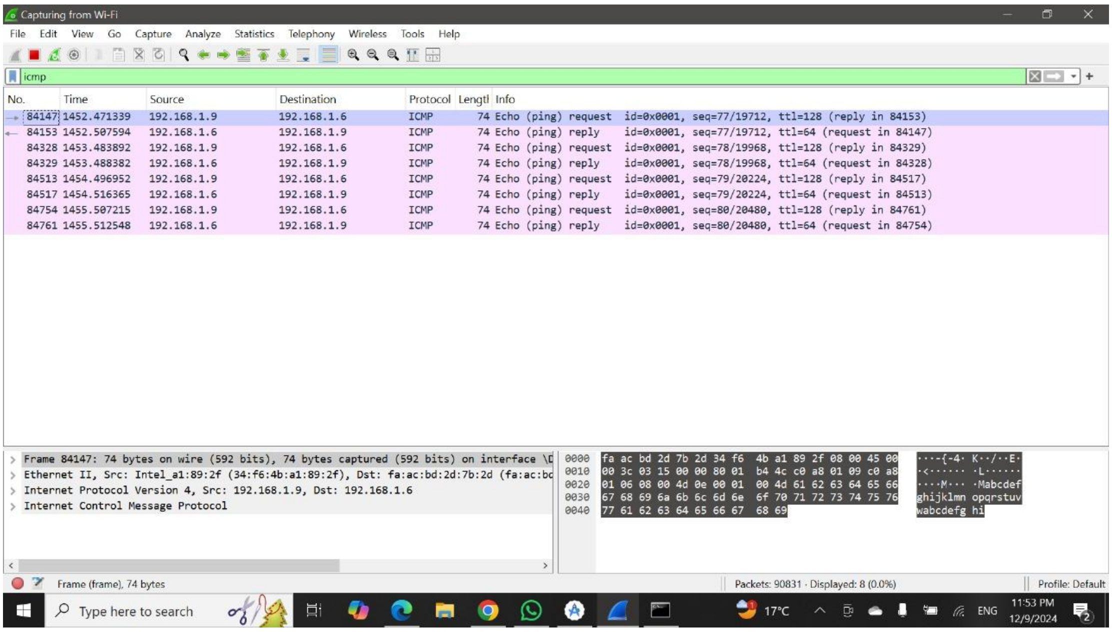
ICMP packets



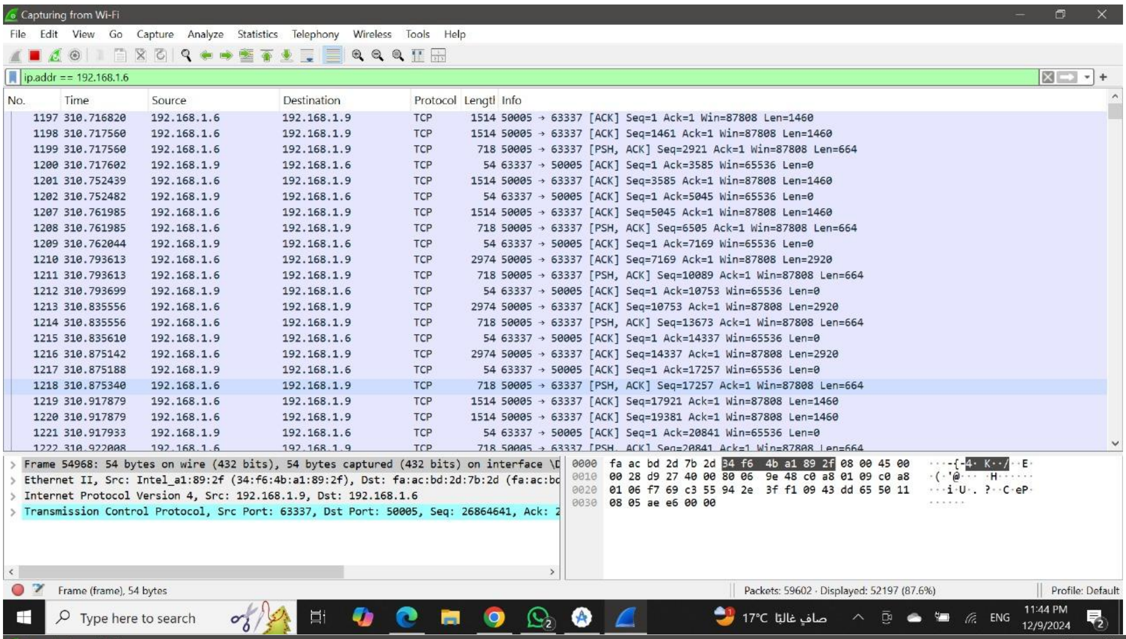
ICMP is used for pinging and make sure the connection is stable.

It is used for reporting errors and performing network diagnostics:

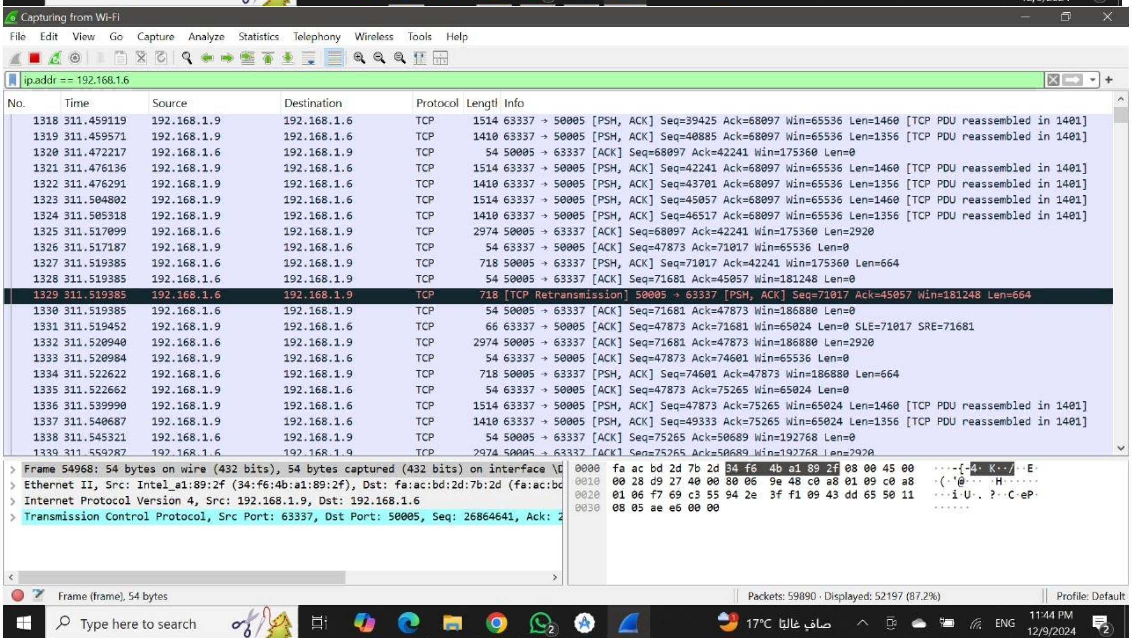
A. Error Reporting:- Unreachable destination or time exceeded

B. Diagnostic Tools:- Ping and trace route

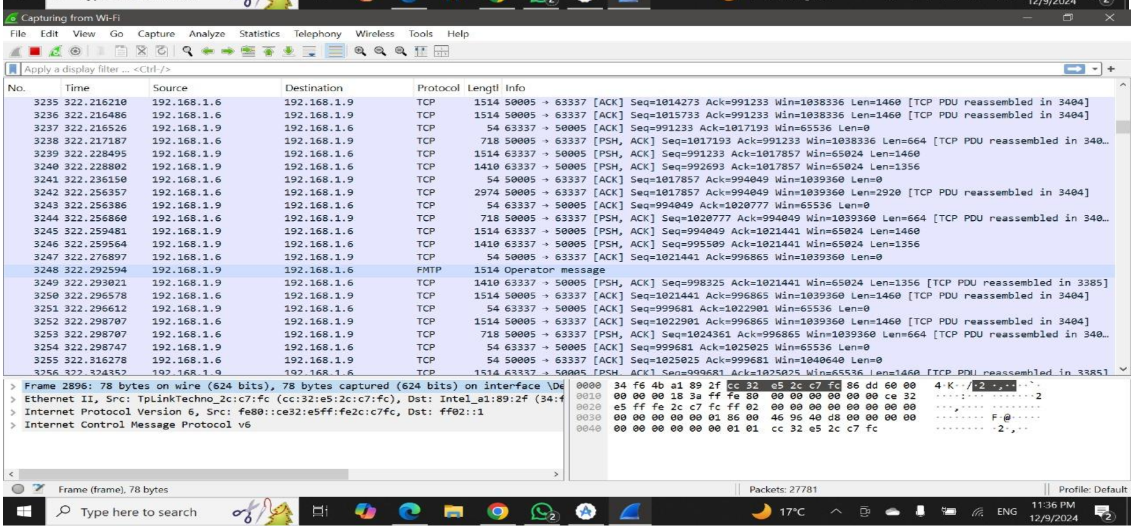
TCP Packets



Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 192.168.1.6
No. Time Source Destination Protocol Length Info
1197 310.716820 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=1 Ack=1 Win=87808 Len=1460
1198 310.717560 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=1461 Ack=1 Win=87808 Len=1460
1199 310.717560 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=2921 Ack=1 Win=87808 Len=664
1200 310.717602 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=1 Ack=3585 Win=65536 Len=0
1201 310.752439 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=3585 Ack=1 Win=87808 Len=1460
1202 310.752482 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=1 Ack=5045 Win=65536 Len=0
1207 310.761985 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=5045 Ack=1 Win=87808 Len=1460
1208 310.761985 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=6505 Ack=1 Win=87808 Len=664
1209 310.762044 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=1 Ack=7169 Win=65536 Len=0
1210 310.793613 192.168.1.6 192.168.1.9 TCP 2974 50005 → 63337 [ACK] Seq=7169 Ack=1 Win=87808 Len=2920
1211 310.793613 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=10089 Ack=1 Win=87808 Len=664
1212 310.793699 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=1 Ack=10753 Win=65536 Len=0
1213 310.835556 192.168.1.6 192.168.1.9 TCP 2974 50005 → 63337 [ACK] Seq=10753 Ack=1 Win=87808 Len=2920
1214 310.835556 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=13673 Ack=1 Win=87808 Len=664
1215 310.835610 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=1 Ack=14337 Win=65536 Len=0
1216 310.875142 192.168.1.6 192.168.1.9 TCP 2974 50005 → 63337 [ACK] Seq=14337 Ack=1 Win=87808 Len=2920
1217 310.875188 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=1 Ack=17257 Win=65536 Len=0
1218 310.875340 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=17257 Ack=1 Win=87808 Len=664
1219 310.917879 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=17921 Ack=1 Win=87808 Len=1460
1220 310.917879 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=19381 Ack=1 Win=87808 Len=1460
1221 310.917933 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=1 Ack=20841 Win=65536 Len=0
Frame 54968: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface
Ethernet II, Src: Intel_a1:89:2f (34:f6:4b:a1:89:2f), Dst: fa:ac:bd:2d:7b:2d
Internet Protocol Version 4, Src: 192.168.1.9, Dst: 192.168.1.6
Transmission Control Protocol, Src Port: 63337, Dst Port: 50005, Seq: 26864641, Ack:
Frame (frame), 54 bytes
Packets: 59602 · Displayed: 52197 (87.6%)
Profile: Default
Type here to search
17°C صاف غالبا
ENG
11:44 PM
12/9/2024



Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 192.168.1.6
No. Time Source Destination Protocol Length Info
1318 311.459119 192.168.1.9 192.168.1.6 TCP 1514 63337 → 50005 [PSH, ACK] Seq=39425 Ack=68097 Win=65536 Len=1460 [TCP PDU reassembled in 1401]
1319 311.459571 192.168.1.9 192.168.1.6 TCP 1410 63337 → 50005 [PSH, ACK] Seq=40885 Ack=68097 Win=65536 Len=1356 [TCP PDU reassembled in 1401]
1320 311.472217 192.168.1.6 192.168.1.9 TCP 54 50005 → 63337 [ACK] Seq=68097 Ack=42241 Win=175360 Len=0
1321 311.476136 192.168.1.9 192.168.1.6 TCP 1514 63337 → 50005 [PSH, ACK] Seq=42241 Ack=68097 Win=65536 Len=1460 [TCP PDU reassembled in 1401]
1322 311.476291 192.168.1.9 192.168.1.6 TCP 1410 63337 → 50005 [PSH, ACK] Seq=43701 Ack=68097 Win=65536 Len=1356 [TCP PDU reassembled in 1401]
1323 311.504802 192.168.1.9 192.168.1.6 TCP 1514 63337 → 50005 [PSH, ACK] Seq=45057 Ack=68097 Win=65536 Len=1460 [TCP PDU reassembled in 1401]
1324 311.505318 192.168.1.9 192.168.1.6 TCP 1410 63337 → 50005 [PSH, ACK] Seq=46517 Ack=68097 Win=65536 Len=1356 [TCP PDU reassembled in 1401]
1325 311.517099 192.168.1.6 192.168.1.9 TCP 2974 50005 → 63337 [ACK] Seq=68097 Ack=42241 Win=175360 Len=2920
1326 311.517187 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=47873 Ack=71017 Win=65536 Len=0
1327 311.519385 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=71017 Ack=42241 Win=175360 Len=664
1328 311.519385 192.168.1.6 192.168.1.9 TCP 54 50005 → 63337 [ACK] Seq=71681 Ack=45057 Win=181248 Len=0
1329 311.519385 192.168.1.6 192.168.1.9 TCP 718 [TCP Retransmission] 50005 → 63337 [PSH, ACK] Seq=71017 Ack=45057 Win=181248 Len=664
1330 311.519385 192.168.1.6 192.168.1.9 TCP 54 50005 → 63337 [ACK] Seq=71681 Ack=47873 Win=186880 Len=0
1331 311.519452 192.168.1.9 192.168.1.6 TCP 66 63337 → 50005 [ACK] Seq=47873 Ack=71681 Win=65024 Len=0 SLE=71017 SRE=71681
1332 311.520940 192.168.1.6 192.168.1.9 TCP 2974 50005 → 63337 [ACK] Seq=71681 Ack=47873 Win=186880 Len=2920
1333 311.520984 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=47873 Ack=74601 Win=65536 Len=0
1334 311.522622 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=74601 Ack=47873 Win=186880 Len=664
1335 311.522662 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=47873 Ack=75265 Win=65024 Len=0
1336 311.539990 192.168.1.9 192.168.1.6 TCP 1514 63337 → 50005 [PSH, ACK] Seq=47873 Ack=75265 Win=65024 Len=1460 [TCP PDU reassembled in 1401]
1337 311.540687 192.168.1.9 192.168.1.6 TCP 1410 63337 → 50005 [PSH, ACK] Seq=49333 Ack=75265 Win=65024 Len=1356 [TCP PDU reassembled in 1401]
1338 311.545321 192.168.1.6 192.168.1.9 TCP 54 50005 → 63337 [ACK] Seq=75265 Ack=50689 Win=192768 Len=0
Frame 54968: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface
Ethernet II, Src: Intel_a1:89:2f (34:f6:4b:a1:89:2f), Dst: fa:ac:bd:2d:7b:2d
Internet Protocol Version 4, Src: 192.168.1.9, Dst: 192.168.1.6
Transmission Control Protocol, Src Port: 63337, Dst Port: 50005, Seq: 26864641, Ack:
Frame (frame), 54 bytes
Packets: 59890 · Displayed: 52197 (87.2%)
Profile: Default
Type here to search
17°C صاف غالبا
ENG
11:44 PM
12/9/2024



Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
3235 322.216210 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=1014273 Ack=991233 Win=1038336 Len=1460 [TCP PDU reassembled in 3404]
3236 322.216486 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=1015733 Ack=991233 Win=1038336 Len=1460 [TCP PDU reassembled in 3404]
3237 322.216526 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=991233 Ack=1017193 Win=65536 Len=0
3238 322.217187 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=1017193 Ack=991233 Win=1038336 Len=664 [TCP PDU reassembled in 340…
3239 322.228495 192.168.1.9 192.168.1.6 TCP 1514 63337 → 50005 [PSH, ACK] Seq=991233 Ack=1017857 Win=65024 Len=1460
3240 322.228802 192.168.1.9 192.168.1.6 TCP 1410 63337 → 50005 [PSH, ACK] Seq=992693 Ack=1017857 Win=65024 Len=1356
3241 322.236150 192.168.1.6 192.168.1.9 TCP 54 50005 → 63337 [ACK] Seq=1017857 Ack=994049 Win=1039360 Len=0
3242 322.256357 192.168.1.6 192.168.1.9 TCP 2974 50005 → 63337 [ACK] Seq=1017857 Ack=994049 Win=1039360 Len=2920 [TCP PDU reassembled in 3404]
3243 322.256386 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=994049 Ack=1020777 Win=65536 Len=0
3244 322.256860 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=1020777 Ack=994049 Win=1039360 Len=664 [TCP PDU reassembled in 340…
3245 322.259481 192.168.1.9 192.168.1.6 TCP 1514 63337 → 50005 [PSH, ACK] Seq=994049 Ack=1021441 Win=65024 Len=1460
3246 322.259564 192.168.1.9 192.168.1.6 TCP 1410 63337 → 50005 [PSH, ACK] Seq=995509 Ack=1021441 Win=65024 Len=1356
3247 322.276897 192.168.1.6 192.168.1.9 TCP 54 50005 → 63337 [ACK] Seq=1021441 Ack=996865 Win=1039360 Len=0
3248 322.292594 192.168.1.9 192.168.1.6 FMTP 1514 Operator message
3249 322.293021 192.168.1.9 192.168.1.6 TCP 1410 63337 → 50005 [PSH, ACK] Seq=998325 Ack=1021441 Win=65024 Len=1356 [TCP PDU reassembled in 3385]
3250 322.296578 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=1021441 Ack=996865 Win=1039360 Len=1460 [TCP PDU reassembled in 3404]
3251 322.296612 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=999681 Ack=1022901 Win=65536 Len=0
3252 322.298707 192.168.1.6 192.168.1.9 TCP 1514 50005 → 63337 [ACK] Seq=1022901 Ack=996865 Win=1039360 Len=1460 [TCP PDU reassembled in 3404]
3253 322.298707 192.168.1.6 192.168.1.9 TCP 718 50005 → 63337 [PSH, ACK] Seq=1024361 Ack=996865 Win=1039360 Len=664 [TCP PDU reassembled in 340…
3254 322.298747 192.168.1.9 192.168.1.6 TCP 54 63337 → 50005 [ACK] Seq=999681 Ack=1025025 Win=65536 Len=0
3255 322.316278 192.168.1.6 192.168.1.9 TCP 54 50005 → 63337 [ACK] Seq=1025025 Ack=999681 Win=1040640 Len=0
Frame 2896: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface
Ethernet II, Src: TpLinkTechno_2c:c7:fc (cc:32:e5:2c:c7:fc), Dst: Intel_a1:89:2f
Internet Protocol Version 6, Src: fe80::ce32:e5ff:fe2c:c7fc, Dst: ff02::1
Internet Control Message Protocol v6
Frame (frame), 78 bytes
Packets: 27781
Profile: Default
Type here to search
17°C
ENG
11:36 PM
12/9/2024

How the Code Works
Server-Client Communication Setup:

Server: Creates a Server Socket on a specific port (50005) and waits for a client to connect.Once a client connects, it proceeds to handle communication.

Client: Connects to the server using its IP address and port (50005).

Audio Capture and Playback: Uses Audio Record to capture audio from the microphone. Uses Audio Track to play back received audio. Both are configured to use mono channels, a 44100 Hz sample rate, and PCM 16-bit encoding.

Audio Data Transmission:

Sending: Captured audio is read into a buffer and sent over the socket using the server-client connection.

Receiving: Audio data received from the socket is written to Audio Track for playback.

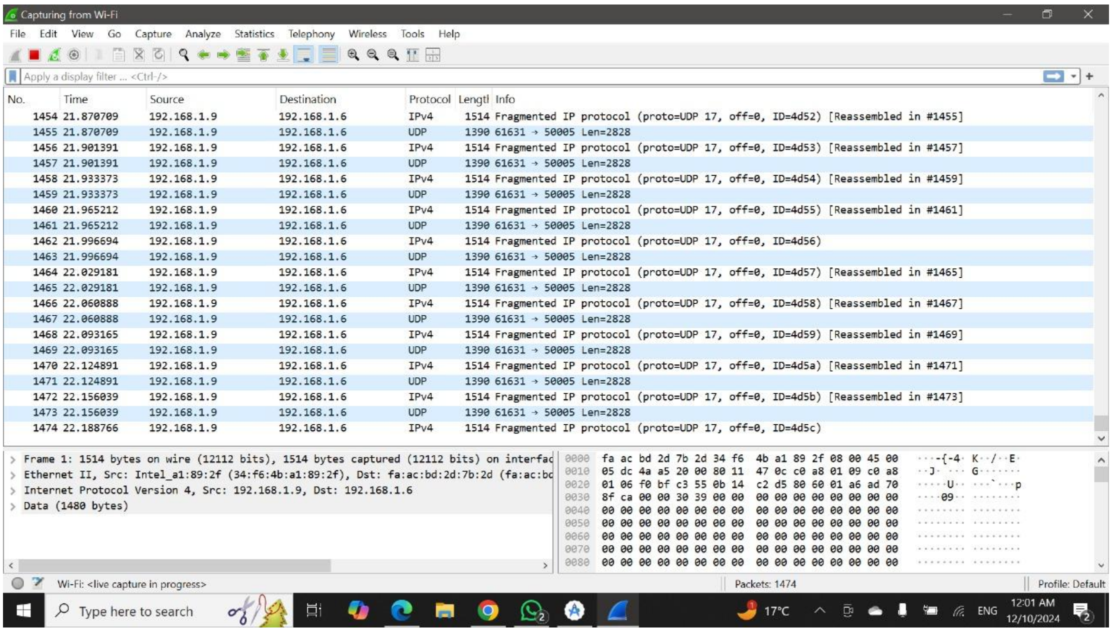
Threads for Parallel Processing: Separate threads handle sending and receiving audio data concurrently, ensuring continuous audio streaming.

Why It Doesn’t Use SIP or RTP

No SIP Implementation:SIP is not used in this code because it doesn't perform any session initiation, modification, or termination signaling. The connection is directly established using raw sockets without a signaling protocol.

No RTP for Media Transport: RTP is a protocol designed for real-time audio/video streaming. However, the code uses raw TCP sockets to transfer audio data, which is simpler but less optimized for real-time communication compared to RTP.

Missing Protocol Handling: There is no SDP (Session Description Protocol) for negotiating media parameters, which is essential for RTP. RTP usually operates over UDP for lower latency, but this code uses TCP, which is less suited for real-time communication due to potential delays from re-transmissions.



Relationship Between RTP and UDP: RTP relies on UDP as its transport layer to provide real-time delivery of data.The additional RTP header (12 bytes minimum) gives context and management features for real-time media, which plain UDP lacks.

RTP over UDP in Practice: RTP packets are encapsulated inside UDP datagrams. UDP provides basic delivery, while RTP adds context for the media. Typically, RTP works alongside RTCP (RTP Control Protocol), which helps monitor and manage the RTP stream (e.g., reporting quality, handling synchronization).

Comparison between TCP and UDP :

1. Reliability

TCP (our Code): TCP is a connection-oriented protocol. It ensures reliable delivery of data through mechanisms like acknowledgments, re-transmissions, and sequencing. Data packets arrive in order, and missing or corrupted packets are resent. Suitable for applications where accuracy is critical, e.g., file transfers or web browsing.

UDP: UDP is a connection-less protocol. It does not guarantee the delivery, order, or integrity of data packets. Packets may be lost, arrive out of order, or be duplicated. Suitable for applications where speed is more critical than reliability, e.g., live audio/video streaming or gaming.

2. Latency

TCP: Higher latency because of connection establishment, acknowledgments, and re-transmission of lost packets. Suitable for scenarios where reliability outweighs speed, such as in the code example provided.

UDP: Lower latency since there is no connection setup or acknowledgment overhead. Ideal for real-time applications like voice or video calls where minor data loss is acceptable to maintain speed.

3. Data Transmission

TCP: Data is sent as a continuous stream, and the protocol ensures that the receiver gets the data in the correct order. Overhead from headers, handshakes, and re-transmissions can make TCP slower for real-time applications.

UDP: Data is sent in discrete packets (datagrams).
No re-transmissions or guarantees, which makes it faster for time-sensitive tasks.