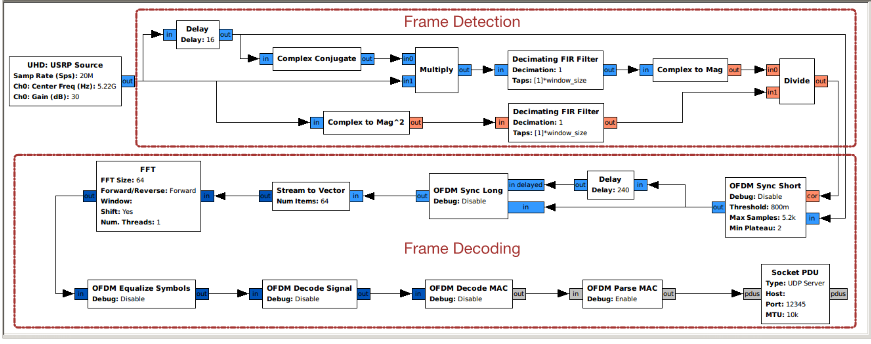
# An IEEE 802.11a/g/p OFDM Receiver for GNURadio

Bloessl et. Al ACM SIGCOMM 2013

* Wifi transmitters exist already
* Matt Ettsu has built a receiver, but not to the a/g/p specs (20 MHz Bandwidth)
* **Lookup**: (maximum likelihood estimation, pseudonoise)
* **Contributions**
  + 802.11 a/g/p receiver
  + 20 MHz Bandwidth, no changes to N210 image
  + Supports PHY and decodes MAC

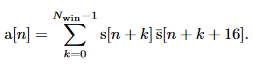


## Receiver Overview

* Split into frame detection and decoding
* **Stream Tagging:** provides dataflow metadata. Utilized to indicate frame starts
* **Message Parsing:** asynchronous message parsing to encapsulate packets. Selectively used against stream-based processing (GNURadio default)
* **Vectorized Library of Kernels (VOLK):** a c++ Single-Instruction, Multiple Data (SIMD) allows for TONS of optimization of vector math/signal processing

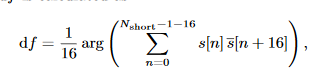
## Frame Detection

* Every 802.11a/g/p frame begins with a preamble/training sequence. 16 samples repeated 10 times



* Detection based on autocorrelation delayed by 16 (right), acts as Low-Pass Filter
* Averaged against (autocorrelation with no conjugate delay)
* Maximum autocorellation during STF/Sequence, ‘plateu’

## Frequency Offset Correction (N/A



* **OFDM Sync Long**: oscillators of tx and rx bound to vary. Implements the algorithm
* During the short sequence, symbols are periodic. This autocorellation **should be purely real.** Taking the Argument is therefore the phase shift from the ideal (0). Cyclic average taken.

## Symbol Alignment

* Still **OFDM Sync Long. 80 sample span** (16 cyclic prefix, 64 data)
* **Goal:** calculate symbol start, extract symbol, pass to FFT.
  + **Accomplished with LTF/S:** 64-sample pattern, repeated x2.5
  + **Algo…..**

## Phase Offset Correction

* **FFT and OFDM Equalize Symbols**
* Uses Pilot Carriers to calculate Phase Offset, allowing for symbol alignment
  + Linear regression

## Channel Estimation

* **OFDM Equalize Symbols:** Correct carrier magnitude
  + **ESP FOR QAM-**16/64/256
  + **Restricted to BPSK/QPSK**
  + Removes non-data subcarriers as well

## Signal Decoding

* **OFDM Decode Signal:** Decode convolutional code. Uses IT++ library
* **On correct decode:** annotated stream with tuple and length fir payload decoding (req: valid code rate, correct parity bit)

## Frame Decoding

* Payload Decoding substeps below – OFDM Decode MAC
  + Demodulation: receives symbol[48] -> floating-point values
  + Deinterleaving: permutation of symbols based on MCS
  + Convolutional Decode/Puncturing: IT++
  + Descrambling: ???
  + Output: to GNURadio Message
  + **Final: UDP sink, view payload w/ netcat**

## Interoperability, Performance, Conclusion (skipped)

**More on autocorellation [3]**

# An OFDM Transmitter/Receiver Using NI USRP with LabVIEW

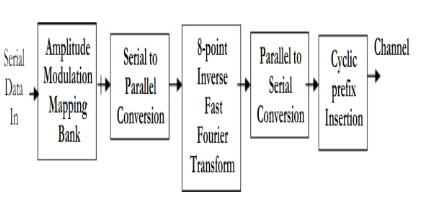
Firdose, Sushma International Journal of Engineering Research and Technology NCESC 2018

## Intro

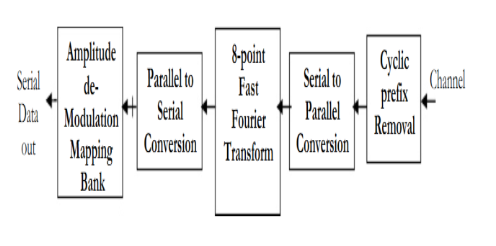
* Mcm, orthogonality,
* Effective in channels with frequency-selective fading
  + Treats channel as flat-fading for each subcarrier

## Literature Survey - meh

## Methodology



* Right – transmitter and Receiver flowgraphs
* Their design skips cyclic prefixes. Yours is going to as well, at least for now.



# NI Lab 13: Principles of OFDM

<https://learn.ni.com/teach/resources/1310/principles-of-ofdm>