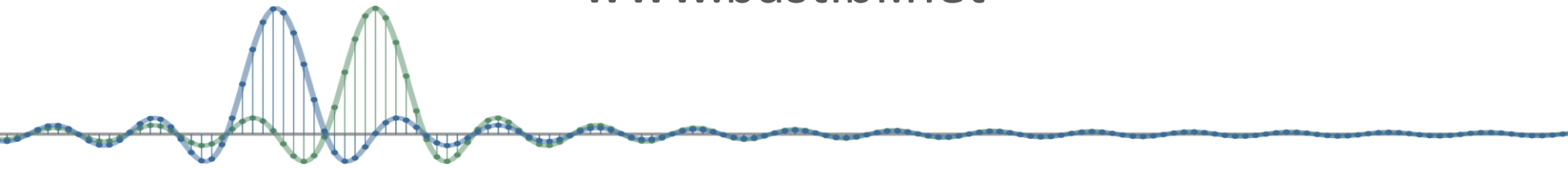
GNU Radio Meets Scapy

mail@bastibl.net

www.bastibl.net

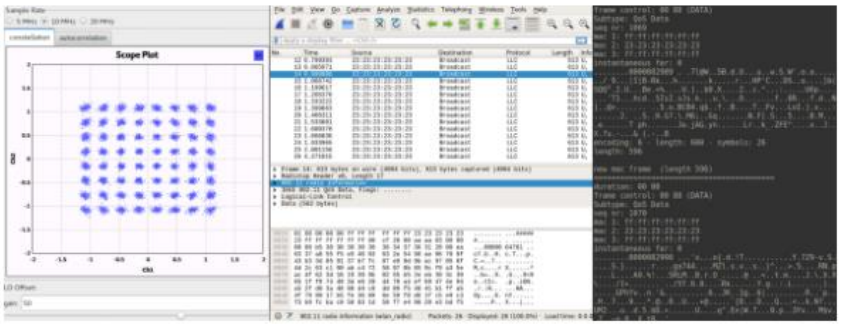


February 2019 ▪ FOSDEM SDR Dev Room ▪ Brussels, Belgium

www.wime-project.net

- **WLAN and ZigBee Transceiver**

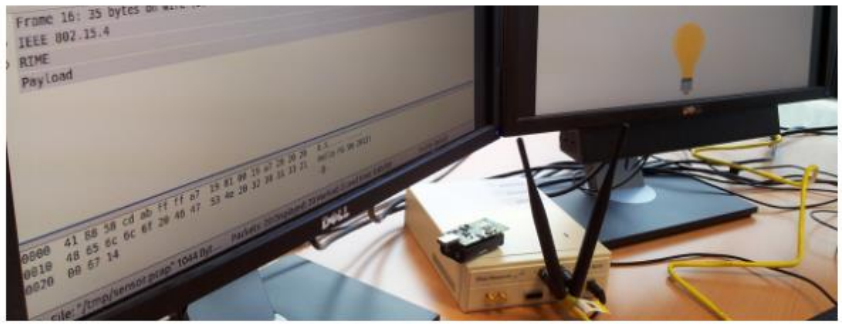
IEEE 802.11a/g/p – WiFi



WiFi networks, which are already ubiquitous today, will soon spread even further by providing the base for inter-vehicular communication systems. To study these networks, we provide a complete physical layer implementation, including a tool chain for simulation and experimentation.

Learn more »

IEEE 802.15.4 – ZigBee



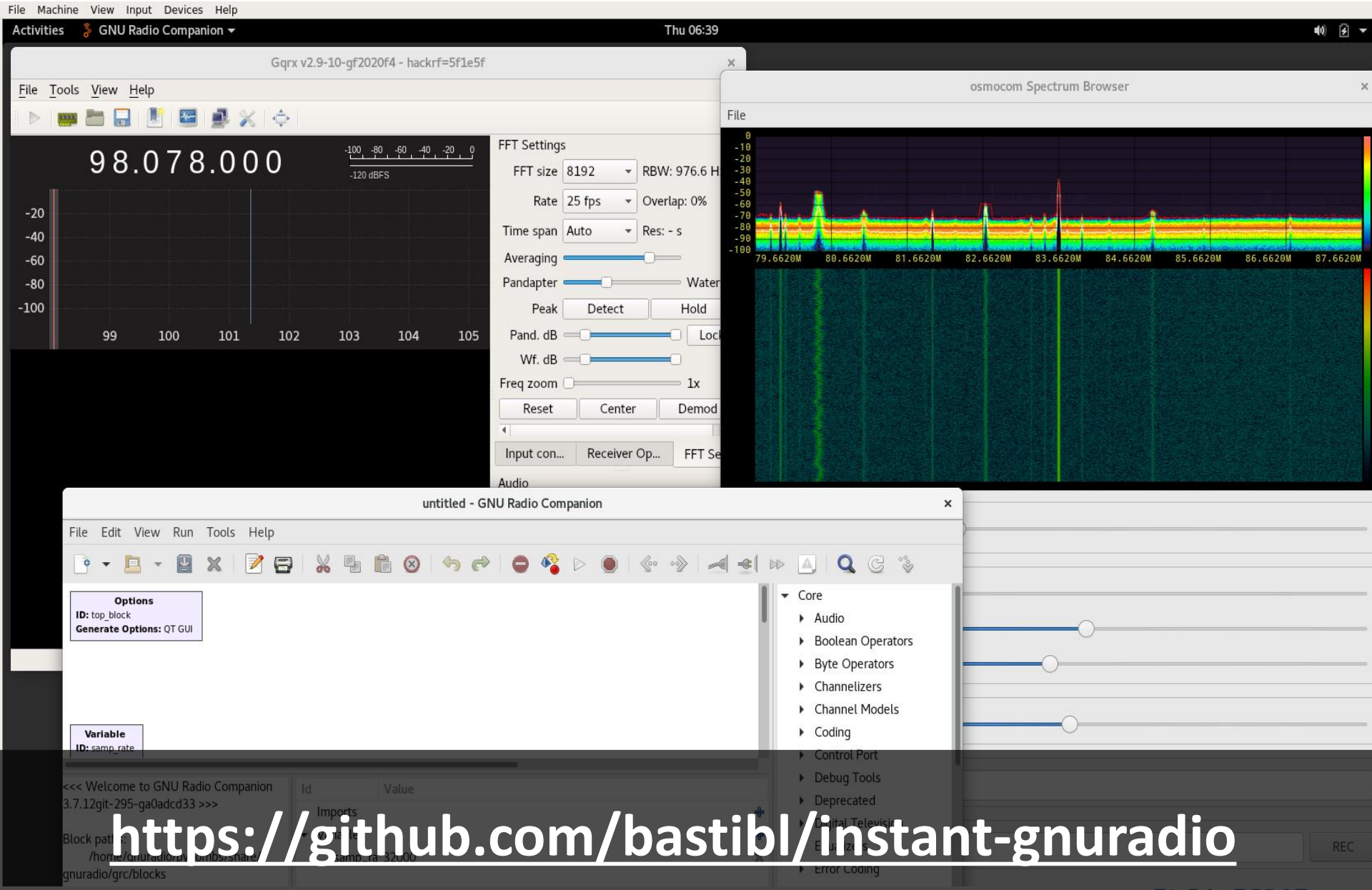
Energy-efficient wireless networks have many important applications in health care, industrial automation, and form the base for an Internet of Things. We provide a complete IEEE 802.15.4 stack that is interoperable with the ContikiOS up tc

WirelessHART LoWPAN zigbee

Instant GNU Radio

https://github.com/bastibl/instant-gnuradio

Instant GNU Radio



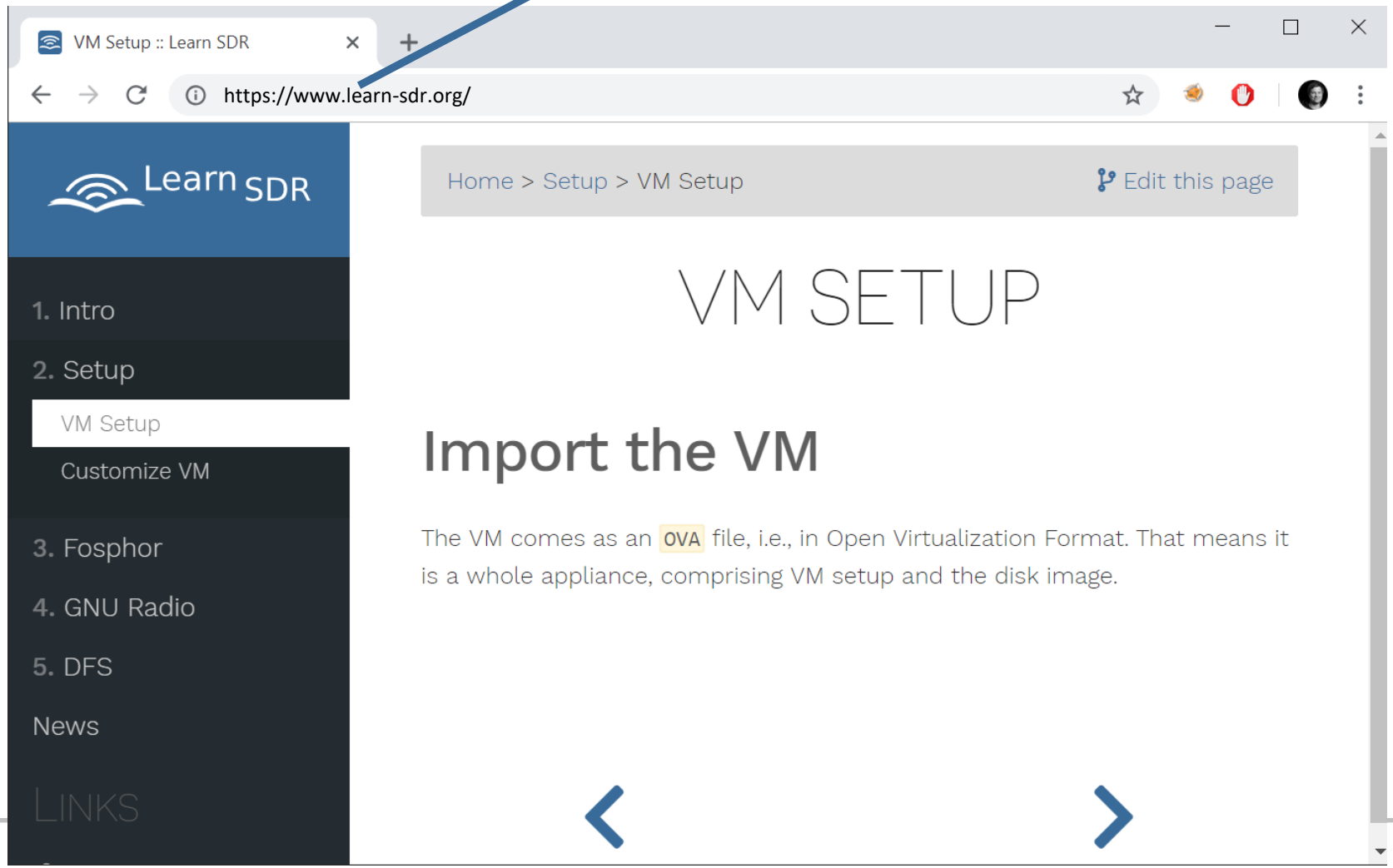
https://github.com/bastibl/instant-gnuradio

Instant GNU Radio

- Built with Packer
- VM Image
- Many applications pre-installed
- Easy to extend and customize

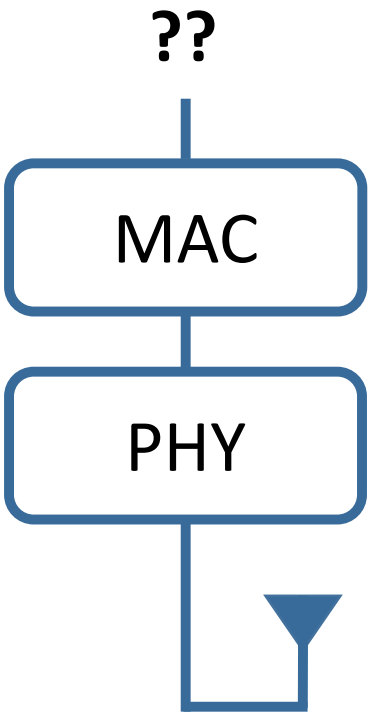
Learn SDR

www.learn-sdr.org



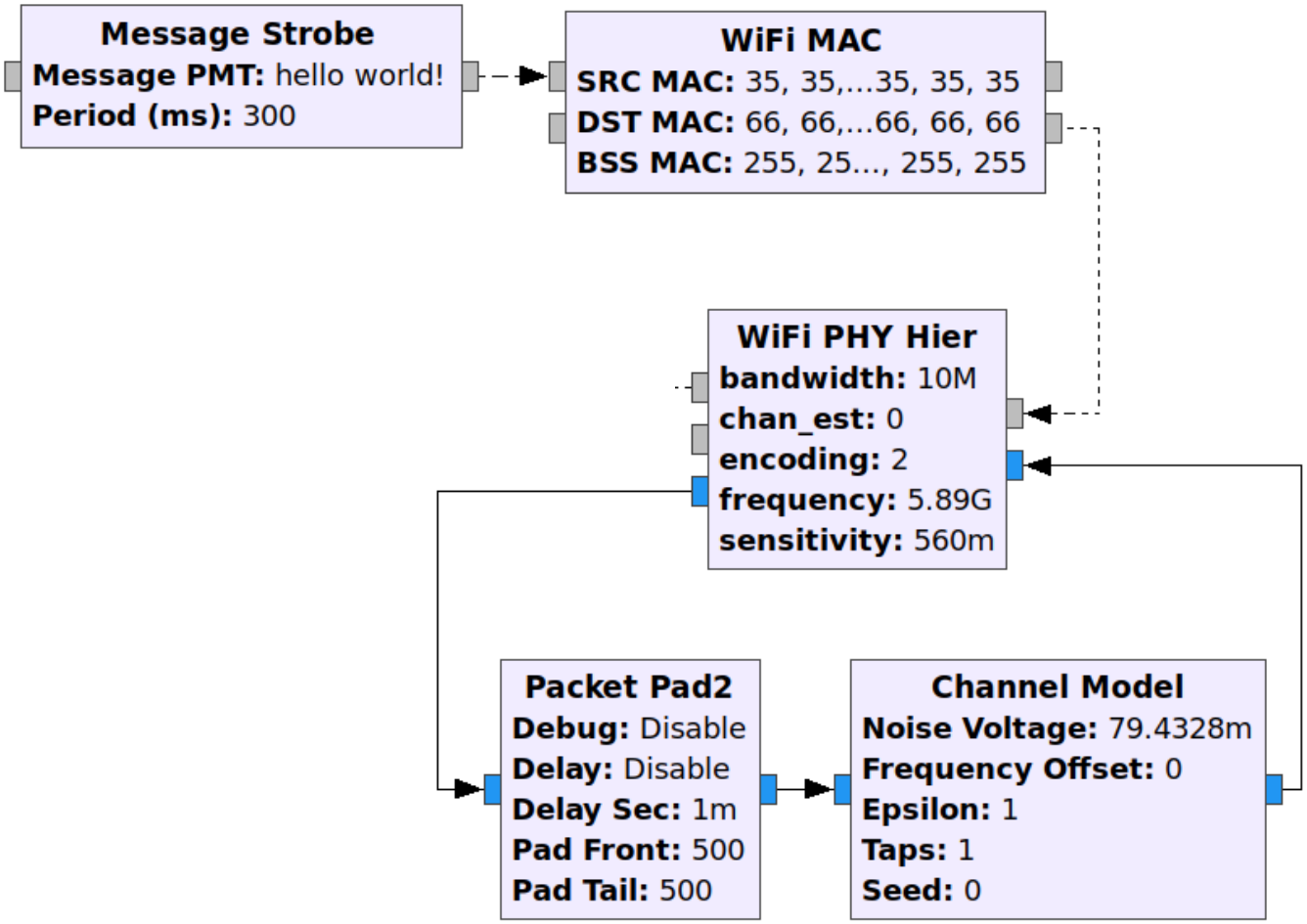
GNU Radio WLAN/ZigBee

- PHY only (MAC adds only static wrapper)
- No CSMA, no ACKs, no network stack
- **How can I send data?**
- **How can I interact with devices?**



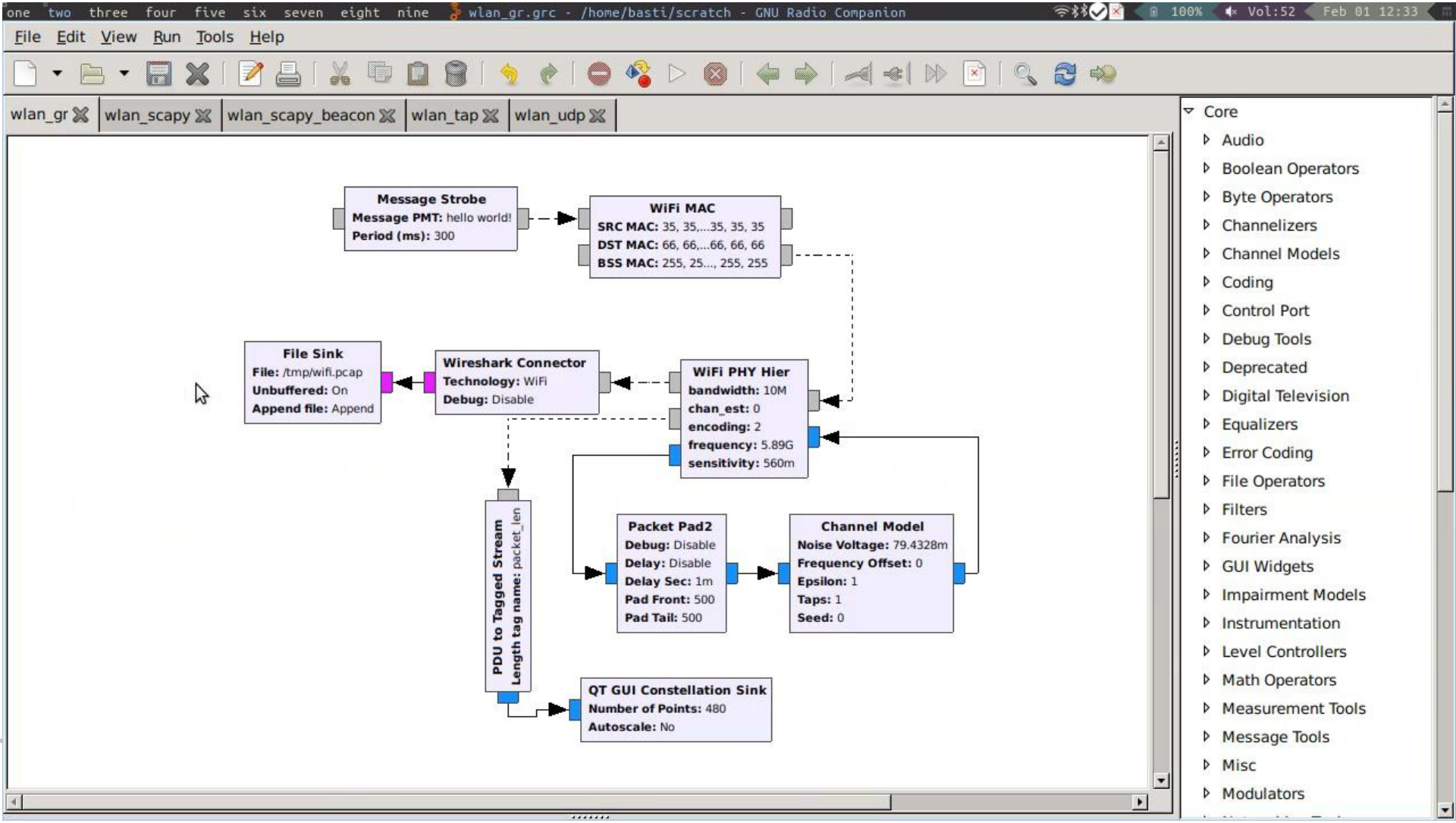
A GNU Radio Transceiver

“hello world!”	MAC	PHY

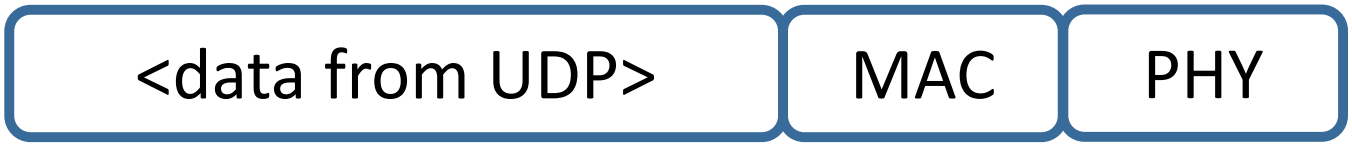


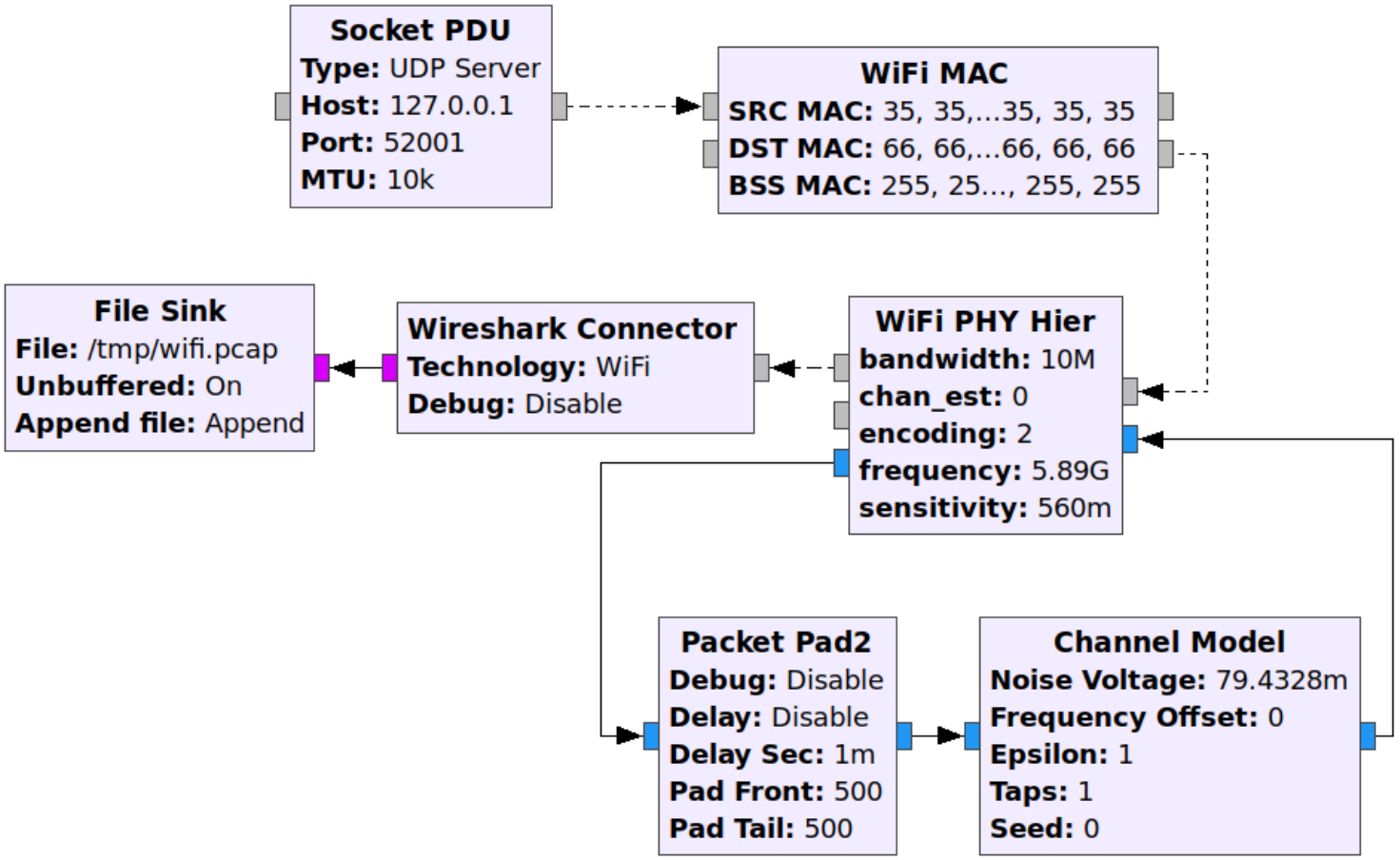
A Simple WLAN Frame

“hello world!”	MAC	PHY



A Simple WLAN Frame





Connecting to UDP Socket

- Netcat

```
32 nc -u 127.0.0.1 52001
33
```

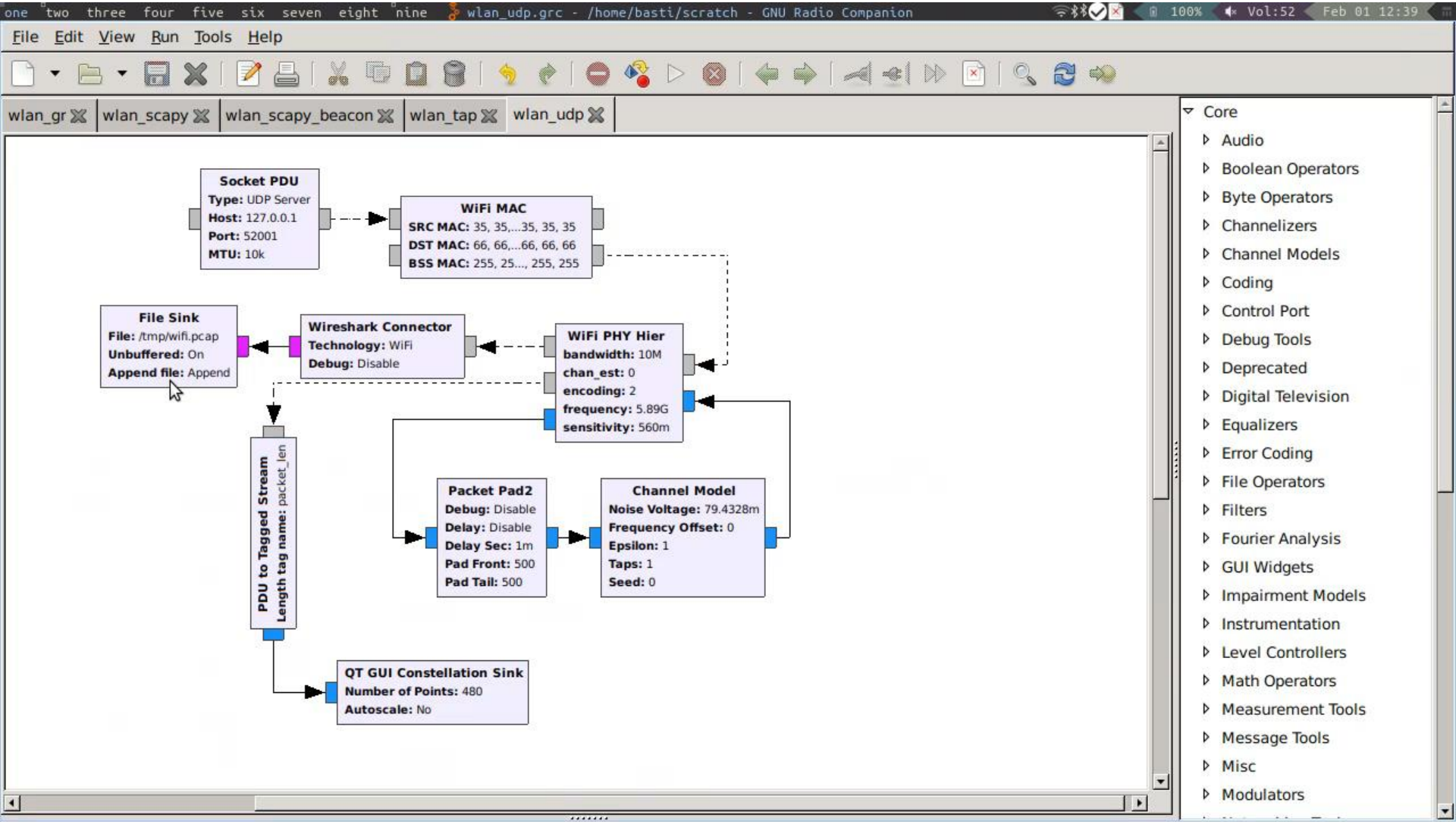
- Python

vim udp.py

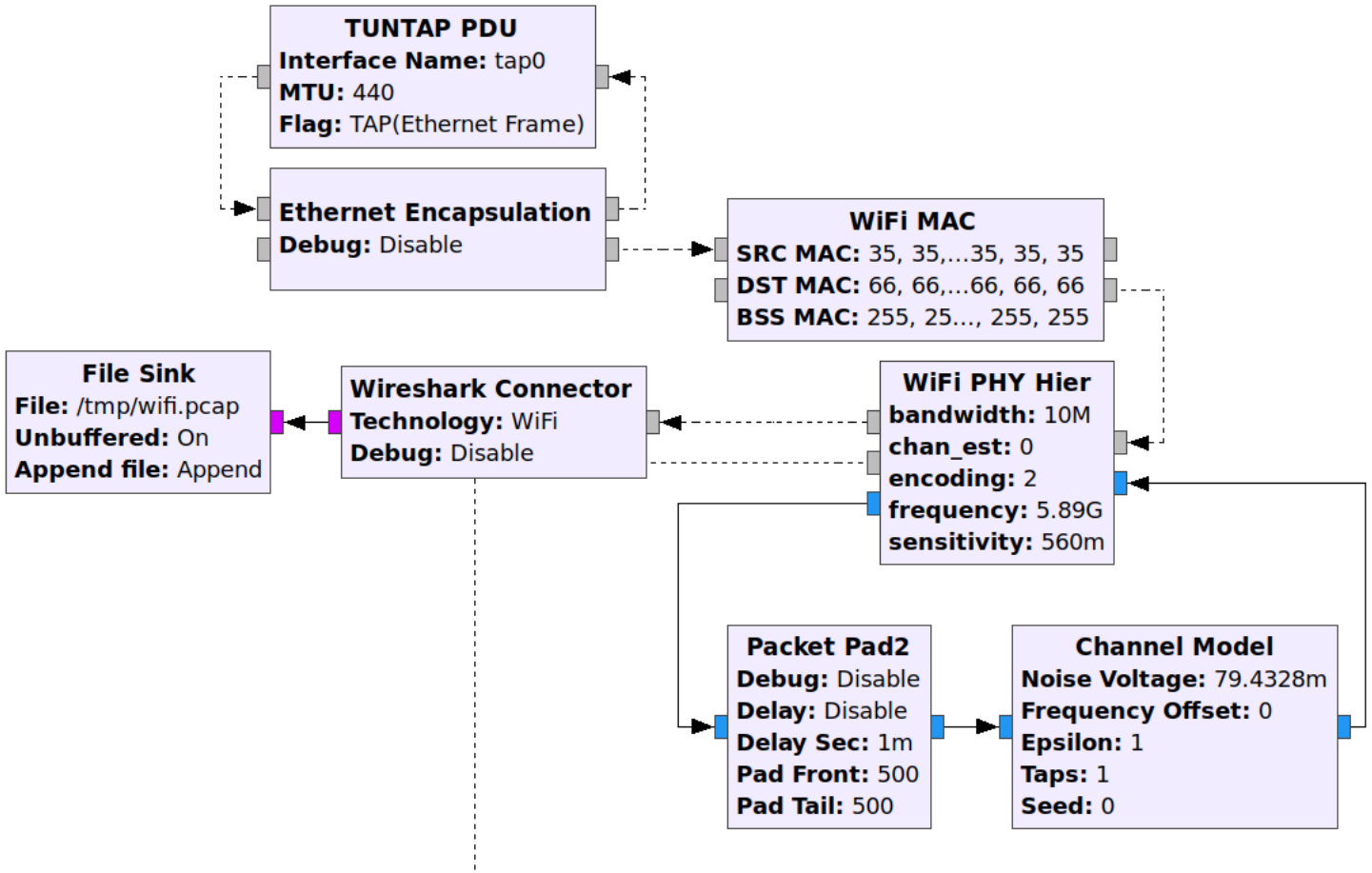
```
1 #!/usr/bin/env python
2
3 import socket
4
5 MESSAGE = "Hello, from python!\n"
6
7 sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
8 sock.sendto(MESSAGE, ("127.0.0.1", 52001))
```


A Simple WLAN Frame

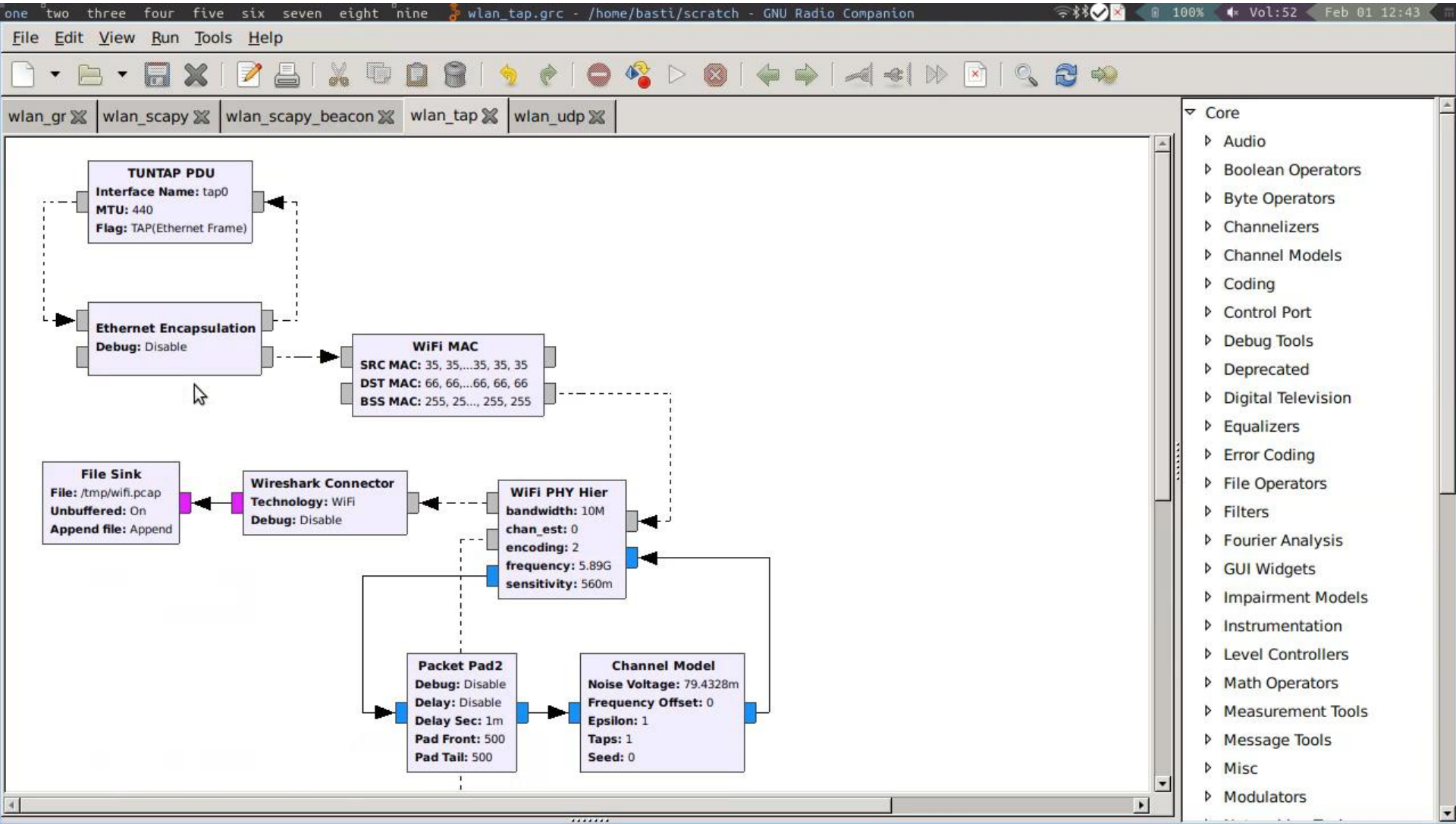
<data from UDP>	MAC	PHY



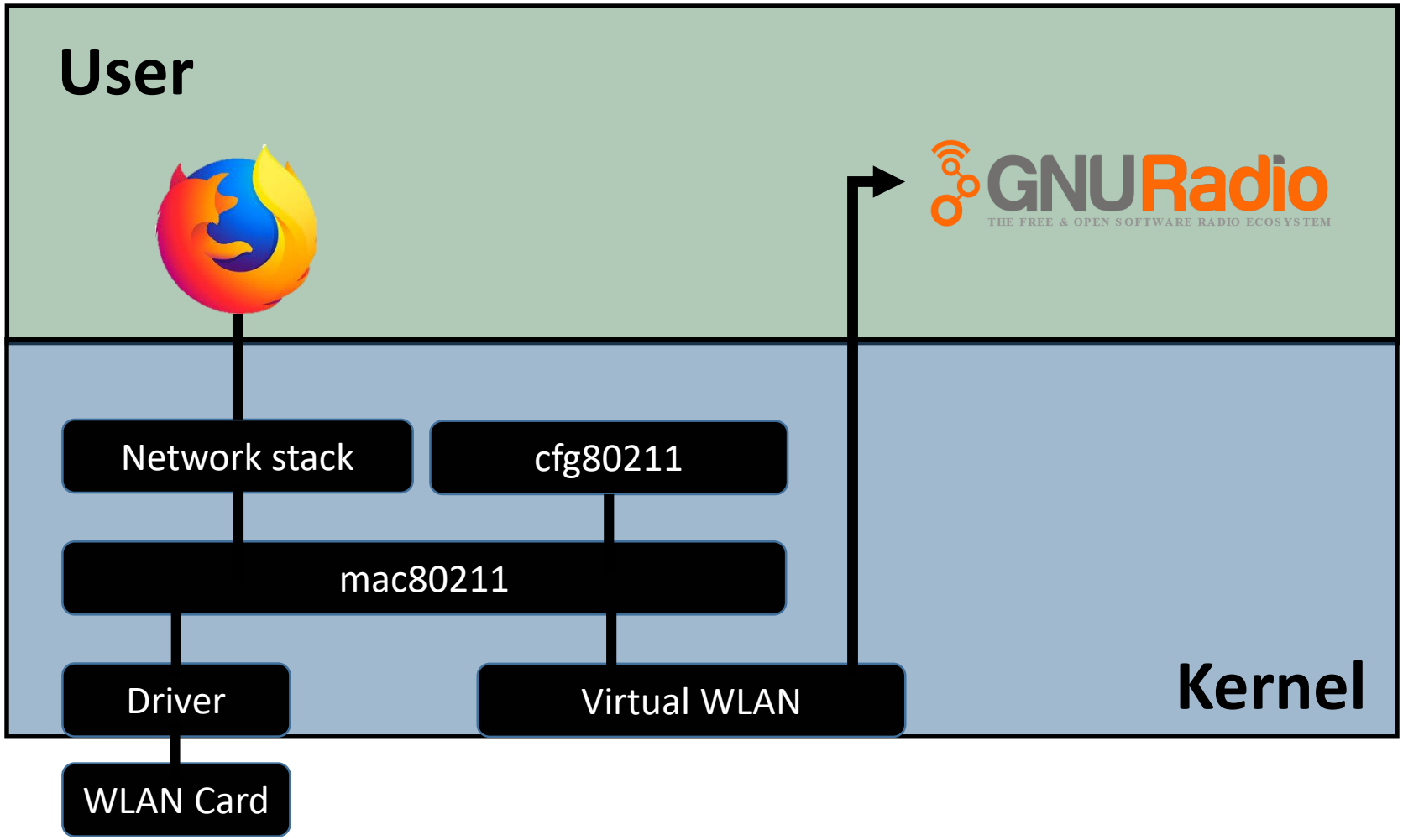
Connecting to the Network Stack



Connecting to the Network Stack



Virtual WLAN Device



Packet Crafting

- Ethernet frame (the hard way)

```
1 frame = struct.pack(\
2         "!6s6sH",\
3         binascii.unhexlify(dst.replace(":","")),\
4         binascii.unhexlify(src.replace(":","")),\
5         protocol)
```


Scapy

- Python turned into a domain-specific language
- Open Source
- https://scapy.net/

```
frames = Ether(dst="ff:ff:ff:ff:ff:ff")\
       /IP(dst=["gnuradio.org","libvolk.org"],ttl=(1,9))\
       /UDP(sport=12345, dport=52001)/"Hello World!"

list(frames)[0].show()
```

Packet Crafting

```
1 frames = Ether(dst="ff:ff:ff:ff:ff:ff")\
2       /IP(dst=["gnuradio.org","libvolk.org"],ttl=(1,9))\
3       /UDP(sport=12345, dport=52001)/"Hello World!"
4
5 list(frames)[0].show()
```

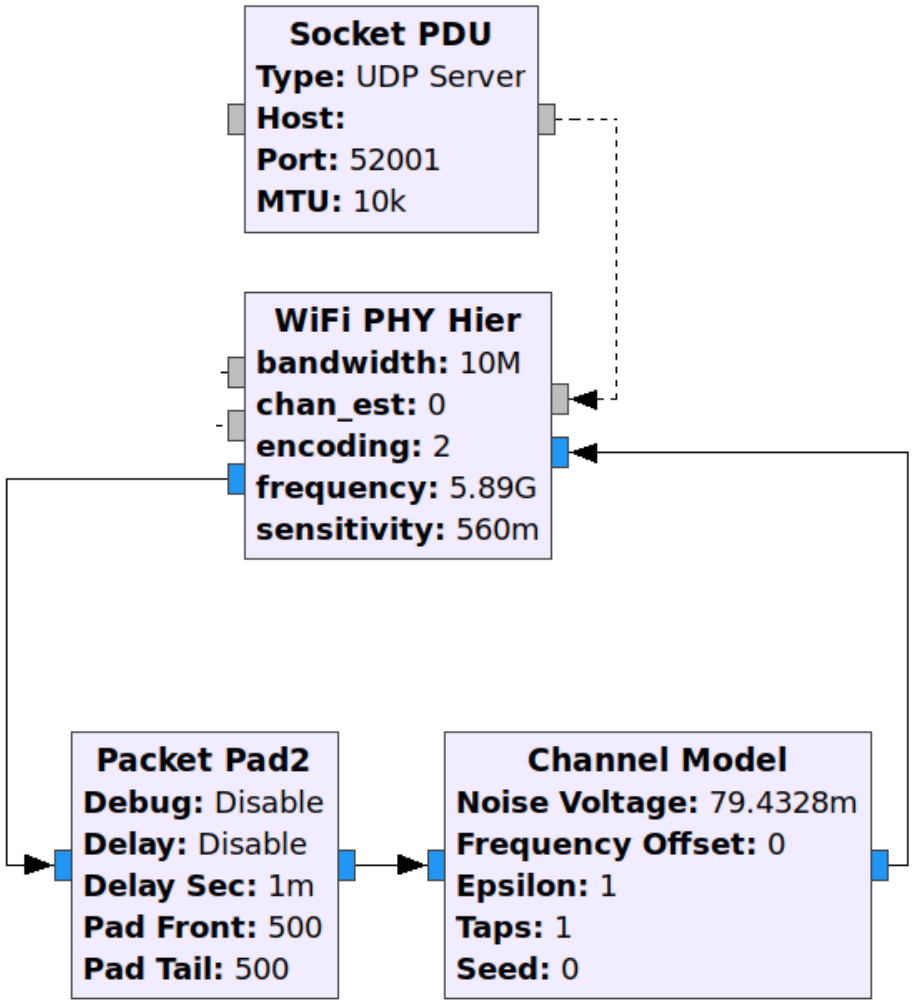
```
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 08:00:27:d0:52:2e
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = None
```

```
###[ UDP ]###
        sport     = 12345
        dport     = 52001
        len       = None
        chksum    = None
###[ Raw ]###
           load      = 'Hello World!'
```

Advantages

- More flexibility
  - Drivers
  - No device configuration
- No Prototypes (802.11p)
- More accessible (ZigBee)

Flow Graph with Scapy



WLAN Frames

- Beacon frame

```
frame = Dot11FCS(addr1='ff:..', addr2='23:..', addr3='23:..')\
        /Dot11Beacon()/Dot11Elt(ID='SSID', info='GR WLAN')

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(bytes(frame), ("127.0.0.1", 52001))
```

- Fuzzing

```
frame = Dot11FCS(addr1='ff:..', addr2='23:..', addr3='23:..')\
        /fuzz(Dot11Beacon()/Dot11Elt(ID='SSID', info='GR WLAN'))
```

- Deauth

```
frame = Dot11FCS(addr1=client,addr2=bssid,addr3=bssid)\
        /Dot11Deauth()
```


Smart Meter

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol		
2239	526.623187	0xee64	0x0000	ZigBee		
2240	526.626813			IEEE 8		
2241	527.249026			IEEE 8		
2242	527.279050	0xee64	0x0000	ZigBee		
2243	527.337580	0xee64	0x0000	ZigBee	57	Data, Dst: 0x0000, Src: 0xee64
2244	527.371972			IEEE 802.15.4	5	Ack
2245	527.375773	0xee64	0x0000	ZigBee	55	Command, Dst: 0x0000, Src: 0xee64
2246	527.412023	0xee64	0x0000	ZigBee	57	Data, Dst: 0x0000, Src: 0xee64
2247	527.436228			IEEE 802.15.4	5	Ack
2248	527.437268	0xee64	0x0000	ZigBee	55	Command, Dst: 0x0000, Src: 0xee64
2249	527.539784	0xee64	0x0000	ZigBee	57	Data, Dst: 0x0000, Src: 0xee64
2250	527.543641	0xee64	0x0000	ZigBee	55	Command, Dst: 0x0000, Src: 0xee64

```
▸ Frame 11: 55 bytes on wire (440 bits), 55 bytes captured (440 bits)
▾ IEEE 802.15.4 Data, Dst: 0x0000, Src: 0xee64
  ▸ Frame Control Field: 0x8861, Frame Type: Data, Acknowledge Request, PAN ID Compression, Destination Addressing Mode:
    Sequence Number: 187
    Destination PAN: 0x47d0
    Destination: 0x0000
    Source: 0xee64
    [Extended Source: SecureMe_00:00:28:45:44 (b8:79:7e:00:00:28:45:44)]
    [Origin: 1]
    FCS: 0x4e60 (Correct)
```

```
0000  61 88 bb d0 47 00 00 64  ee 09 1a 00 00 64 ee 1e   a···G··d ·····d··
0010  de d7 61 28 00 00 7e 79  b8 44 45 28 00 00 7e 79   ··a(··~y ·DE(··~y
0020  b8 28 07 18 00 00 44 45  28 00 00 7e 79 b8 14 7f   ·(····DE (··~y···
0030  f7 d7 06 14 aa 60 4e                               ·····`N
```

ZigBee Frame Injection

- Data

```
frame = Dot15d4FCS()\
        /Dot15d4Data(dest_panid=0x4242, dest_addr=0x2323)\
        /"Hello World!"

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(bytes(frame), ("127.0.0.1", 52001))
```

- Fuzzing

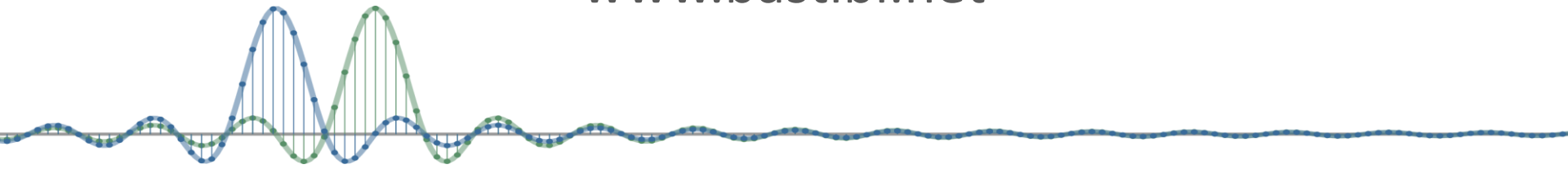
```
frame = fuzz(Dot15d4FCS()\
        /Dot15d4Data(dest_panid=0x4242, dest_addr=0x2323))

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.sendto(bytes(frame), ("127.0.0.1", 52001))
```


Demo

GNU Radio Meets Scapy

mail@bastibl.net

www.bastibl.net



February 2019 ▪ FOSDEM SDR Dev Room ▪ Brussels, Belgium