Future Internet Winter Camp 2009

Session 4: 무선 테스트베드

GNU Radio를 이용한 Cognitive Radio Network 구현

Sunghyun Choi, Kwanghun Han, Hyewon Lee, Suhwook Kim, and Kyuhwan Kwak

Multimedia & Wireless Networking Lab.

Seoul National University

Table of Contents

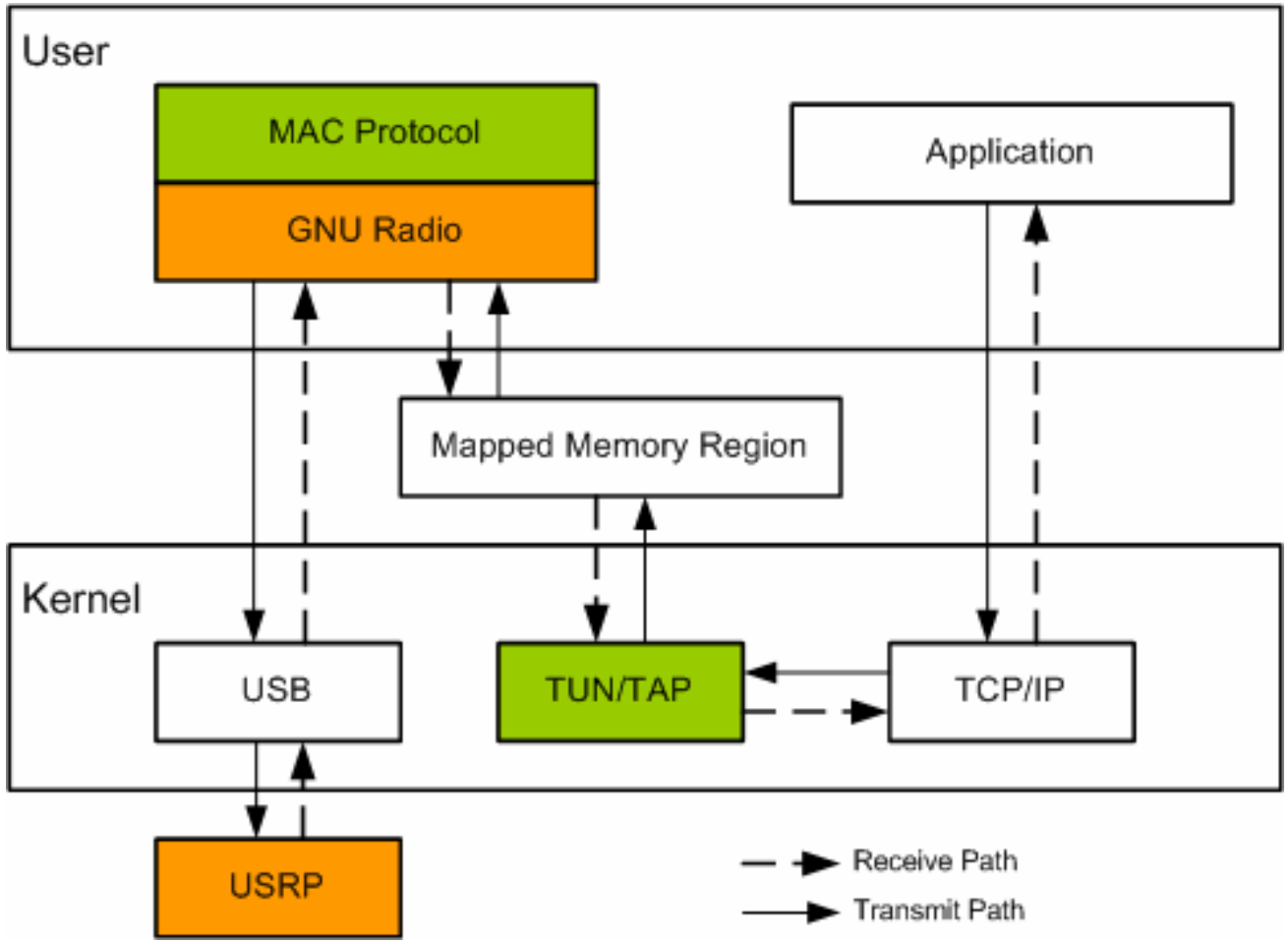
Introduction

GNU Radio

- An open source software toolkit for building Software Defined Radio (SDR)
- Signal processing package
- Supports Linux and Mac OS
- Programming languages
  - C++
    - Performance critical applications
    - Signal processing blocks
  - Python
    - Using Python for creating flow graphs
    - Also used for creating GUI's
    - and other non performance critical applications

Introduction

◆ GNU Radio Data Path



Introduction

◆ Available GNU Radio Blocks

Source / Sinks

Signal
Noise
Null
File
Network
Packet
Video
Audio
USRP
FFT
Scope

Type Conversions

Complex <-> IntShort / Real/ Imag
Complex <-> Mag / Arg
Float <-> Complex / Char / UChar
Packed <-> Unpacked
Symbols <-> Chunks
Vector <-> Stream <-> Streams
Interleaver / Deinterleaver
Complex Conjugate

Modulation

WFM / NBFM
AM / PM / SSB
FSK / PSK / QAM
GMSK / VSB-8 / OFDM

Coding

Differential
Trellis
Viterbi
BCJR
Reed Solomon

Math operations

Add
Subtract
Multiply
Divide
Log

Filters

FIR
IIR (single pole)
FFT/IFFT
Freq. Translating FIR
Rational Resampling FIR
Root Raised Cosine
Hilbert
Power Squelch

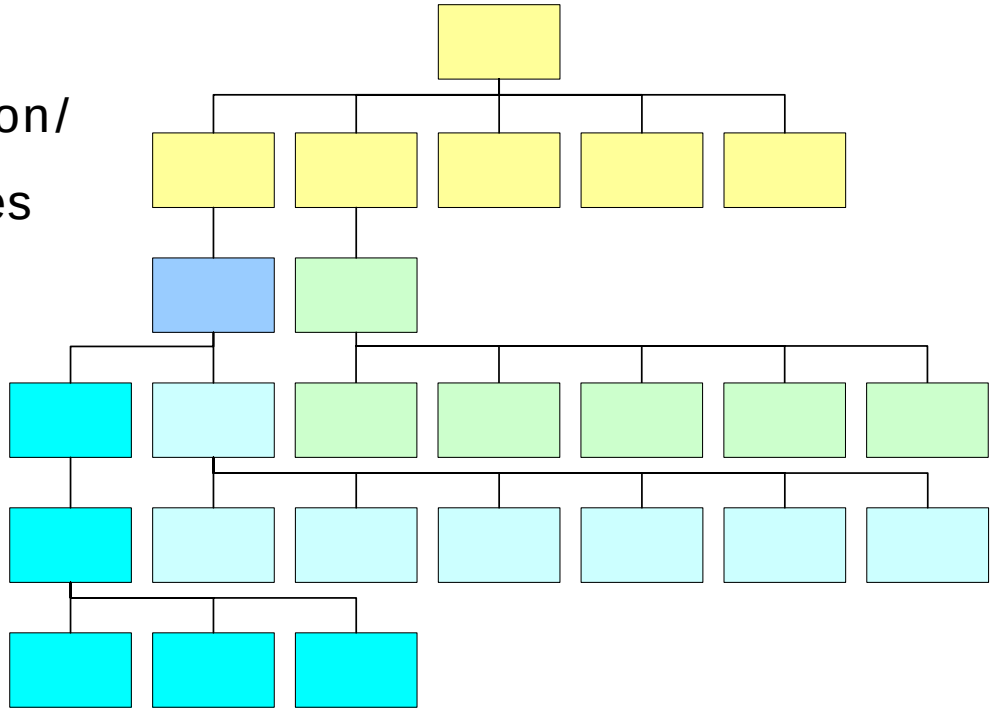
Misc.

M&M Clock Recovery
AGC
PLL
Costas Loop
Adaptive Equalizer

Introduction

Directory Structure of GNU Radio

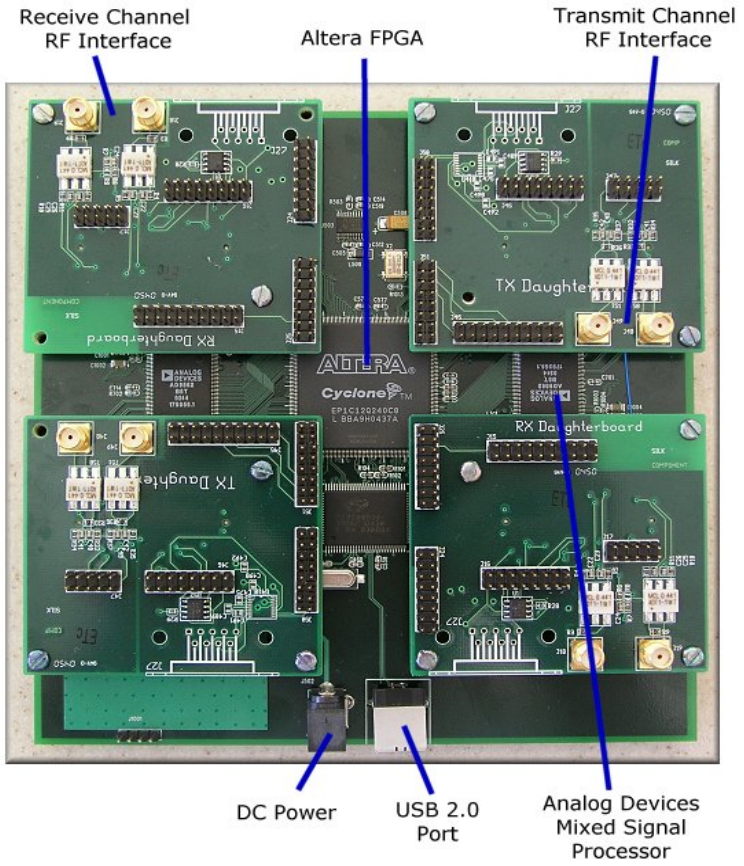
- /gr-utils/
  - Plotting
- /gnuradio-examples/python/
  - Execution example files
  - Written in Python
- /gnuradio-core/
  - lib
    - Core blocks
    - Written in C++
  - python
    - Python modules
    - Modulation, OFDM, Packetizing, etc.



Introduction

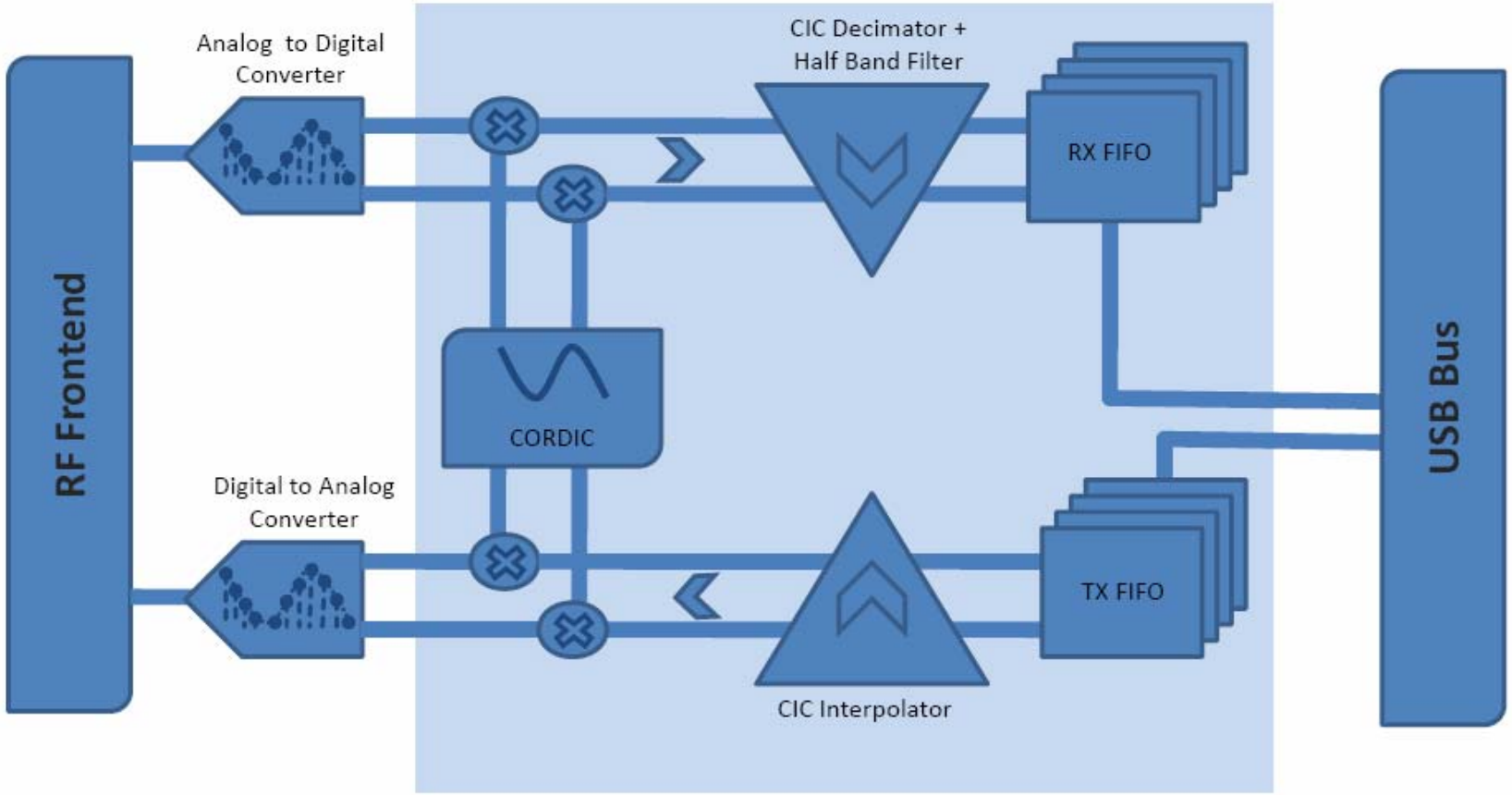
GNU Radio Hardware Platform

- Universal Software Radio Peripheral (USRP)
  - The USRP is a device developed especially for GNU Radio
  - Supporting
    - Two Receive daughter boards
    - Two Transmit daughter boards
    - Full duplex
    - 4×ADC, 12 bit @ 64 MS/S
    - 4×DAC, 14 bit @ 128 MS/S
    - FPGA
    - USB 2.0 interface to host PC
      - It can sustain 32 MB/sec



Introduction

◆ USRP Architecture



Introduction

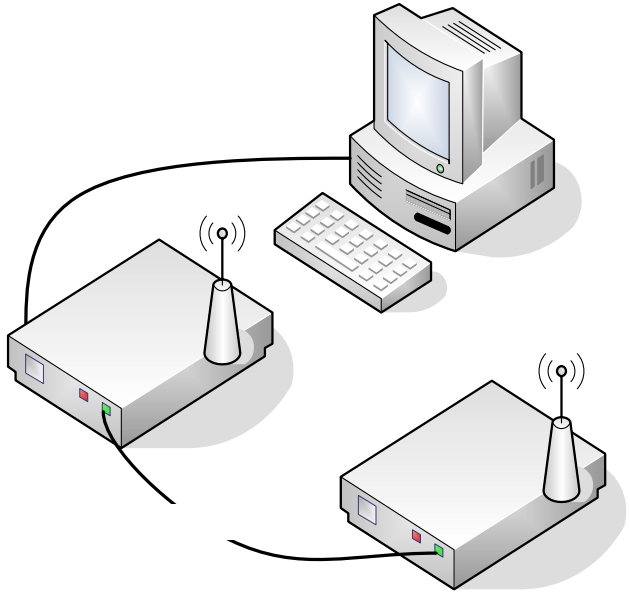
USRP Daughter Boards

- RF frontend of USRP
  - Transceiver: RFX series
    - RFX400: 400-500 MHz, 100 mW output
    - RFX900: 800-1000 MHz, 200 mW output
    - RFX1200: 1150-1450 MHz, 200 mW output
    - RFX1800: 1.5-2.1 GHz, 100 mW output
    - RFX2400: 2.3-2.9 GHz, 20 mW output

Introduction

USRP2

- Using gigabit Ethernet interface instead of USB 2.0 interface
  - Maximum bandwidth is not limited by interface between host and USRP2 anymore
    - Supporting 25 MHz bandwidth (8 MHz for USRP1) for one USRP device
    - A gigabit Ethernet interface can be used commonly for more than 2 USRP devices
      - Secondary USRP2 device can be connected with primary USRP2 device using MIMO cable



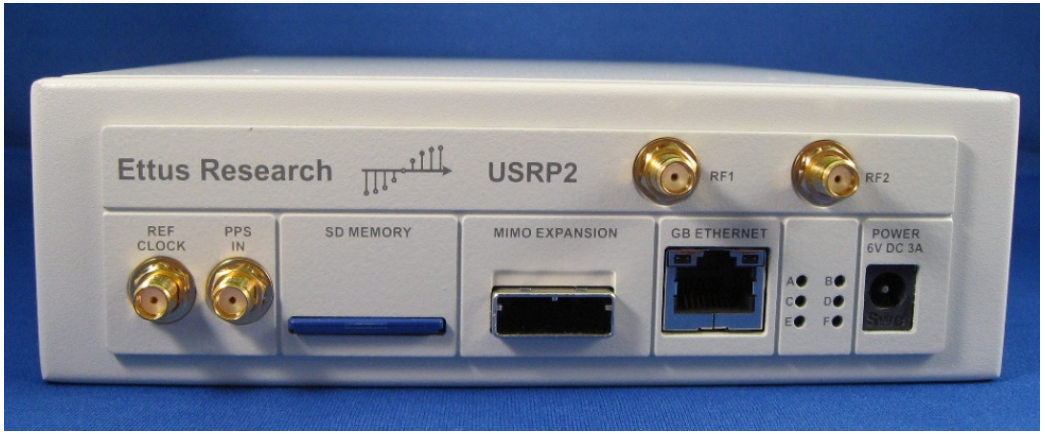
Introduction

USRP2

- Possibly supports stand-alone mode in the future
  - It has a SD card
  - GNU radio does not support this yet
    - Being developing
  - Too long processing delay problem can be reduced if stand-alone mode is supported
- MIMO capability using multiple USRP2 boards
  - USRP2 has MIMO cable port to exchange clock and data among USRP2 boards
  - GNU radio software does not support this mode yet.
    It is under development now

Introduction

◆ USRP2

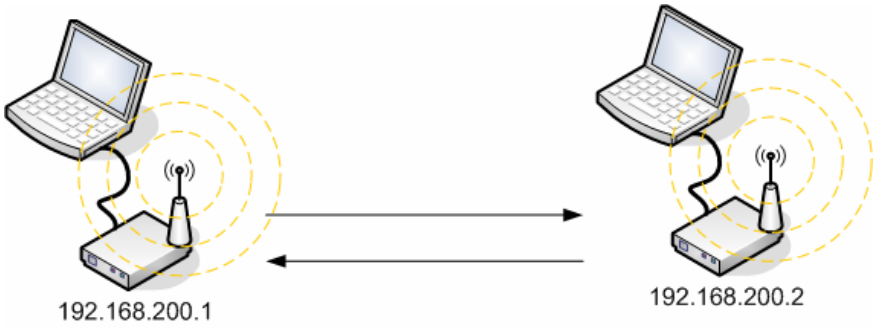


	USRP1	USRP2
Interface	USB 2.0	Gigabit Ethernet
FPGA	Altera EP1C12	Xilinx Spartan 3 2000
RF Bandwidth to/from host	8 MHz @ 16bits	25 MHz @ 16bits
Cost	$700	$1400
ADC Samples	12-bit, 64 MS/s	14-bit, 100 MS/s
DAC Samples	14-bit, 128 MS/s	16-bit, 400 MS/s
Daughterboard capacity	2 TX, 2 RX	1 TX, 1 RX
SRAM	None	1 Megabyte
Power	6V, 3A	6V, 3A

Implementation of CR Network

Simple Performance

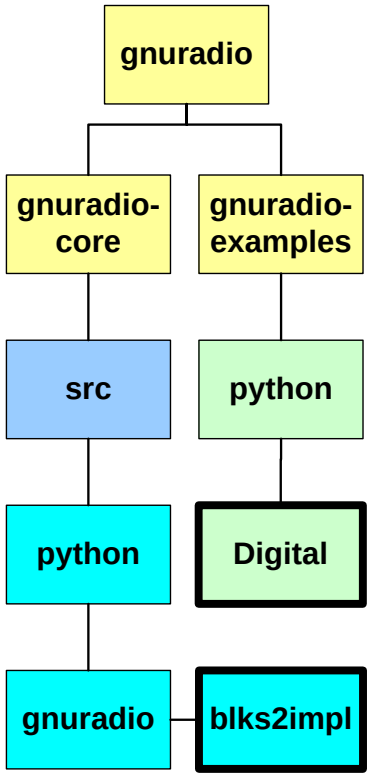
- Test environment
  - Ubuntu 8.10 (Kernel ver. 2.6.27)
  - Intel Pentium dual core 2.0 GHz
  - 1 GB Memory
- Delay
  - Ping test
  - RTT: ~10 msec
- Throughput
  - ~1 Mbps (without any MAC protocol)



Implementation of CR Network

Demo Environment

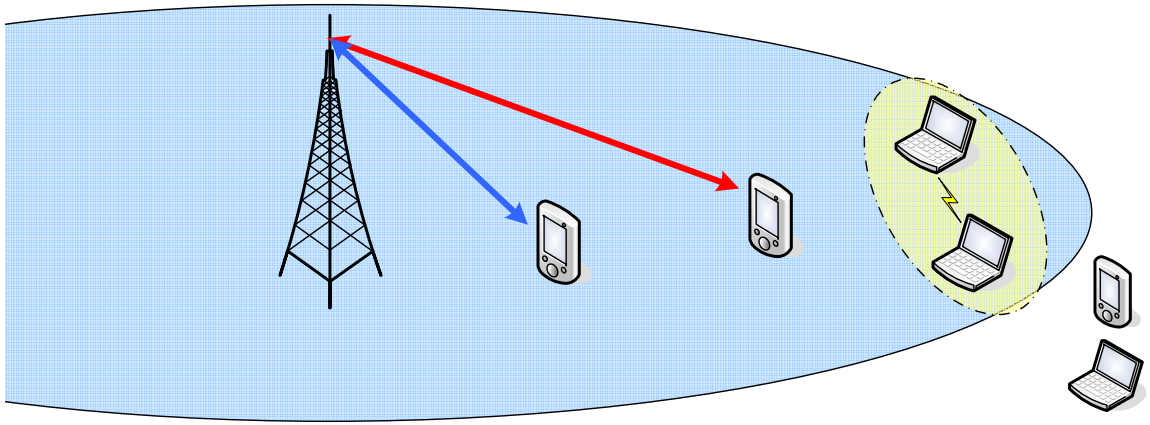
- Operating frequency band: 1.3 GHz
- Bandwidth: 8 MHz
- Modulation: DBPSK
- Bit-rate: 300 kb/sec
- Primary network
  - Cellular network: 1 BS, 2 Primary user
  - Polling MAC
- Secondary network
  - Uplink bandwidth sharing
  - Underlay communication & Overlay communication



Implementation of CR Network

Demo Topology

- Band sharing illustration



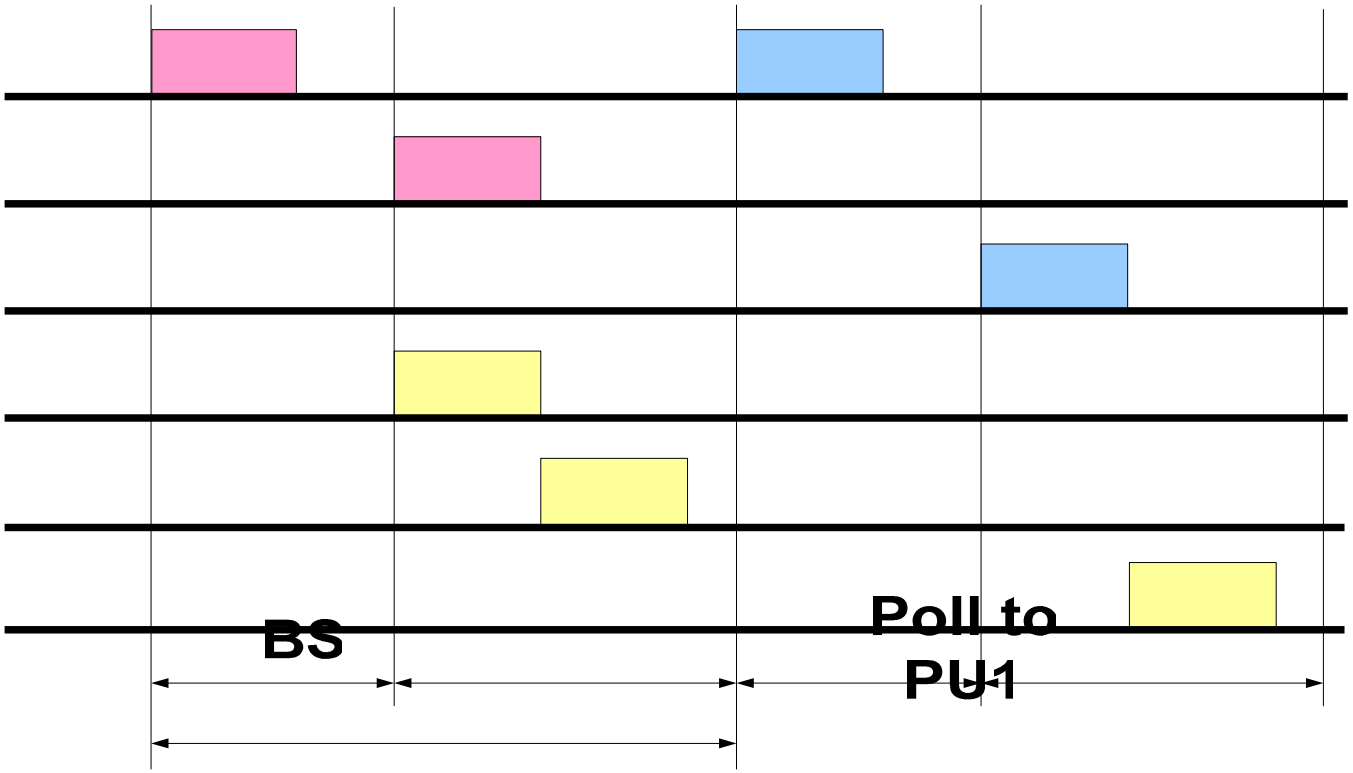
- Implementation

Primary Network

Implementation of CR Network

Demo Scenario

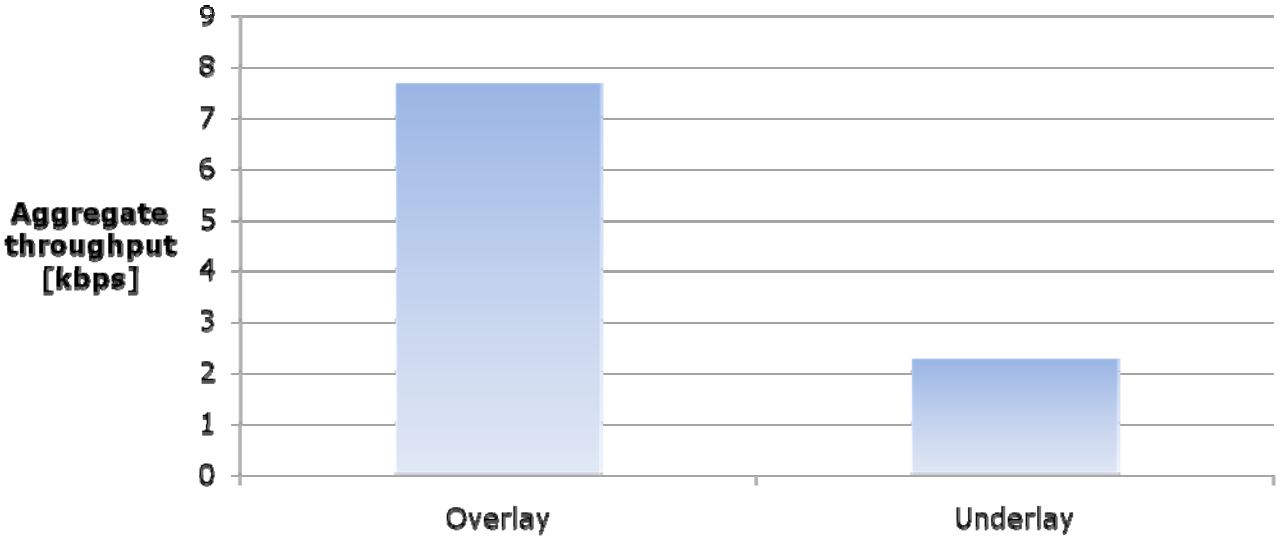
- Primary traffic pattern: CBR (8 kbps for each PU, 16 kbps for BS)
- Secondary users: Using ARQ for error recovery



Implementation of CR Network

Demo Results

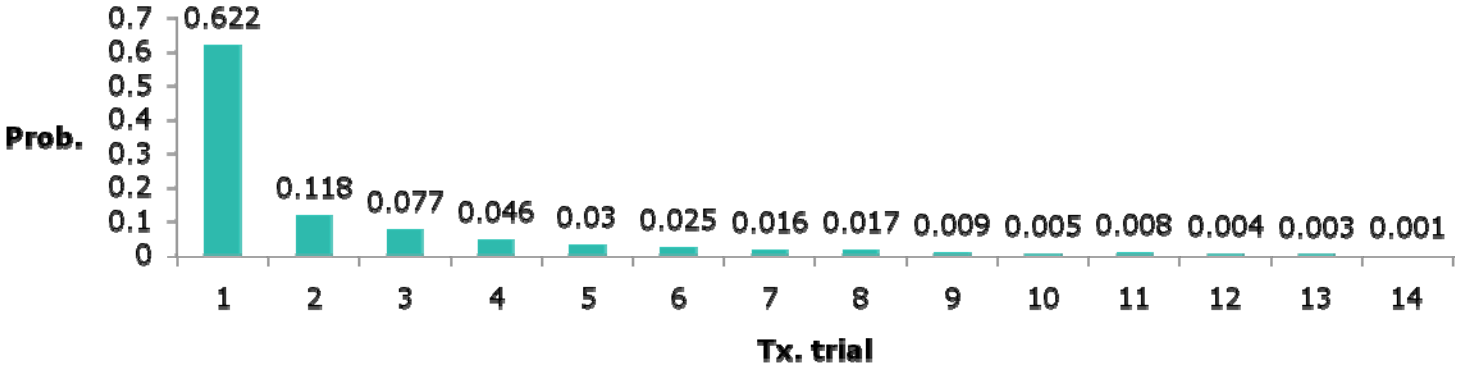
- Aggregate throughput of primary users and CR users
  - Primary network: 32 kbps
  - Secondary
    - Overlay communication: 7.67 kbps
    - Underlay communication: 2.28 kbps



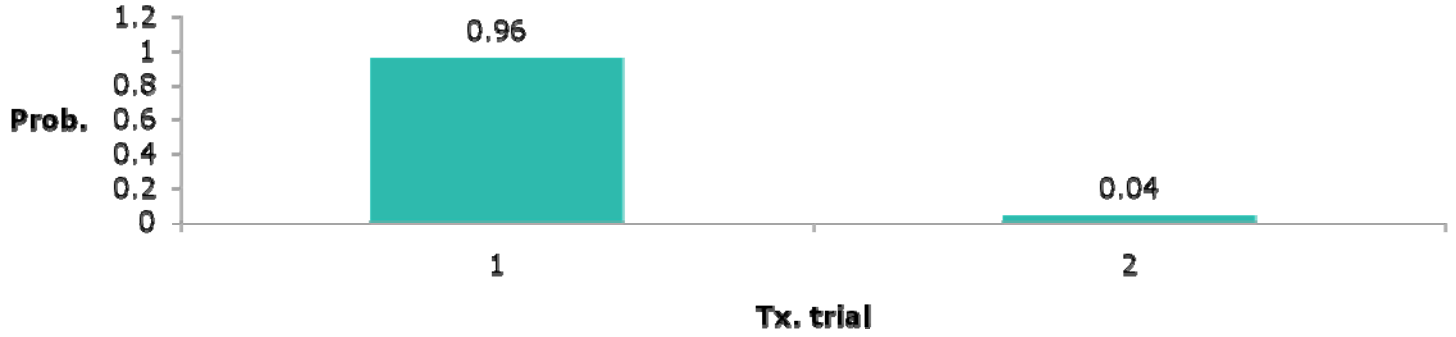
Implementation of CR Network

Demo Results

- Transmission trial of secondary user
  - PDF of transmission trial
    - Underlay communication: Average 2.56 trial/successful Tx.

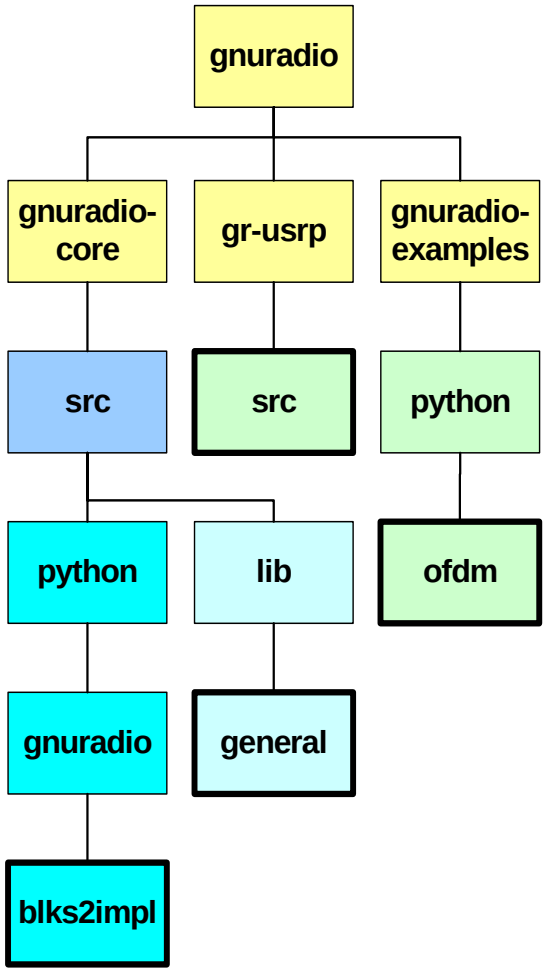


    - Overlay communication: Average 1.04 trial/successful Tx.



Implementation of OFDM

- **OFDM**
  - Parameters
    - Packet size
    - Operating frequency band
    - Transmit power
    - Modulation
    - FPGA interpolation/decimation rate
    - FFT length
    - Occupied tones
    - CP length



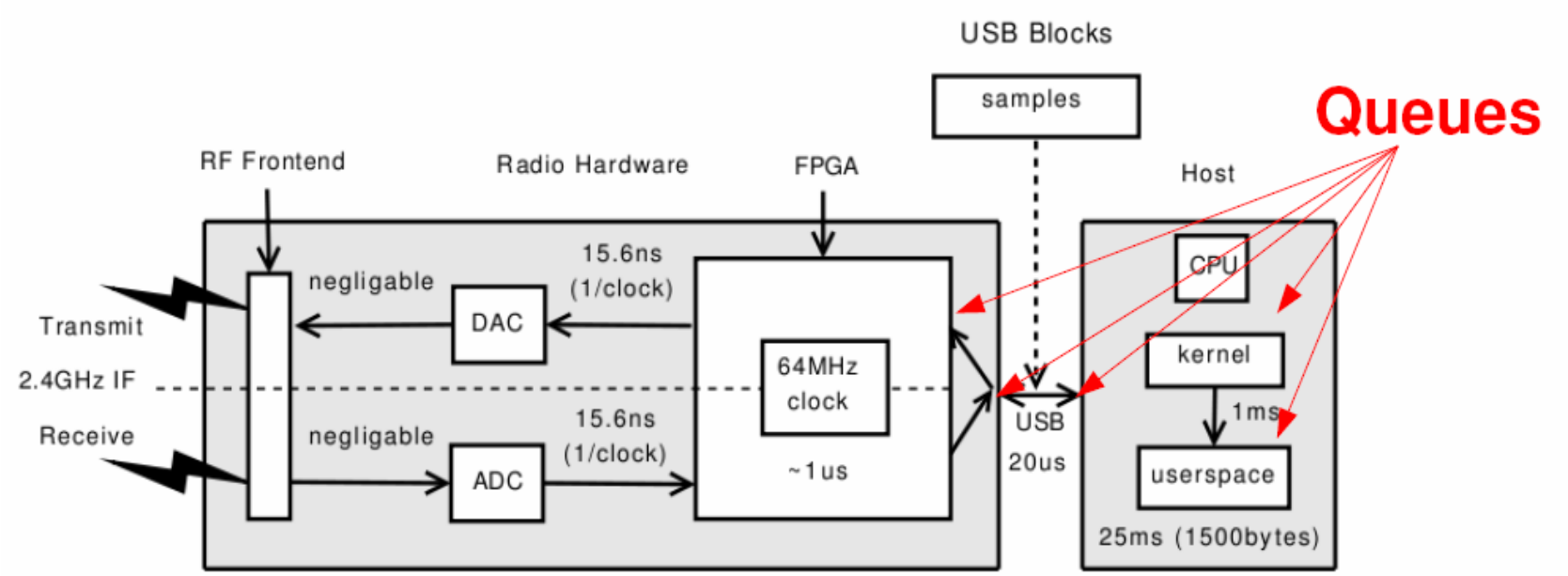
Implementation of OFDM

- **OFDM**
  - Measurements of available operating point
    - Interpolation rate: 128~512
    - Decimation rate: 64~256
    - FFT length: ~1024
    - Occupied tones: ~800
  - Maximum throughput
    - 1024 FFT with 800 occupied tones
    - 128 interpolation
    - Modulation: QPSK

    ➔ ~1.28 Mbps

Limitation of GNU Radio

- **Long & Unpredictable Processing Delay**
  - It may cause critical problems in
    - MAC protocol which requires fine time synchronization
    - Dynamic control using feedback information
    - Time varying environments



Limitation of GNU Radio

- **Unknown Tx. Power**
  - We know digital amplitude at ADC/DAC only
    - 2 bytes value (0~32000)
  - Actual Tx. power depends on USRP, daughter board, antenna, and host computer
    - To know real Tx. power, we should measure Tx. power at the RF frontend
- **Insufficiency of packet-based concept**
  - Message blocks (M-blocks)
    - GNU Radio extension that allows more natural handling of a packet-based data
    - However, GNU radio does not fully support this yet
      - It will be new feature of release-3.2

Concluding Remark

Conclusion

- GNU Radio gives us a lot of flexibility to implementing SDR
  - OFDM, MIMO
  - Flexible MAC
  - Hydra
    - 802.11 like system with time scaling (slot time = 10 ms)
- However, GNU Radio has some limitations
  - Long and unpredictable delay
  - Uncertain operation of USRP
  - Insufficiency of packet-based concept
- For more information, please refer to
  - GNU Radio homepage: http://gnuradio.org
  - GNU Radio forum: http://www.nabble.com/GnuRadio-f1878.html


Thank You !
Any Questions?