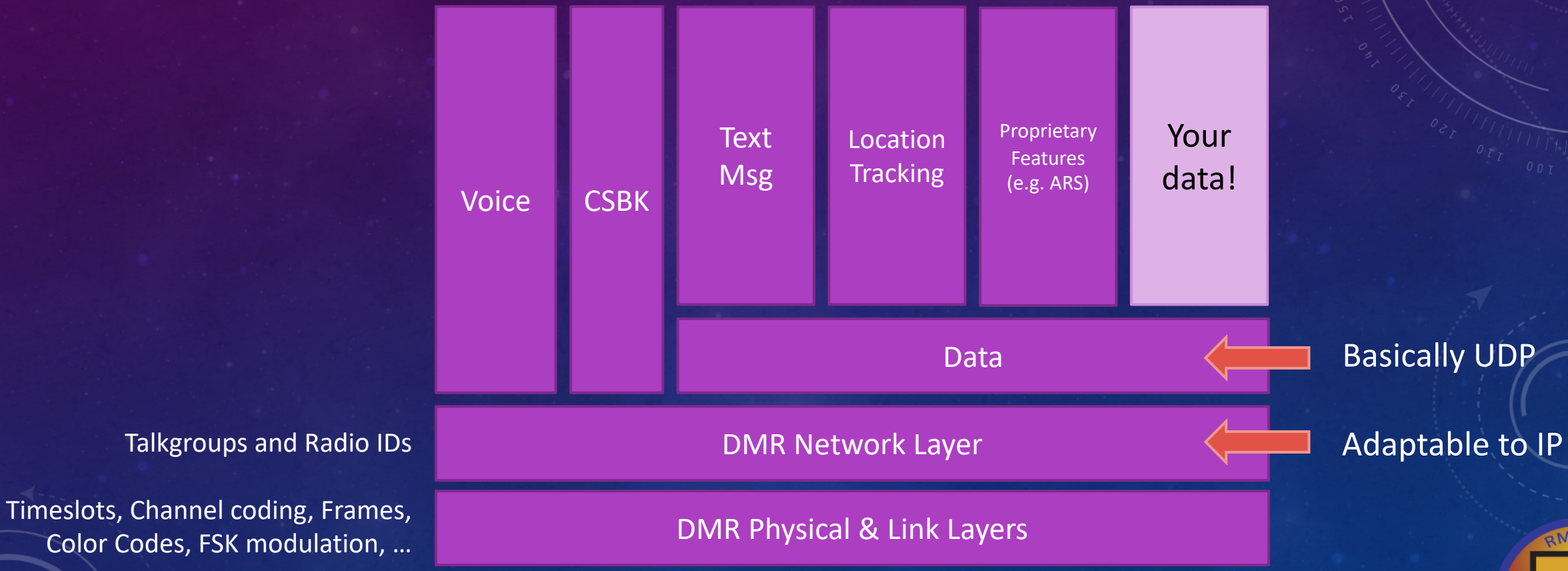
IP DATA OVER DMR

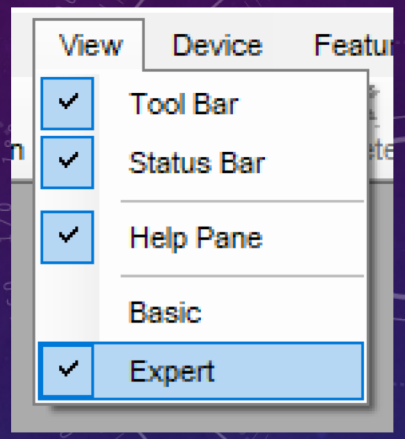
TRISTAN HONSCHEID, NM0TH

FEB 11, 2023

DMR BEYOND VOICE



CLUES IN MOTOTRBO CPS



Services

ARS Radio ID	
ARS IP	0.0.0.0
ARS UDP Port	4005
TMS Radio ID	
TMS IP	0.0.0.0
TMS UDP Port	4007
User Defined UDP Port 1	Disabled
User Defined UDP Port 2	Disabled
User Defined UDP Port 3	Disabled
XCMP Server ID	
XCMP Server IP	0.0.0.0
Battery Management Server ID	
Battery Management Server IP	0.0.0.0

Compressed UDP Data Header: None

ARS Monitoring ID	
ARS Monitoring IP	0.0.0.0
Location Server UDP Port	4001
XCMP Server UDP Port	4004
Battery Management Server UDP Port	4012

YOUR RADIO IS A USB NETWORK CARD

- Radio is a USB Ethernet Adapter
- Radio hosts a DHCP server that assigns PC an IP address
  - Usually 192.168.10.2/24
- Effectively a link network


```
Ethernet adapter usb_xhci 7:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8363:b5ec
   IPv4 Address. . . . . . . . . . . : 192.168.10.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.10.1
```

YOUR RADIO IS A ROUTER

WiFi
(DHCP)

Bluetooth PAN
192.168.11.1/24

USB
192.168.10.1/24

Internal Services
Text messaging, etc

DMR CAI
12.X.X.X/8
13.X.X.X/8

ID TO IP MAPPING

Radio (Individual) IDs

ID: 12345678 → Hex → BC 61 4E

Each Octet to Decimal → 12.188.97.78

Arbitrary Class A network set in CPS

CAI Network 12

Packets set to 13.188.97.78 are forwarded to USB-attached PC.

(CAI Network + 1)

Talkgroup IDs (Group Calls)

Same process, but uses IPv4 multicast address space.

ID: 700 → 225.0.2.188

CAI Group Network 225

ROUTING SETUP

```
# Add routes so your PC knows how to reach the
# DMR CAI network

$ ip route add 12.0.0.0/8 via 192.168.10.1
$ ip route add 13.0.0.0/8 via 192.168.10.1
```

Smaller subnets appear to work if they cover your radio ID range

DEMOS

Mobile Station
XPR 7550e
ID 101

Control Station
XPR 5350
ID 100

DEMO #1: PING

- Mobile Station laptop pings the control station radio
- ICMP ping packets go over the air and are answered by the control station radio's internal IP stack

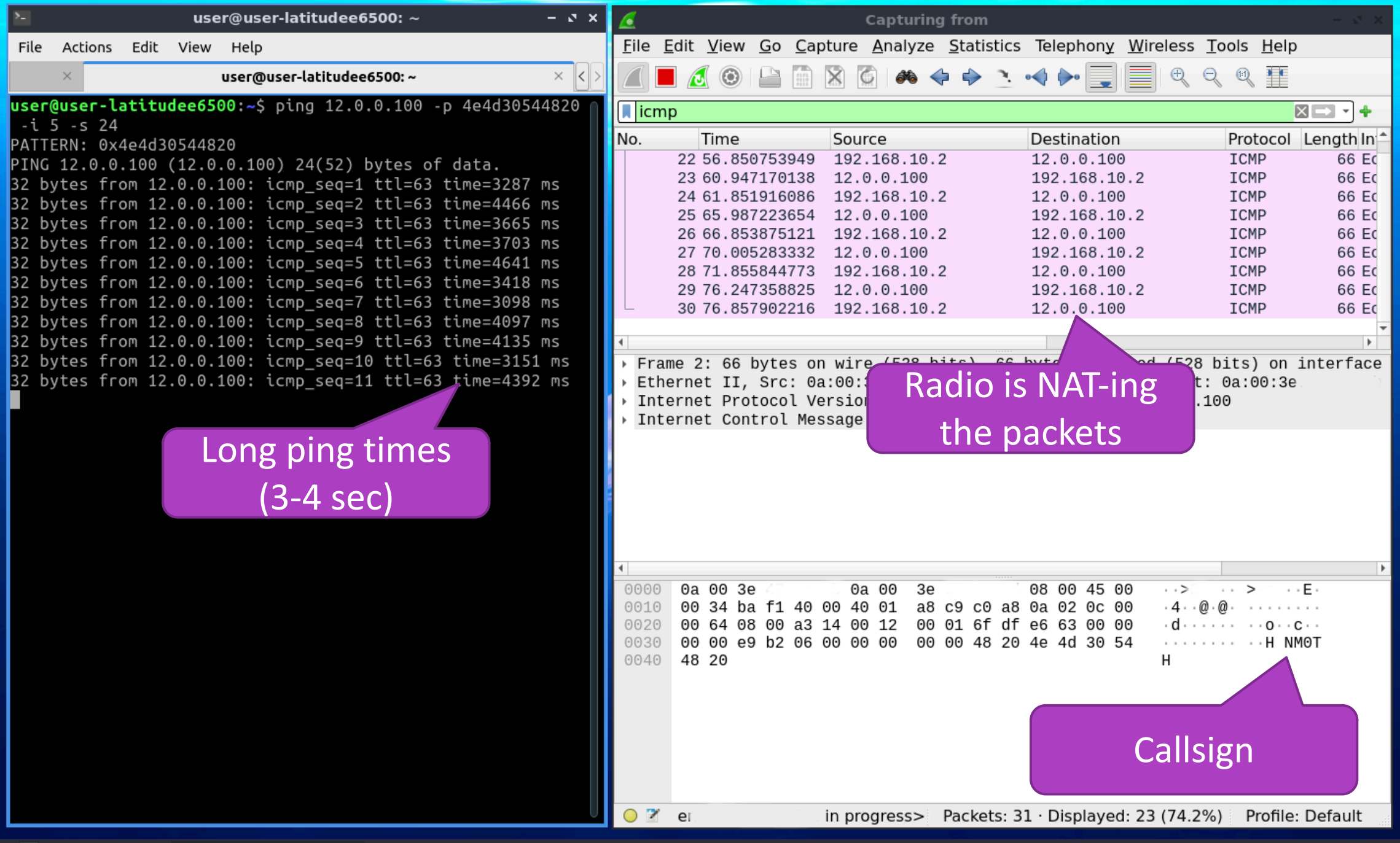
```
$ ping 12.0.0.100 -p 4e4d30544820 -i 5 -s 24
```

Control station radio ID encoded as IP address

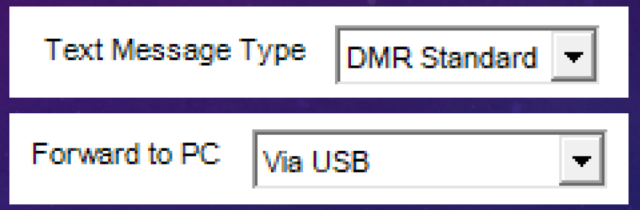
Embed my callsign In the packets for ID (just hex ASCII chars)

5 sec between pings

Smaller packets


user@user-latitudee6500: ~
File Actions Edit View Help
user@user-latitudee6500: ~
user@user-latitudee6500:~$ ping 12.0.0.100 -p 4e4d30544820
-i 5 -s 24
PATTERN: 0x4e4d30544820
PING 12.0.0.100 (12.0.0.100) 24(52) bytes of data.
32 bytes from 12.0.0.100: icmp_seq=1 ttl=63 time=3287 ms
32 bytes from 12.0.0.100: icmp_seq=2 ttl=63 time=4466 ms
32 bytes from 12.0.0.100: icmp_seq=3 ttl=63 time=3665 ms
32 bytes from 12.0.0.100: icmp_seq=4 ttl=63 time=3703 ms
32 bytes from 12.0.0.100: icmp_seq=5 ttl=63 time=4641 ms
32 bytes from 12.0.0.100: icmp_seq=6 ttl=63 time=3418 ms
32 bytes from 12.0.0.100: icmp_seq=7 ttl=63 time=3098 ms
32 bytes from 12.0.0.100: icmp_seq=8 ttl=63 time=4097 ms
32 bytes from 12.0.0.100: icmp_seq=9 ttl=63 time=4135 ms
32 bytes from 12.0.0.100: icmp_seq=10 ttl=63 time=3151 ms
32 bytes from 12.0.0.100: icmp_seq=11 ttl=63 time=4392 ms
Long ping times (3-4 sec)
Capturing from
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
icmp
No. Time Source Destination Protocol Length
22 56.850753949 192.168.10.2 12.0.0.100 ICMP 66
23 60.947170138 12.0.0.100 192.168.10.2 ICMP 66
24 61.851916086 192.168.10.2 12.0.0.100 ICMP 66
25 65.987223654 12.0.0.100 192.168.10.2 ICMP 66
26 66.853875121 192.168.10.2 12.0.0.100 ICMP 66
27 70.005283332 12.0.0.100 192.168.10.2 ICMP 66
28 71.855844773 192.168.10.2 12.0.0.100 ICMP 66
29 76.247358825 12.0.0.100 192.168.10.2 ICMP 66
30 76.857902216 192.168.10.2 12.0.0.100 ICMP 66
Radio is NAT-ing the packets
0000 0a 00 3e 0a 00 3e 08 00 45 00
0010 00 34 ba f1 40 00 40 01 a8 c9 c0 a8 0a 02 0c 00
0020 00 64 08 00 a3 14 00 12 00 01 6f df e6 63 00 00
0030 00 00 e9 b2 06 00 00 00 00 00 48 20 4e 4d 30 54
0040 48 20
Callsign
Packets: 31 · Displayed: 23 (74.2%)
Profile: Default
user@user-latitudee6...
Capturing from enx0a...
17:22

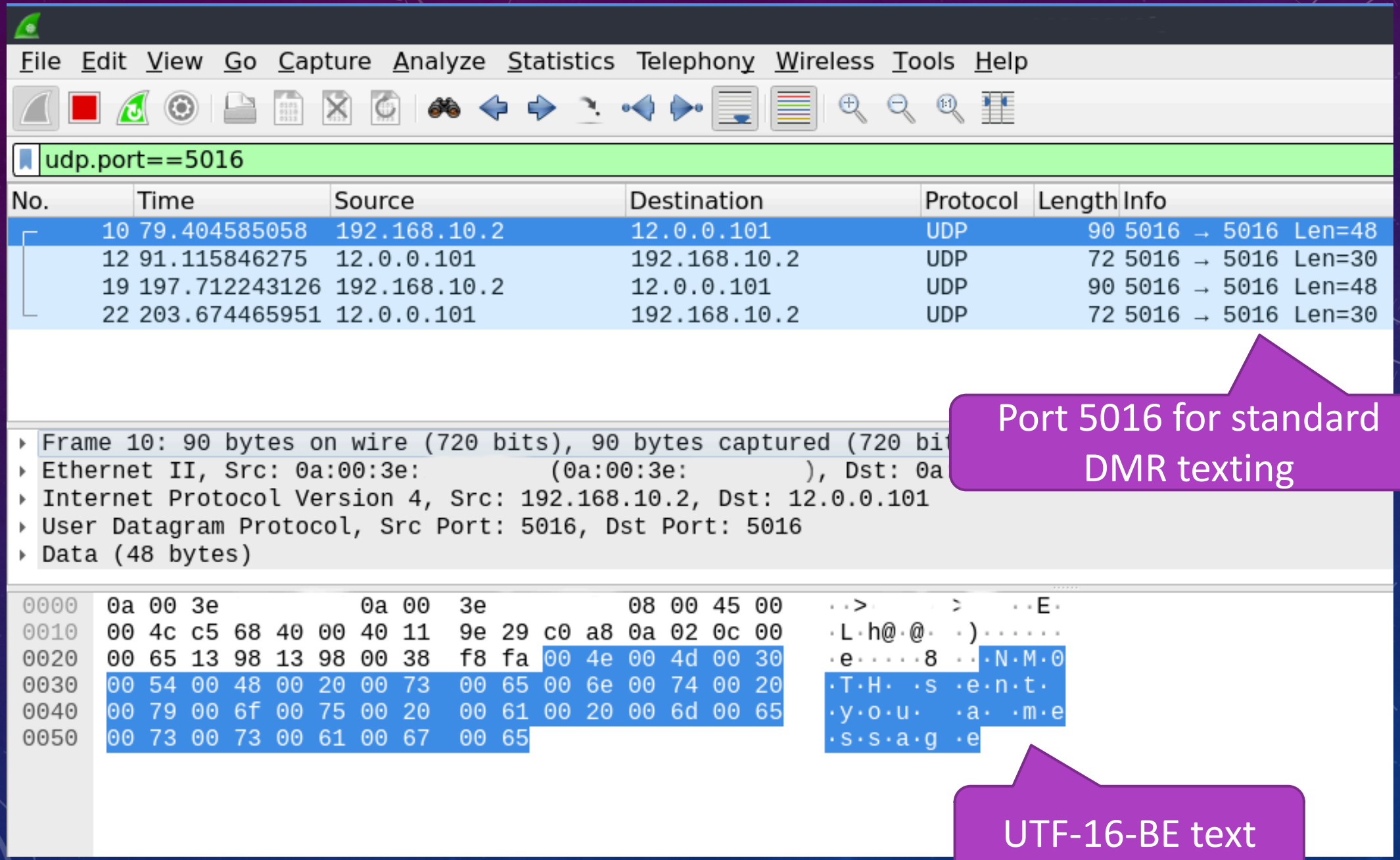
DEMO #2: TEXT MESSAGE I/O



Channel setting

Network menu

- Use DMR Standard message format
- Text message forwarding enabled
- Control station PC sends a text to the other radio and listens for a reply
  - Simple custom Python script to encode text and send packet
  - Text message is just a UDP packet on port 5016
  - Body is UTF-16 encoded (big endian)

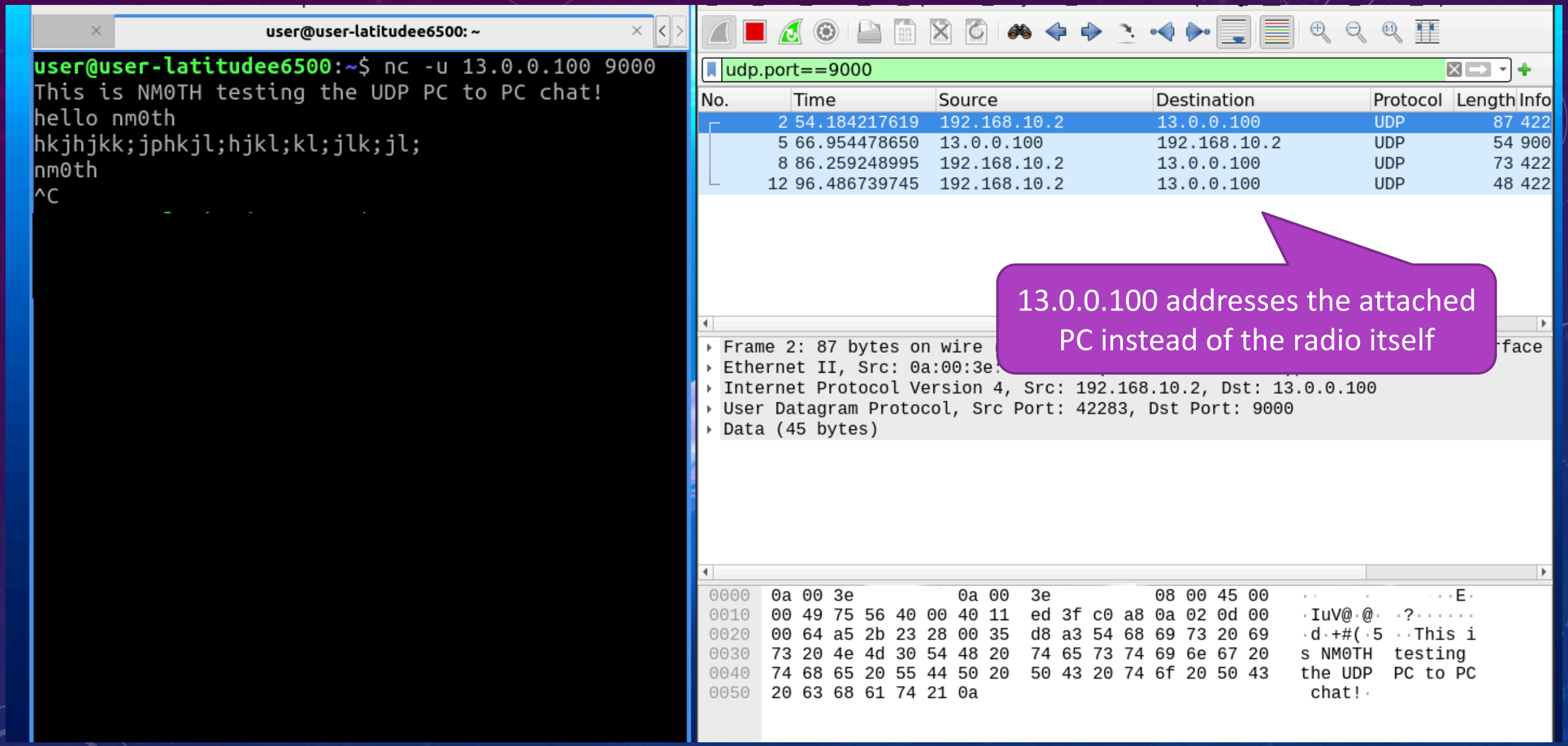

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.port==5016
No. Time Source Destination Protocol Length Info
10 79.404585058 192.168.10.2 12.0.0.101 UDP 90 5016 → 5016 Len=48
12 91.115846275 12.0.0.101 192.168.10.2 UDP 72 5016 → 5016 Len=30
19 197.712243126 192.168.10.2 12.0.0.101 UDP 90 5016 → 5016 Len=48
22 203.674465951 12.0.0.101 192.168.10.2 UDP 72 5016 → 5016 Len=30
Frame 10: 90 bytes on wire (720 bits), 90 bytes captured (720 bi
Ethernet II, Src: 0a:00:3e: (0a:00:3e:), Dst: 0a
Internet Protocol Version 4, Src: 192.168.10.2, Dst: 12.0.0.101
User Datagram Protocol, Src Port: 5016, Dst Port: 5016
Data (48 bytes)
0000 0a 00 3e 0a 00 3e 08 00 45 00
0010 00 4c c5 68 40 00 40 11 9e 29 c0 a8 0a 02 0c 00
0020 00 65 13 98 13 98 00 38 f8 fa 00 4e 00 4d 00 30
0030 00 54 00 48 00 20 00 73 00 65 00 6e 00 74 00 20
0040 00 79 00 6f 00 75 00 20 00 61 00 20 00 6d 00 65
0050 00 73 00 73 00 61 00 67 00 65
Port 5016 for standard DMR texting
UTF-16-BE text

DEMO #3: PC ↔ PC INTERACTIVE CHAT

- This time, PC to PC traffic. Use 13.0.0.100 and .101
- Control station PC hosts a UDP server on port 9000 (Netcat)
- Mobile station PC has a client

```
# Mobile station
$ nc -u 13.0.0.100 9000
```

```
# Control station
$ nc -ul 9000
```


user@user-latitudee6500: ~
user@user-latitudee6500:~$ nc -u 13.0.0.100 9000
This is NM0TH testing the UDP PC to PC chat!
hello nm0th
hkjhjkk;jphkjl;hjkl;kl;jlk;jl;
nm0th
^C
udp.port==9000
No. Time Source Destination Protocol Length Info
2 54.184217619 192.168.10.2 13.0.0.100 UDP 87 422
5 66.954478650 13.0.0.100 192.168.10.2 UDP 54 900
8 86.259248995 192.168.10.2 13.0.0.100 UDP 73 422
12 96.486739745 192.168.10.2 13.0.0.100 UDP 48 422
13.0.0.100 addresses the attached PC instead of the radio itself
Frame 2: 87 bytes on wire
Ethernet II, Src: 0a:00:3e:
Internet Protocol Version 4, Src: 192.168.10.2, Dst: 13.0.0.100
User Datagram Protocol, Src Port: 42283, Dst Port: 9000
Data (45 bytes)
0000 0a 00 3e 0a 00 3e 08 00 45 00
0010 00 49 75 56 40 00 40 11 ed 3f c0 a8 0a 02 0d 00
0020 00 64 a5 2b 23 28 00 35 d8 a3 54 68 69 73 20 69
0030 73 20 4e 4d 30 54 48 20 74 65 73 74 69 6e 67 20
0040 74 68 65 20 55 44 50 20 50 43 20 74 6f 20 50 43
0050 20 63 68 61 74 21 0a

DMR DATA TERMINOLOGY

- Confirmed Data – Receiving radio must ACK or sending radio will retry.
  - Only for individual (not group) calls
  - Not good for Part 97 ID rules
- Unconfirmed Data – No retries
  - Can be multicasted to the talkgroup

Channel setting

IDENTIFICATION

- DMR IDs are not legal identification
- No built-in way to tag packets with your callsign
  - Need to handle this at the application layer
  - For example, encapsulate AX.25 or include your callsign in text messages

APPLICATIONS

- Why not APRS?
  - Sometimes it makes sense to leverage an existing DMR network or consolidate
  - DMR is more robust
- Custom applications for events and deployments
  - Mobile stations can report status or make requests to a central dispatch server
- Fun experimentation like text gateways

FURTHER READING

- ETSI DMR specification – details the entire DMR protocol, including IP-attached peripherals
  - https://www.dmrassociation.org/dmr-standards.html
- MOTOTRBO CPS Help Pages – lots of info on configuring data-related settings on radios