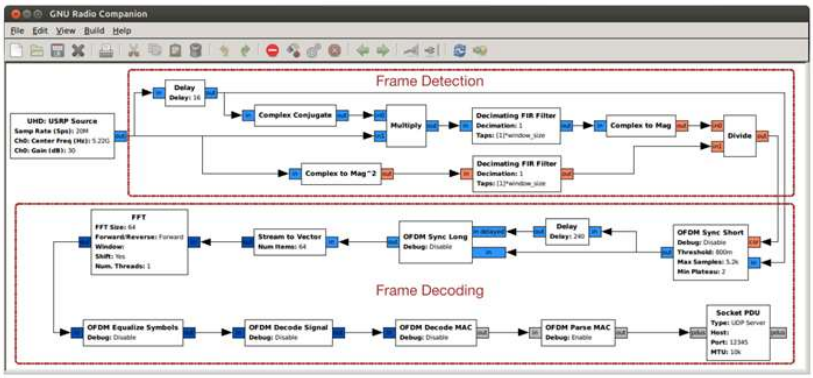
Imprimir

Main Project

Acording to the design parameteres:



1. Overview

We can visualize the Bandwidth for the Orthogonal Frequency Division Multiplexing (OFDM) receiver implemented in GNU Radio that supports both WiFi with a bandwidth of 20 MHz and IEEE 802.11p DSRC with a bandwidth of 10 MHz. It supports both WiFi with a bandwidth of 20 MHz and IEEE 802.11p DSRC with a bandwidth of 10 MHz.

There 12 unused sub-carries. Theses are of 2 types: DC(Center Subcarriar) and edge subcarries

Center Sub-carrier is not used, the center subcarrier in 802.11a is intentionally left unused because it corresponds to DC (0 Hz) in the baseband signal. In practical hardware, imperfections in the transmter and receiver, such as mixer leakage or small DC biases, can introduce a DC offset, which appears as unwanted energy right at 0Hz.

If that subcarrier were used carry data, the DC offset could easily easily distort or completely mask the transmitted symbol, making reli iable demodutation impossibo

This has 2 benefits:

It prevent hardware DC bias from distorting a data tone

It creates a notch (a visible hole) at the center of the spectrum

Edg subcarriers: the outermost 11 subcarriers at both end of the spectrum are set to zero

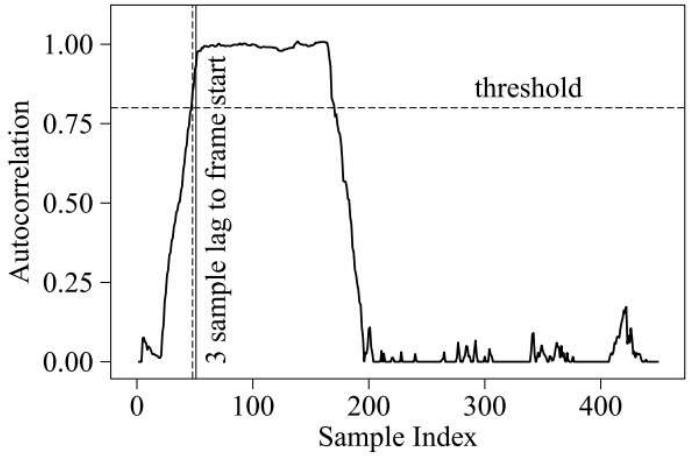
Frame Detection: The first task in the receive chain is to actually detect the start of an OFDM frame. Each IEEE 802.11a/g/p frame starts with a short preamble sequence, which consists of a pattern that spans 16 samples and repeats ten times. The employed frame detection algorithm has been introduced in [3]. It is based on the autocorrelation3 of the short training sequence.Following [3, Algorithm 1], we exploit this cyclic property and calculate the autocorrelation value a of the incoming sample stream s with lag 16 by summing up the autocorrelation coefficients over an adjustable window Nwin

$$\mathrm{a}[n] = \sum_{k=0}^{N_{\mathrm{win}}-1} \mathrm{s}[n+k]\,\bar{\mathrm{s}}[n+k+16].$$

The summation over the window (which finally results in the calculation of a moving average) acts as a low-pass filter. We experimented with different window sizes and found 48 to work well. Due to the cyclic property of the short training sequence, the autocorrelation is high at the start of an IEEE 802.11a/g/p frame. In order to be independent of the absolute level of incoming samples, we normalize the autocorrelation with the average power p and calculate the autocorrelation coefficient c as

$$\mathrm{p}[n] = \sum_{k=0}^{N_{\mathrm{win}}-1} \mathrm{s}[n+k]\,\bar{\mathrm{s}}[n+k];$$

$$\mathrm{c}[n] = \frac{|\mathrm{a}[n]|}{\mathrm{p}[n]}.$$



2. Understand the OFDM Sync Short block

2.1What the Block Does

The **OFDM Sync Short** block performs the **first stage of synchronization** in the IEEE 802.11a receiver. It detects the **start of a new frame** and provides an estimate of the **coarse frequency offset**.

It identifies the **Short Training Sequence (STS)** at the beginning of each Wi-Fi frame, which consists of **10 identical repetitions of 16 sample patterns**.
By measuring the autocorrelation between consecutive repetitions, the block detects the periodic structure of the STS.

It performs:

2.1.1 Autocorrelation calculation:

$$R[n] = \sum_{k=0}^{N_{\text{win}}-1} r[n+k] \cdot r^*[n+k+16]$$

2.1.2 Power estimation:

$$P[n] = \sum_{k=0}^{N_{\text{win}}-1} |r[n+k]|^2$$

2.1.3 Detection condition:

$$M[n] = \frac{|R[n]|^2}{(P[n])^2} > \text{Threshold} \Rightarrow \text{Frame Detectado}$$

2.1.4 Coarse frequency offset estimate:

$$\Delta f_{\text{coarse}} = \frac{1}{2\pi N_d} \arg(R[n])$$

2.2 Tests and Observations

- Smaller (R[n]) → faster but less stable detection;
- Higher threshold → fewer false alarms but more missed frames;
- Performance degrades under low SNR or large offsets.

2.3 Meaning of the Threshold

The **threshold** defines the **minimum normalized correlation** required to decide that a Wi-Fi frame has arrived.

- **Low threshold:** more sensitive → may detect noise as frames.
- **High threshold:** more selective → may miss real frames in noisy environments.

Typical thresholds: **0.75–0.9**, depending on the signal-to-noise ratio (SNR).

2.4 Effect of Changing the Threshold

- **Decreasing the threshold:**
  - Increases sensitivity, allowing detection at lower SNR.
    − Causes **false detections** from noise or interference.
- **Increasing the threshold:**
  - Reduces false alarms.
    − Causes **missed detections** when the signal is weak or distorted.

Experimental results show:

- For **high SNR (>20 dB)** → threshold ≈ 0.9 works well.
- For **low SNR (<10 dB)** → threshold ≈ 0.7 is preferred.

2.5 Limitations of the Approach

This simple autocorrelation-based approach has intrinsic limitations:

1. **Noise sensitivity:** correlation degrades under low SNR.
2. **Channel distortion:** multipath propagation destroys perfect periodicity.
3. **False correlations:** repetitive noise can mimic STS patterns.
4. **Partial frequency correction:** coarse offset estimation leaves residual errors, to be corrected later by the **OFDM Sync Long** block.

2.6. Case Where It Breaks

The block fails in environments with:

- **High frequency offset** (e.g., >250 kHz), or
- **Very low SNR** (<5 dB).

In this case:

- The phase rotation between repetitions becomes unstable,
- The autocorrelation term (R[n]) decreases sharply,
- The metric (M[n]) never exceeds the threshold → **frame not detected**,
- Or, conversely, false triggers appear in noise peaks.

A practical failure example occurs in **mobile or Doppler-shifted channels**, or when transmitter and receiver oscillators are poorly calibrated — the simple periodic correlation approach breaks and synchronization is lost.

2.7 Conclusion

The **OFDM Sync Short** block is crucial for initial frame synchronization and coarse frequency correction. Proper tuning of (R[n]) and threshold ensures optimal detection performance.

3. Understanding the OFDM Sync Long Block

OFDM Sync Long block is responsible for **fine frequency correction** and **symbol synchronization refinement.**

3.1 Working Principle

After STS detection, the **Long Training Sequence (LTS)** follows, formed by two identical OFDM symbols.

Fine frequency offset is estimated as:

$$\hat{f}_{\text{offset}} = \frac{1}{2\pi N} \arg\left(\sum_{k=0}^{N-1} r[k]\, r^*[k+N] \right)$$

3.2 Implementation

- Implemented after Sync Short detection;
- Uses a **delayed input** to align both halves of the LTS;
- Applies fine frequency correction to the received signal.

3.3 Configurable Parameters

Parameter	Function	Effect
Window size	Correlation length	Precision vs. noise immunity
Delay (≈64)	Aligns LTS halves	Incorrect delay → estimation errors
Threshold	Transition control	Stability of synchronization

3.4 Limitations and Observations

- Sensitive to poor coarse correction;
- Assumes channel stability during LTS; performance may degrade in time-varying channels.

5.5 Conclusion

The **OFDM Sync Long** block enhances synchronization accuracy and corrects fine frequency offset.
Proper parameter and delay settings are essential for precise FFT alignment and robust demodulation.