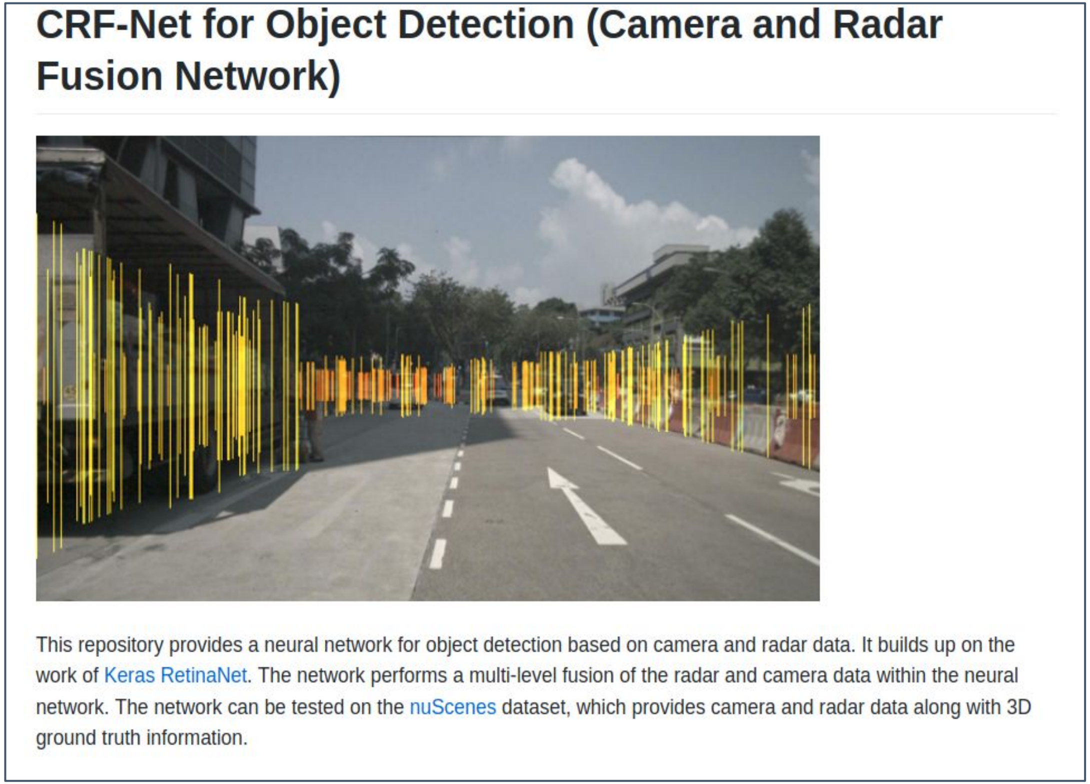
CT-Wall: Perception in through wall scenarios

Shivam Joshi

(sj3104@nyu.edu)

Robot Perception Project

How it started...

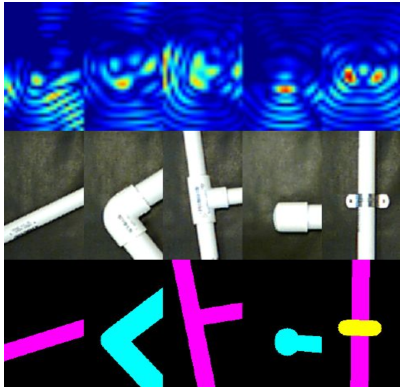


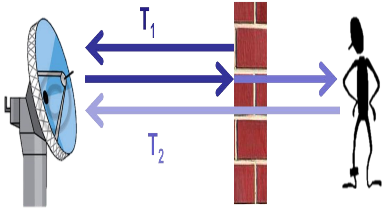
CRF-Net for Object Detection (Camera and Radar Fusion Network)

This repository provides a neural network for object detection based on camera and radar data. It builds up on the work of Keras RetinaNet. The network performs a multi-level fusion of the radar and camera data within the neural network. The network can be tested on the nuScenes dataset, which provides camera and radar data along with 3D ground truth information.

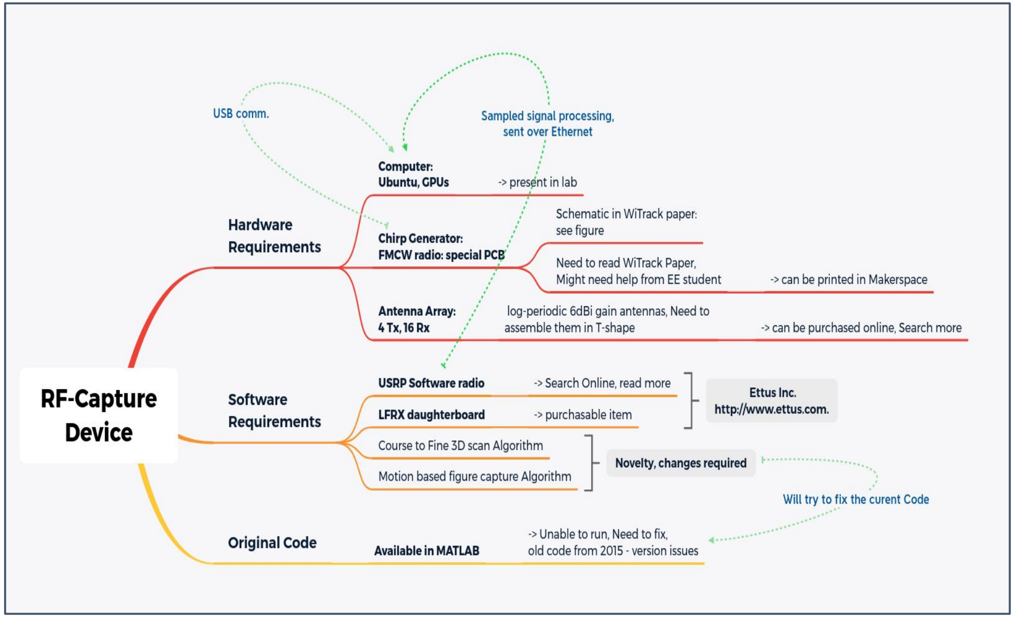
CRF-Net Paper..

And the CT-WALL project



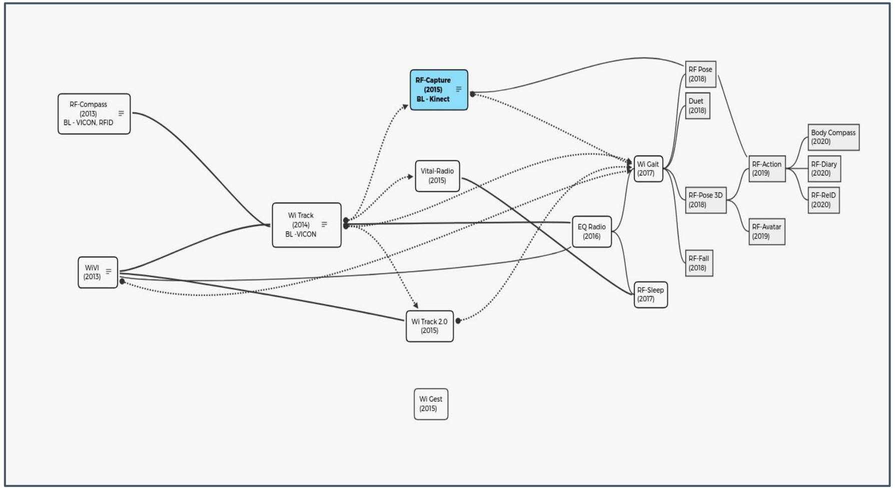


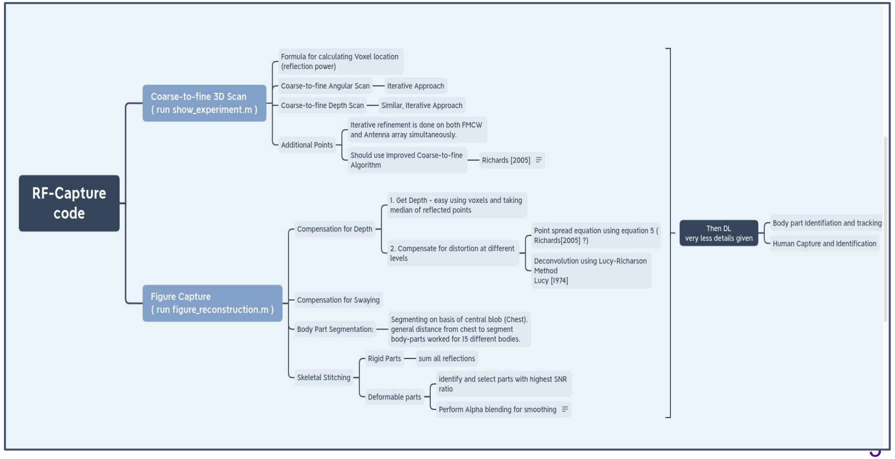
RF-Capture potential



- RF-Capture Device
  - Hardware Requirements
    - Computer: Ubuntu, GPUs -> present in lab
    - Chirp Generator: FMCW radio: special PCB
      - Schematic in WiTrack paper: see figure
      - Need to read WiTrack Paper, Might need help from EE student -> can be printed in Makerspace
    - Antenna Array: 4 Tx, 16 Rx — log-periodic 6dBi gain antennas, Need to assemble them in T-shape -> can be purchased online, Search more
  - Software Requirements
    - USRP Software radio -> Search Online, read more
    - LFRX daughterboard -> purchasable item
      - Ettus Inc. http://www.ettus.com.
    - Course to Fine 3D scan Algorithm
    - Motion based figure capture Algorithm
      - Novelty, changes required
  - Original Code
    - Available in MATLAB -> Unable to run, Need to fix, old code from 2015 - version issues

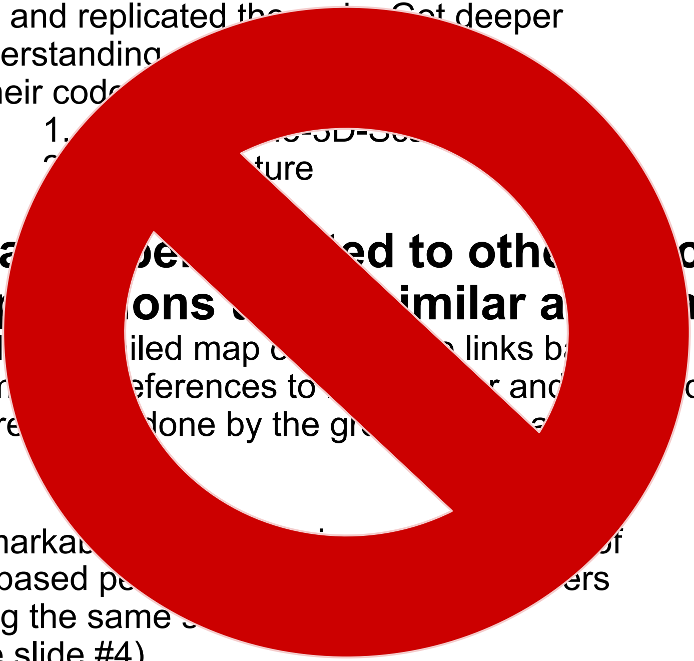
USB comm.
Sampled signal processing, sent over Ethernet
Will try to fix the curent Code



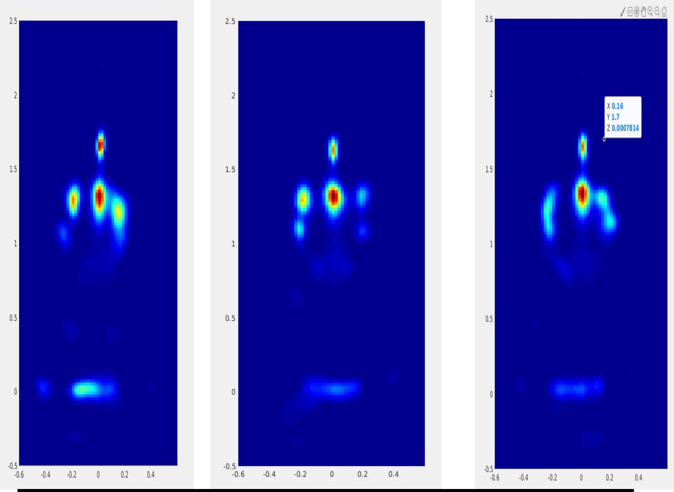


- RF-Capture code
  - Coarse-to-fine 3D Scan (run show_experiment.m)
    - Formula for calculating Voxel location (reflection power)
    - Coarse-to-fine Angular Scan — Iterative Approach
    - Coarse-to-fine Depth Scan — Similar, Iterative Approach
    - Additional Points
      - Iterative refinement is done on both FMCW and Antenna array simultaneously.
      - Should use Improved Coarse-to-fine Algorithm — Richards [2005]
  - Figure Capture (run figure_reconstruction.m)
    - Compensation for Depth
      - 1. Get Depth - easy using voxels and taking median of reflected points
      - 2. Compensate for distortion at different levels
        - Point spread equation using equation 5 (Richards[2005] ?)
        - Deconvolution using Lucy-Richarson Method Lucy [1974]
    - Compensation for Swaying
    - Body Part Segmentation: — Segmenting on basis of central blob (Chest). general distance from chest to segment body-parts worked for 15 different bodies.
    - Skeletal Stitching
      - Rigid Parts — sum all reflections
      - Deformable parts
        - identify and select parts with highest SNR ratio
        - Perform Alpha blending for smoothing
  - Then DL very less details given
    - Body part Identifiation and tracking
    - Human Capture and Identification

Trial and errors



1. **Ran and understood RF-Capture**
   - Ran and replicated the ... Get deeper Understanding ... of their code
     1. ...-3D-Sc...
     2. ...ture
   -
2. **Rea... ...ed to othe... ...ception app... ...ons ... ...imilar a... ...ch**
   - Mad... ...iled map ... links ba... on m... ...eferences to ... r and ... d the re... done by the gr... a...
   - Remarkab... ...of RF-based pe... ...ers using the same ... (see slide #4)

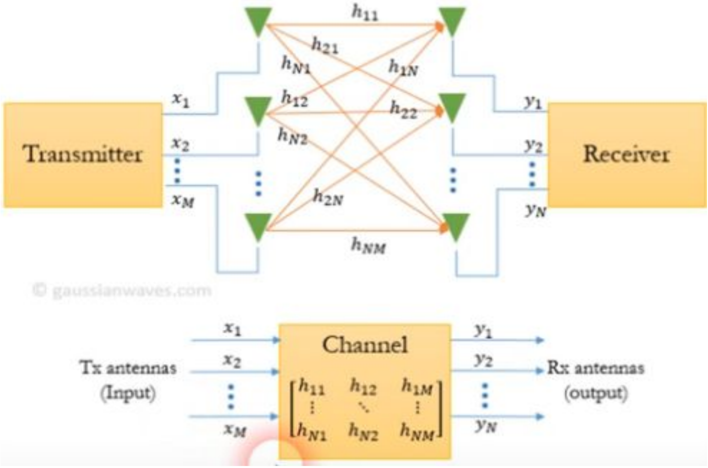


Look into commercial WiFi - CSI

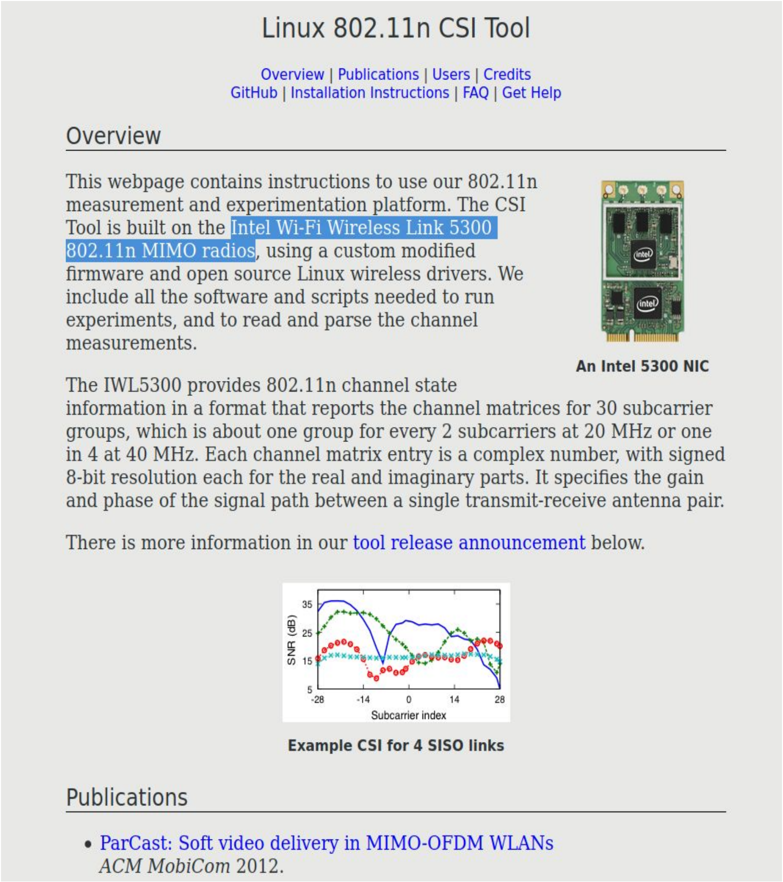
Channel State Information:

1. Refers to channel properties in wireless communication.

2. Describes how signal propagates from Tx to Rx. ie. gives information about media b/w Tx and Rx.



$$Y(f,t) = H(f,t) \otimes X(f,t) + N$$



Linux 802.11n CSI Tool

Overview | Publications | Users | Credits
GitHub | Installation Instructions | FAQ | Get Help

Overview

This webpage contains instructions to use our 802.11n measurement and experimentation platform. The CSI Tool is built on the Intel Wi-Fi Wireless Link 5300 802.11n MIMO radios, using a custom modified firmware and open source Linux wireless drivers. We include all the software and scripts needed to run experiments, and to read and parse the channel measurements.

An Intel 5300 NIC

The IWL5300 provides 802.11n channel state information in a format that reports the channel matrices for 30 subcarrier groups, which is about one group for every 2 subcarriers at 20 MHz or one in 4 at 40 MHz. Each channel matrix entry is a complex number, with signed 8-bit resolution each for the real and imaginary parts. It specifies the gain and phase of the signal path between a single transmit-receive antenna pair.

There is more information in our tool release announcement below.

Example CSI for 4 SISO links

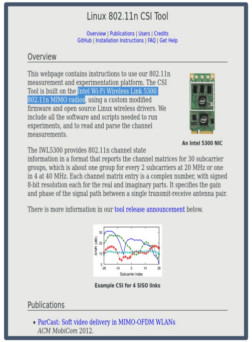
Publications

- ParCast: Soft video delivery in MIMO-OFDM WLANs
  ACM MobiCom 2012.

Link: https://github.com/dhalperi/linux-80211n-csitool/

With Assistance - Prof Wang (MERL)

Get started with information about Wifi 802.11ac → specifics of the router, open-source manuscripts for tweaking those routers and getting rich CSI.



Linux 802.11n CSI Tool

Overview | Publications | Users | Credits
GitHub | Installation Instructions | FAQ | Get Help

Overview

This webpage contains instructions to use our 802.11n measurement and experimentation platform. The CSI Tool is built on the Intel Wi-Fi Wireless Link 5300 802.11n MIMO radios, using a custom modified firmware and open source Linux wireless drivers. We include all the software and scripts needed to run experiments, and to read and parse the channel measurements.

An Intel 5300 NIC

The IWL5300 provides 802.11n channel state information in a format that reports the channel matrices for 30 subcarrier groups, which is about one group for every 2 subcarriers at 20 MHz or one in 4 at 40 MHz. Each channel matrix entry is a complex number, with signed 8-bit resolution each for the real and imaginary parts. It specifies the gain and phase of the signal path between a single transmit-receive antenna pair.

There is more information in our tool release announcement below.

Example CSI for 4 SISO links

Publications

- ParCast: Soft video delivery in MIMO-OFDM WLANs
  ACM MobiCom 2012.

Link:
https://github.com/seemoo-lab/nexmon_csi#getting-started

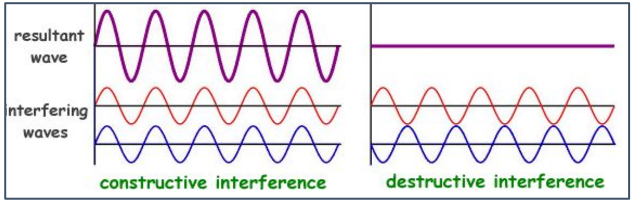
nexmon

Nexmon Channel State Information Extractor

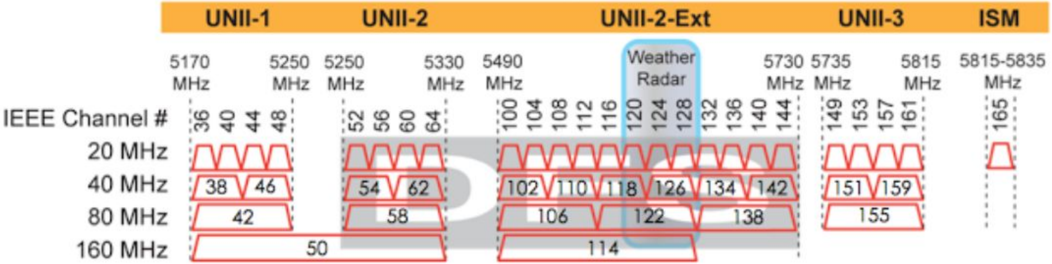
Progress

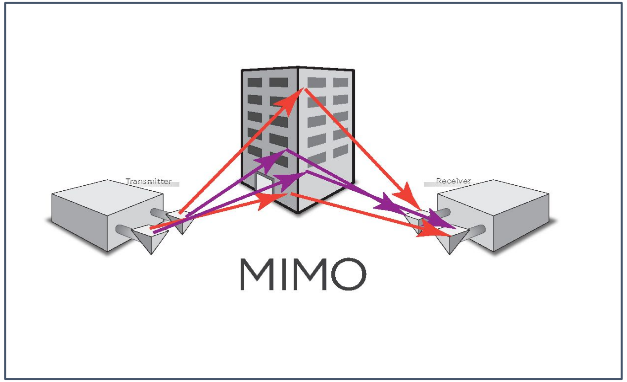
Understand CSI data well, explain next time to peers

- **RF Waves** - wavelength, amplitude, frequency, phase - interference
- **Wave properties** - attenuation, amplification, *reflection*, refraction, *absorption*, scattering, diffraction. (dependent on freq, power)
- **RF Measurements and Math** - mW and dBm (+3dB == 2x power)
- **MIMO** and **OFDM** - WiFi 802.11n/ac
- **Channels** and **Medium**

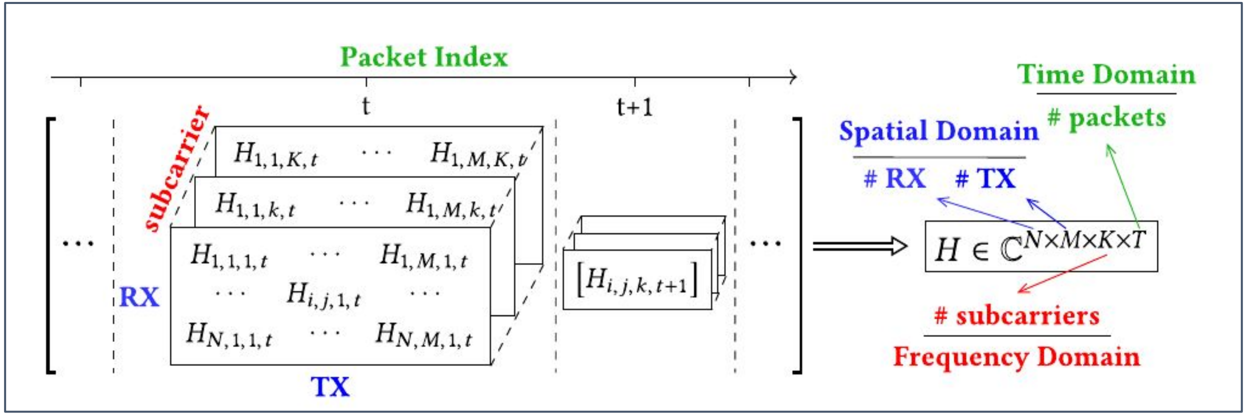


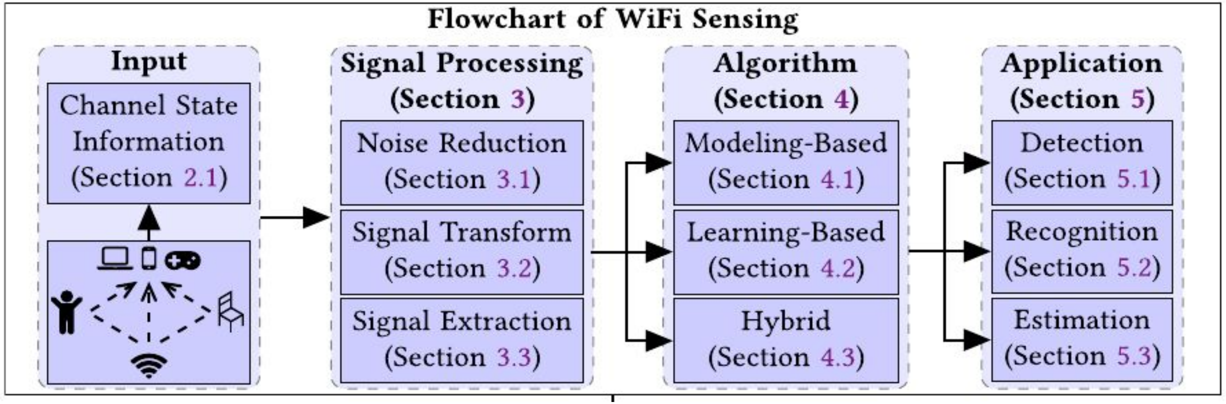
So here is a quick reference guide to referencing the 5 GHz Wi-Fi channels.





How to use CSI for our project?


Packet Index
t
t+1
subcarrier
$H_{1,1,K,t}$ ⋯ $H_{1,M,K,t}$
$H_{1,1,k,t}$ ⋯ $H_{1,M,k,t}$
$H_{1,1,1,t}$ ⋯ $H_{1,M,1,t}$
⋯ $H_{i,j,1,t}$ ⋯
$H_{N,1,1,t}$ ⋯ $H_{N,M,1,t}$
RX
TX
$[H_{i,j,k,t+1}]$
$H \in \mathbb{C}^{N \times M \times K \times T}$
Time Domain
# packets
Spatial Domain
# RX
# TX
# subcarriers
Frequency Domain



Flowchart of WiFi Sensing

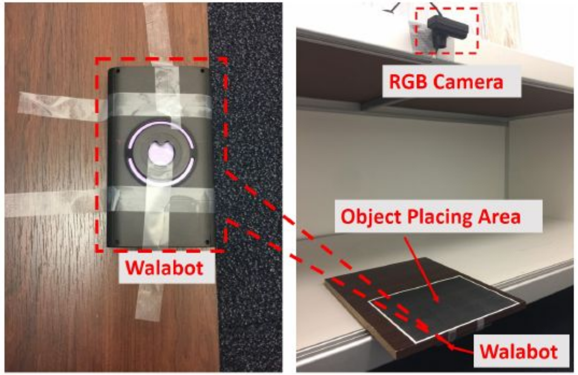
Input
- Channel State Information (Section 2.1)

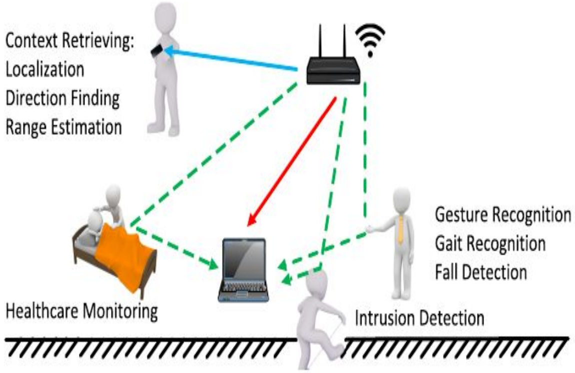
Signal Processing (Section 3)
- Noise Reduction (Section 3.1)
- Signal Transform (Section 3.2)
- Signal Extraction (Section 3.3)

Algorithm (Section 4)
- Modeling-Based (Section 4.1)
- Learning-Based (Section 4.2)
- Hybrid (Section 4.3)

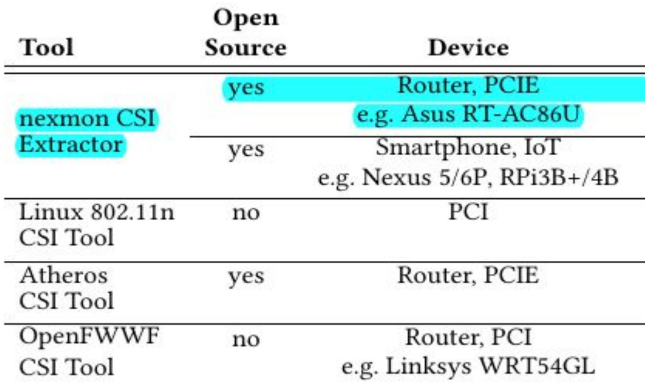
Application (Section 5)
- Detection (Section 5.1)
- Recognition (Section 5.2)
- Estimation (Section 5.3)

Link: http://www.cs.wm.edu/~yma/files/WiFiSensing_YongsenMa_authorversion.pdf

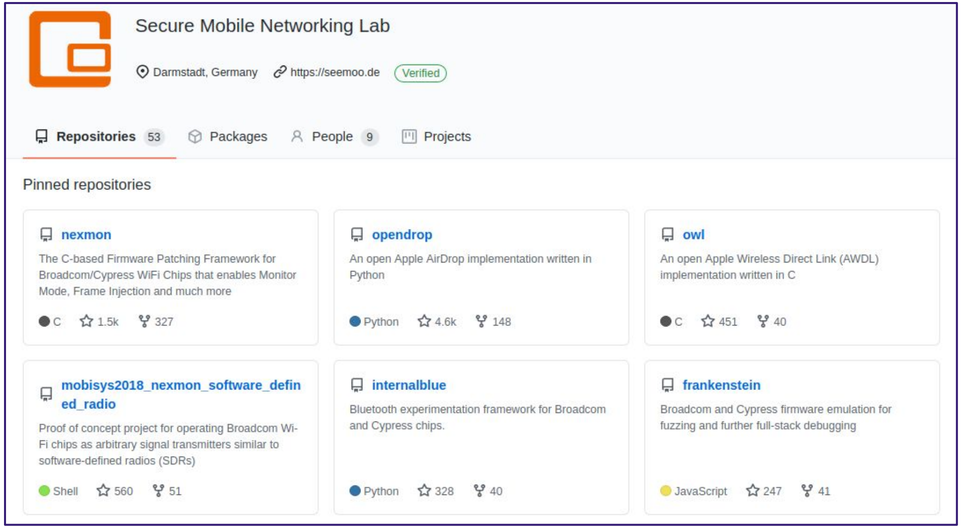




What is nexmon?



Tool	Open Source	Device
nexmon CSI Extractor	yes	Router, PCIE e.g. Asus RT-AC86U
	yes	Smartphone, IoT e.g. Nexus 5/6P, RPi3B+/4B
Linux 802.11n CSI Tool	no	PCI
Atheros CSI Tool	yes	Router, PCIE
OpenFWWF CSI Tool	no	Router, PCI e.g. Linksys WRT54GL



Secure Mobile Networking Lab

Darmstadt, Germany https://seemoo.de Verified

Pinned repositories

nexmon
The C-based Firmware Patching Framework for Broadcom/Cypress WiFi Chips that enables Monitor Mode, Frame Injection and much more
C ☆ 1.5k 327

opendrop
An open Apple AirDrop implementation written in Python
Python ☆ 4.6k 148

owl
An open Apple Wireless Direct Link (AWDL) implementation written in C
C ☆ 451 40

mobisys2018_nexmon_software_defined_radio
Proof of concept project for operating Broadcom Wi-Fi chips as arbitrary signal transmitters similar to software-defined radios (SDRs)
Shell ☆ 560 51

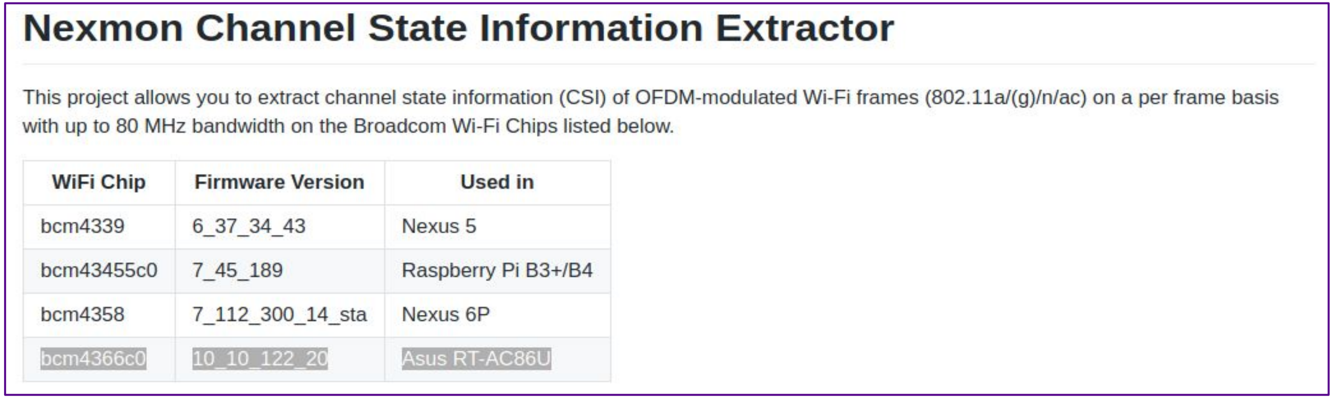
internalblue
Bluetooth experimentation framework for Broadcom and Cypress chips.
Python ☆ 328 40

frankenstein
Broadcom and Cypress firmware emulation for fuzzing and further full-stack debugging
JavaScript ☆ 247 41

Contact

Prof. Dr.-Ing. Matthias Hollick

Secure Mobile Networking Lab
Department of Computer Science
Technische Universität Darmstadt



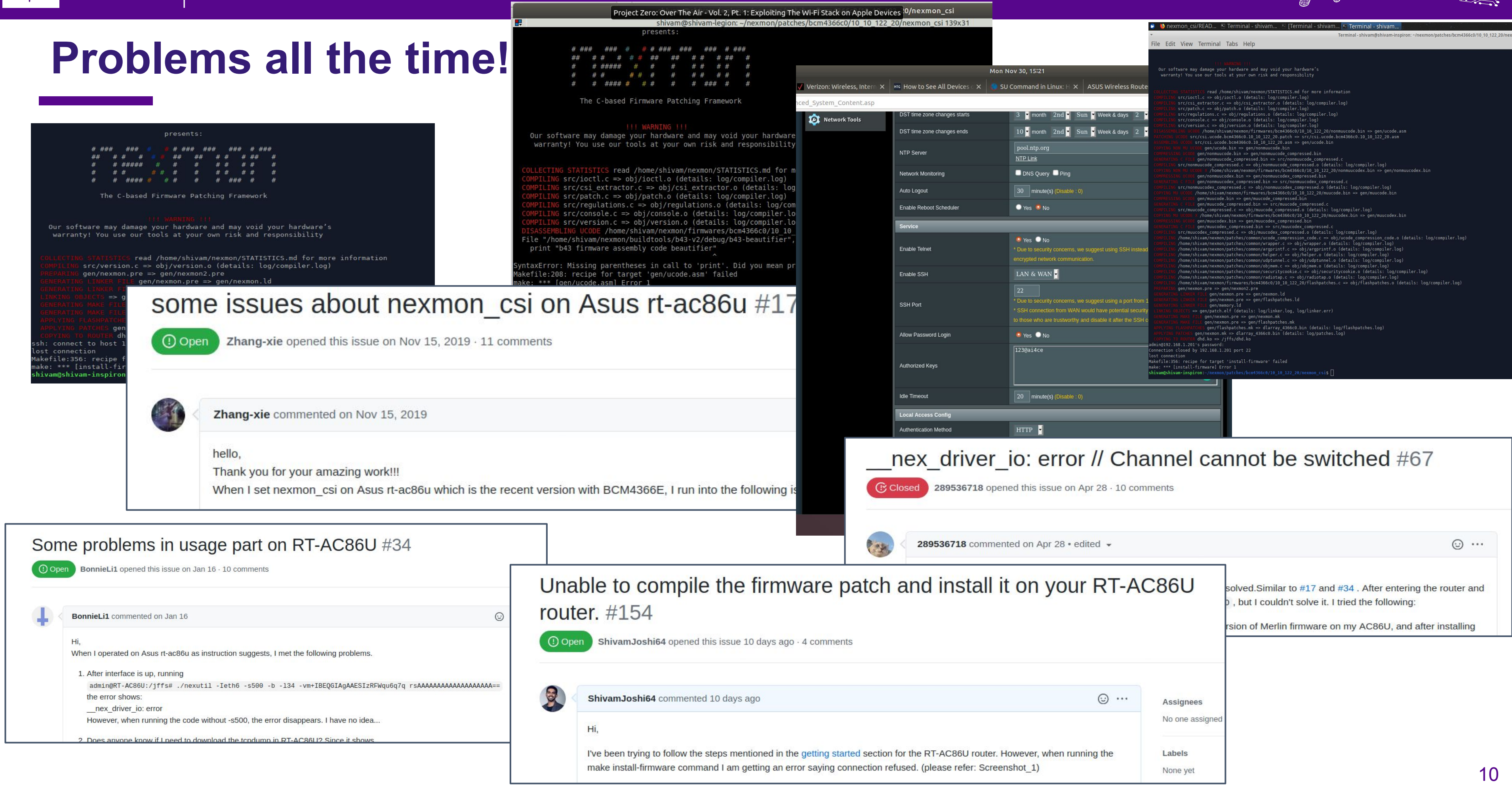
Nexmon Channel State Information Extractor

This project allows you to extract channel state information (CSI) of OFDM-modulated Wi-Fi frames (802.11a/(g)/n/ac) on a per frame basis with up to 80 MHz bandwidth on the Broadcom Wi-Fi Chips listed below.

WiFi Chip	Firmware Version	Used in
bcm4339	6_37_34_43	Nexus 5
bcm43455c0	7_45_189	Raspberry Pi B3+/B4
bcm4358	7_112_300_14_sta	Nexus 6P
bcm4366c0	10_10_122_20	Asus RT-AC86U

Link: Seemo

Problems all the time!



```
presents:

 # ###   ###  #    # # ###  ###   ###  # ###
 ##   # #   #  # #  ##   ##   # #   # ##   #
 #    # #####   #   #    #    # #   # #    #
 #    # #      # #  #    #    # #   # #    #
 #    #  #### #   # #    #    #  ###  #    #

   The C-based Firmware Patching Framework


              !!! WARNING !!!
 Our software may damage your hardware and may void your hardware's
  warranty! You use our tools at your own risk and responsibility


 COLLECTING STATISTICS read /home/shivam/nexmon/STATISTICS.md for more information
 COMPILING src/version.c => obj/version.o (details: log/compiler.log)
 PREPARING gen/nexmon.pre => gen/nexmon2.pre
 GENERATING LINKER FILE gen/nexmon.pre => gen/nexmon.ld
 GENERATING LINKER FI
 LINKING OBJECTS => g
 GENERATING MAKE FILE
 GENERATING MAKE FILE
 APPLYING FLASHPATCHE
 APPLYING PATCHES gen
 COPYING TO ROUTER dh
ssh: connect to host 1
lost connection
Makefile:356: recipe f
make: *** [install-fir
shivam@shivam-inspiron
```

Project Zero: Over The Air - Vol. 2, Pt. 1: Exploiting The Wi-Fi Stack on Apple Devices

0/nexmon_csi

shivam@shivam-legion: ~/nexmon/patches/bcm4366c0/10_10_122_20/nexmon_csi 139x31

```
                 presents:

 # ###   ###  #    # # ###  ###   ###  # ###
 ##   # #   #  # #  ##   ##   # #   # ##   #
 #    # #####   #   #    #    # #   # #    #
 #    # #      # #  #    #    # #   # #    #
 #    #  #### #   # #    #    #  ###  #    #

   The C-based Firmware Patching Framework


              !!! WARNING !!!
 Our software may damage your hardware and may void your hardware
  warranty! You use our tools at your own risk and responsibility


 COLLECTING STATISTICS read /home/shivam/nexmon/STATISTICS.md for m
 COMPILING src/ioctl.c => obj/ioctl.o (details: log/compiler.log)
 COMPILING src/csi_extractor.c => obj/csi_extractor.o (details: log
 COMPILING src/patch.c => obj/patch.o (details: log/compiler.log)
 COMPILING src/regulations.c => obj/regulations.o (details: log/com
 COMPILING src/console.c => obj/console.o (details: log/compiler.lo
 COMPILING src/version.c => obj/version.o (details: log/compiler.lo
 DISASSEMBLING UCODE /home/shivam/nexmon/firmwares/bcm4366c0/10_10_
 File "/home/shivam/nexmon/buildtools/b43-v2/debug/b43-beautifier",
   print "b43 firmware assembly code beautifier"
                                               ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean pr
Makefile:208: recipe for target 'gen/ucode.asm' failed
make: *** [gen/ucode.asm] Error 1
```

Mon Nov 30, 15:21

Verizon: Wireless, Intern ✕ | How to See All Devices ✕ | SU Command in Linux: H ✕ | ASUS Wireless Route

nced_System_Content.asp

Network Tools

DST time zone changes starts	3 month 2nd Sun Week & days 2
DST time zone changes ends	10 month 2nd Sun Week & days 2
NTP Server	pool.ntp.org NTP Link
Network Monitoring	□ DNS Query □ Ping
Auto Logout	30 minute(s) (Disable : 0)
Enable Reboot Scheduler	○ Yes ● No
Service	
Enable Telnet	● Yes ○ No * Due to security concerns, we suggest using SSH instead encrypted network communication.
Enable SSH	LAN & WAN
SSH Port	22 * Due to security concerns, we suggest using a port from 1 * SSH connection from WAN would have potential security to those who are trustworthy and disable it after the SSH c
Allow Password Login	● Yes ○ No
Authorized Keys	123@ai4ce
Idle Timeout	20 minute(s) (Disable : 0)
Local Access Config	
Authentication Method	HTTP

nexmon_csi/READ... | Terminal - shivam... | [Terminal - shivam... | Terminal - shivam...

Terminal - shivam@shivam-inspiron: ~/nexmon/patches/bcm4366c0/10_10_122_20/nex

File Edit View Terminal Tabs Help

```
                  !!! WARNING !!!
  Our software may damage your hardware and may void your hardware's
   warranty! You use our tools at your own risk and responsibility


 COLLECTING STATISTICS read /home/shivam/nexmon/STATISTICS.md for more information
 COMPILING src/ioctl.c => obj/ioctl.o (details: log/compiler.log)
 COMPILING src/csi_extractor.c => obj/csi_extractor.o (details: log/compiler.log)
 COMPILING src/patch.c => obj/patch.o (details: log/compiler.log)
 COMPILING src/regulations.c => obj/regulations.o (details: log/compiler.log)
 COMPILING src/console.c => obj/console.o (details: log/compiler.log)
 COMPILING src/version.c => obj/version.o (details: log/compiler.log)
 DISASSEMBLING UCODE /home/shivam/nexmon/firmwares/bcm4366c0/10_10_122_20/nonmuucode.bin => gen/ucode.asm
 PATCHING UCODE src/csi.ucode.bcm4366c0.10_10_122_20.patch => src/csi.ucode.bcm4366c0.10_10_122_20.asm
 ASSEMBLING UCODE src/csi.ucode.bcm4366c0.10_10_122_20.asm => gen/ucode.bin
 COPYING NON MU UCODE gen/ucode.bin => gen/nonmuucode.bin
 COMPRESSING UCODE gen/nonmuucode.bin => gen/nonmuucode_compressed.bin
 GENERATING C FILE gen/nonmuucode_compressed.bin => src/nonmuucode_compressed.c
 COMPILING src/nonmuucode_compressed.c => obj/nonmuucode_compressed.o (details: log/compiler.log)
 COPYING NON MU UCODE X /home/shivam/nexmon/firmwares/bcm4366c0/10_10_122_20/nonmuucodex.bin => gen/nonmuucodex.bin
 COMPRESSING UCODE gen/nonmuucodex.bin => gen/nonmuucodex_compressed.bin
 GENERATING C FILE gen/nonmuucodex_compressed.bin => src/nonmuucodex_compressed.c
 COMPILING src/nonmuucodex_compressed.c => obj/nonmuucodex_compressed.o (details: log/compiler.log)
 COPYING MU UCODE /home/shivam/nexmon/firmwares/bcm4366c0/10_10_122_20/muucode.bin => gen/muucode.bin
 COMPRESSING UCODE gen/muucode.bin => gen/muucode_compressed.bin
 GENERATING C FILE gen/muucode_compressed.bin => src/muucode_compressed.c
 COMPILING src/muucode_compressed.c => obj/muucode_compressed.o (details: log/compiler.log)
 COPYING MU UCODE X /home/shivam/nexmon/firmwares/bcm4366c0/10_10_122_20/muucodex.bin => gen/muucodex.bin
 COMPRESSING UCODE gen/muucodex.bin => gen/muucodex_compressed.bin
 GENERATING C FILE gen/muucodex_compressed.bin => src/muucodex_compressed.c
 COMPILING src/muucodex_compressed.c => obj/muucodex_compressed.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/patches/common/ucode_compression_code.c => obj/ucode_compression_code.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/patches/common/wrapper.c => obj/wrapper.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/patches/common/argprintf.c => obj/argprintf.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/patches/common/helper.c => obj/helper.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/patches/common/udptunnel.c => obj/udptunnel.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/patches/common/objmem.c => obj/objmem.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/patches/common/securitycookie.c => obj/securitycookie.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/patches/common/radiotap.c => obj/radiotap.o (details: log/compiler.log)
 COMPILING /home/shivam/nexmon/firmwares/bcm4366c0/10_10_122_20/flashpatches.c => obj/flashpatches.o (details: log/compiler.log)
 PREPARING gen/nexmon.pre => gen/nexmon2.pre
 GENERATING LINKER FILE gen/nexmon.pre => gen/nexmon.ld
 GENERATING LINKER FILE gen/nexmon.pre => gen/flashpatches.ld
 GENERATING LINKER FILE gen/memory.ld
 LINKING OBJECTS => gen/patch.elf (details: log/linker.log, log/linker.err)
 GENERATING MAKE FILE gen/nexmon.pre => gen/nexmon.mk
 GENERATING MAKE FILE gen/nexmon.pre => gen/flashpatches.mk
 APPLYING FLASHPATCHES gen/flashpatches.mk => dlarray_4366c0.bin (details: log/flashpatches.log)
 APPLYING PATCHES gen/nexmon.mk => dlarray_4366c0.bin (details: log/patches.log)
 COPYING TO ROUTER dhd.ko => /jffs/dhd.ko
admin@192.168.1.201's password:
Connection closed by 192.168.1.201 port 22
lost connection
Makefile:356: recipe for target 'install-firmware' failed
make: *** [install-firmware] Error 1
shivam@shivam-inspiron:~/nexmon/patches/bcm4366c0/10_10_122_20/nexmon_csi$
```

some issues about nexmon_csi on Asus rt-ac86u #17

Open **Zhang-xie** opened this issue on Nov 15, 2019 · 11 comments

Zhang-xie commented on Nov 15, 2019

hello,

Thank you for your amazing work!!!

When I set nexmon_csi on Asus rt-ac86u which is the recent version with BCM4366E, I run into the following is

__nex_driver_io: error // Channel cannot be switched #67

Closed **289536718** opened this issue on Apr 28 · 10 comments

289536718 commented on Apr 28 • edited

solved.Similar to #17 and #34 . After entering the router and

, but I couldn't solve it. I tried the following:

rsion of Merlin firmware on my AC86U, and after installing

Some problems in usage part on RT-AC86U #34

Open **BonnieLi1** opened this issue on Jan 16 · 10 comments

BonnieLi1 commented on Jan 16

Hi,
When I operated on Asus rt-ac86u as instruction suggests, I met the following problems.

1. After interface is up, running
   `admin@RT-AC86U:/jffs# ./nexutil -Ieth6 -s500 -b -l34 -vm+IBEQGIAgAAESIzRFWqu6q7q rsAAAAAAAAAAAAAAAAAA==`
   the error shows:
   __nex_driver_io: error
   However, when running the code without -s500, the error disappears. I have no idea...
2. Does anyone know if I need to download the tcpdump in RT-AC86U? Since it shows

Unable to compile the firmware patch and install it on your RT-AC86U router. #154

Open **ShivamJoshi64** opened this issue 10 days ago · 4 comments

ShivamJoshi64 commented 10 days ago

Hi,

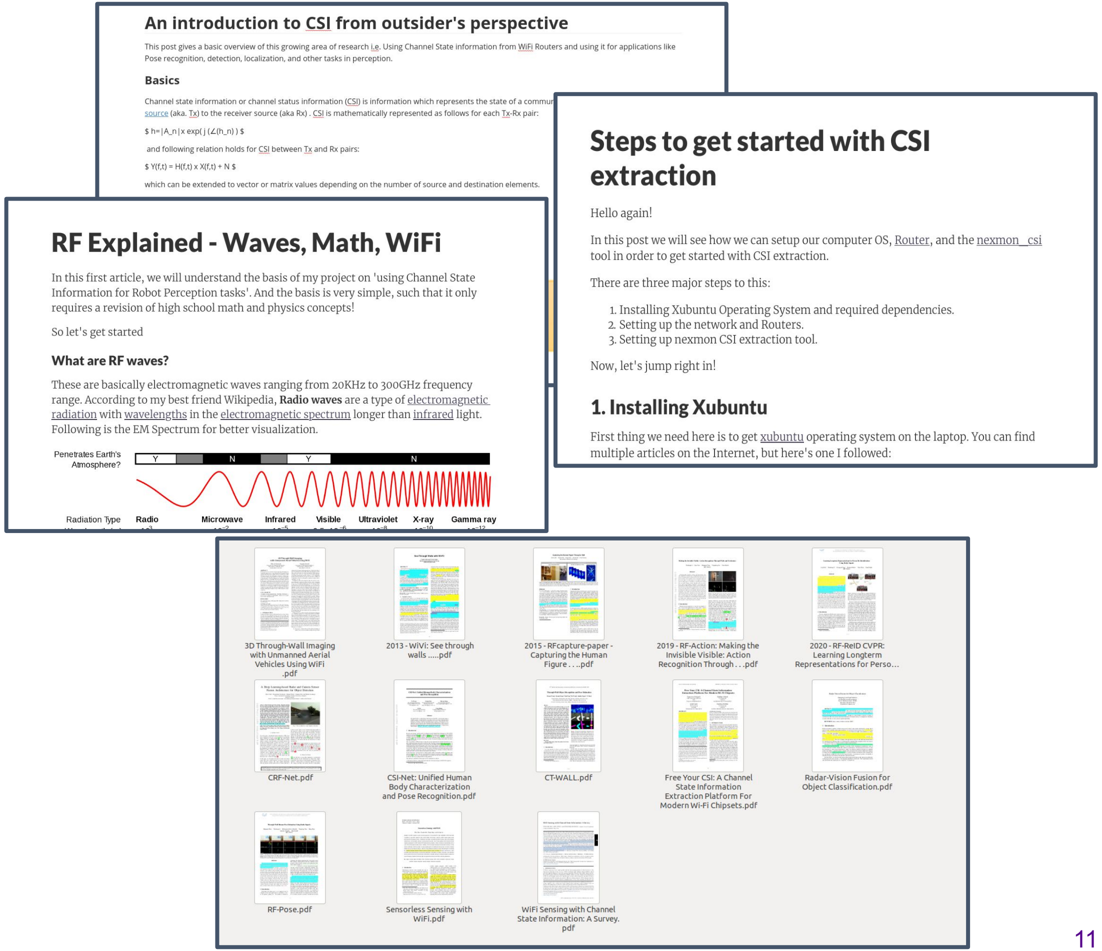
I've been trying to follow the steps mentioned in the getting started section for the RT-AC86U router. However, when running the make install-firmware command I am getting an error saying connection refused. (please refer: Screenshot_1)

Assignees
No one assigned

Labels
None yet

Deliverables Promised

1. Having a working setup
2. Generation of datasets
3. # reviewed papers >= 10
4. Paper summary posts >= 5
5. Successful initial experiments



An introduction to CSI from outsider's perspective

This post gives a basic overview of this growing area of research i.e. Using Channel State information from WiFi Routers and using it for applications like Pose recognition, detection, localization, and other tasks in perception.

Basics

Channel state information or channel status information (CSI) is information which represents the state of a commur source (aka. Tx) to the receiver source (aka Rx) . CSI is mathematically represented as follows for each Tx-Rx pair:

$ h=|A_n|x exp(j (∠(h_n)) $

and following relation holds for CSI between Tx and Rx pairs:

$ Y(f,t) = H(f,t) x X(f,t) + N $

which can be extended to vector or matrix values depending on the number of source and destination elements.

RF Explained - Waves, Math, WiFi

In this first article, we will understand the basis of my project on 'using Channel State Information for Robot Perception tasks'. And the basis is very simple, such that it only requires a revision of high school math and physics concepts!

So let's get started

What are RF waves?

These are basically electromagnetic waves ranging from 20KHz to 300GHz frequency range. According to my best friend Wikipedia, **Radio waves** are a type of electromagnetic radiation with wavelengths in the electromagnetic spectrum longer than infrared light. Following is the EM Spectrum for better visualization.

Steps to get started with CSI extraction

Hello again!

In this post we will see how we can setup our computer OS, Router, and the nexmon_csi tool in order to get started with CSI extraction.

There are three major steps to this:

1. Installing Xubuntu Operating System and required dependencies.
2. Setting up the network and Routers.
3. Setting up nexmon CSI extraction tool.

Now, let's jump right in!

1. Installing Xubuntu

First thing we need here is to get xubuntu operating system on the laptop. You can find multiple articles on the Internet, but here's one I followed:

3D Through-Wall Imaging with Unmanned Aerial Vehicles Using WiFi .pdf

2013 - WiVi: See through walls .....pdf

2015 - RFcapture-paper - Capturing the Human Figure . . . .pdf

2019 - RF-Action: Making the Invisible Visible: Action Recognition Through . . .pdf

2020 - RF-ReID CVPR: Learning Longterm Representations for Perso…

CRF-Net.pdf

CSI-Net: Unified Human Body Characterization and Pose Recognition.pdf

CT-WALL.pdf

Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets.pdf

Radar-Vision Fusion for Object Classification.pdf

RF-Pose.pdf

Sensorless Sensing with WiFi.pdf

WiFi Sensing with Channel State Information: A Survey. pdf