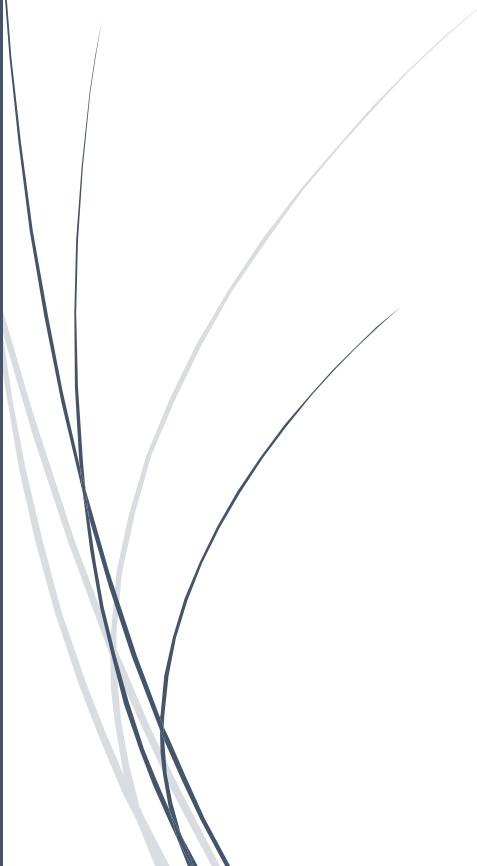




Signaling Report

Network scanning and sniffing



Kareen Raafat 55-25390
Khaled Wael 55-6548
Ganna Reda 55-20722
Hana Ahmed 55-5384
Farah Miniesy 55-2057
Rawan Miniesy 55-24969

Network Scanning:

A Scan of the private network to find the live hosts using the Ping command.

No.	Time	Source	Destination	Protocol	Length Info
380	11.766542	192.168.1.20	192.168.1.5	ICMP	74 Echo (ping) request id=0x0001, seq=27/6912, ttl=128 (no response found!)
493	16.330863	192.168.1.20	192.168.1.5	ICMP	74 Echo (ping) request id=0x0001, seq=28/7168, ttl=128 (reply in 495)
495	16.378003	192.168.1.5	192.168.1.20	ICMP	74 Echo (ping) reply id=0x0001, seq=28/7168, ttl=64 (request in 493)
532	17.361834	192.168.1.20	192.168.1.5	ICMP	74 Echo (ping) request id=0x0001, seq=29/7424, ttl=128 (reply in 533)
533	17.388441	192.168.1.5	192.168.1.20	ICMP	74 Echo (ping) reply id=0x0001, seq=29/7424, ttl=64 (request in 532)
542	18.375510	192.168.1.20	192.168.1.5	ICMP	74 Echo (ping) request id=0x0001, seq=30/7680, ttl=128 (reply in 543)
543	18.379483	192.168.1.5	192.168.1.20	ICMP	74 Echo (ping) reply id=0x0001, seq=30/7680, ttl=64 (request in 542)

Network Sniffing:

Wire shark Sniff the TCP packets during the call.

Closing Packets:

4011	25.379130	192.168.1.5	192.168.1.20	TCP	1514 Ignored Unknown Record
4012	25.379130	192.168.1.5	192.168.1.20	TCP	1514 Ignored Unknown Record
4013	25.379130	192.168.1.5	192.168.1.20	TCP	718 Ignored Unknown Record
4014	25.379193	192.168.1.20	192.168.1.5	TCP	54 53887 + 58828 [RST, ACK] Seq=1725713 Ack=1711043 Win=0 Len=0

Call Started:



Initiating:

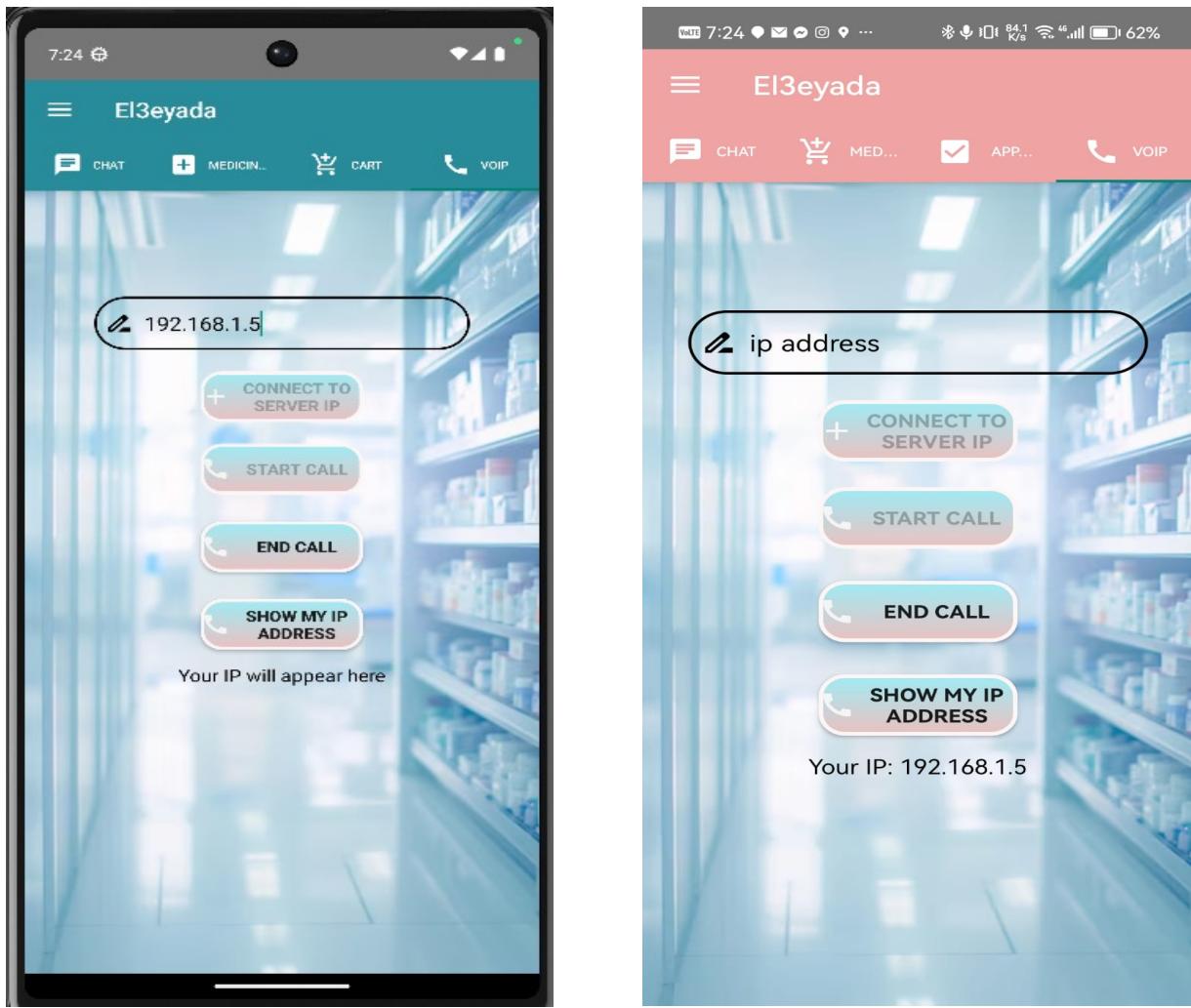
27	5.256048	192.168.1.20	192.168.1.5	TCP	66 54637 + 58828 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM
28	5.257292	192.168.1.20	192.168.1.5	ECHO	123 Request
29	5.435051	192.168.1.5	192.168.1.20	TCP	66 58828 + 54637 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 SACK_PERM WS=512
30	5.435173	192.168.1.20	192.168.1.5	TCP	54 54637 + 58828 [ACK] Seq=1 Ack=1 Win=65280 Len=0

Call Ended:

.....!%.(.)&.".....
.....&.-.+%.%.).+.(.&%.\$.#.#.%%.(..,2.7.5.0.,*.,.,.'.....
.....#.#.".|.!.....\$.)*.).'..(.*.*.*.).*.).%.....".#.".
.....'.[REDACTED]
.....!.\$.&).-.-.,.,-./.1.3.0.,.*'.%%.%.(..)+.../.1.2./././-.*%.
.....\$.)+.+.)(.%.!.!.%.+.....-.+.*.&!.....\$.*.,.,.+.*.(\$.\$.%.\$.!.....
.....'.[REDACTED]
.....".\$.'-.-1.3.4.3.1.+.).+...1.5.7.7.7.8.8.8.6.5.4.2.0./.....-.+.'.".....
.....".8.>@.<.7./.&.....
.....\$.-8.B.I.J.D.;./(.".[REDACTED]
.....#.%.&.+./2.2.0..../.1.3.1././.,'.
.....".#.%.%.#.%.%.%.(..,././.*.#.1..!\$.)...1.6.8.8.8.7.5.7.8.8.8.4./.+.'.%.\$
.....\$.\$.\$.%.%.\$.!
.....%,2.3./.*#.+.:H.P.R.Q.K.C.9./\$.
.....!.\$.%.&(..,1.6.;@.C.C.@.;7.3.2.2.2./.,*%.[REDACTED]
.....!.#.%.%.%.\$.#!.!#.!.&.,2.7.9.:9.8.8.7.8.8.7.5.3.2.2.1./....1.5.5.3./.+.&%.\$.%.%\$.!
.....\$.6.;<.8.2.).". .!.
#.1.>F.G.C.;5.3.5.7.6.4./.).\$.!.(..,1.6.6.4.1./.3.8.<.>.=9.7.7.7.8.8.8.7.2.)
.....[REDACTED]

.....CTRL:CALL_ENDED

IP address of live host:



30 5.435173	192.168.1.20	192.168.1.5	TCP	54 54637 → 50020 [ACK] Seq=1 Ack=1 Win=65536 Len=0
32 5.447097	192.168.1.20	192.168.1.20	TCP	68 50020 → 54637 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=0
33 5.500629	192.168.1.20	192.168.1.5	TCP	54 54637 → 50020 [ACK] Seq=1 Ack=15 Win=65280 Len=0
34 5.584196	192.168.1.20	192.168.1.5	TCP	1514 54637 → 50020 [PSH, ACK] Seq=1 Ack=15 Win=65280 Len=1460
35 5.584280	192.168.1.20	192.168.1.5	TCP	1282 54637 → 50020 [PSH, ACK] Seq=1461 Ack=15 Win=65280 Len=1228
36 5.589081	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=15 Ack=1461 Win=68608 Len=0
37 5.589081	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=15 Ack=2689 Win=1680 Len=0
38 5.594479	192.168.1.20	192.168.1.5	TCP	1514 54637 → 50020 [PSH, ACK] Seq=2689 Ack=15 Win=65280 Len=1460
39 5.594826	192.168.1.20	192.168.1.5	TCP	1282 54637 → 50020 [PSH, ACK] Seq=4149 Ack=15 Win=65280 Len=1228
40 5.597648	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=15 Ack=4149 Win=74752 Len=0
41 5.597648	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=15 Ack=5377 Win=77312 Len=0
42 5.605983	192.168.1.20	192.168.1.5	TCP	1514 54637 → 50020 [PSH, ACK] Seq=5377 Ack=15 Win=65280 Len=1460
43 5.606064	192.168.1.20	192.168.1.5	TCP	1282 54637 → 50020 [PSH, ACK] Seq=6837 Ack=15 Win=65280 Len=1228
44 5.609514	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=15 Ack=6837 Win=80384 Len=0
45 5.609514	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=15 Ack=8065 Win=83456 Len=0
46 5.633372	192.168.1.20	192.168.1.5	TCP	1514 54637 → 50020 [PSH, ACK] Seq=8065 Ack=15 Win=65280 Len=1460
47 5.633406	192.168.1.20	192.168.1.5	TCP	1282 54637 → 50020 [PSH, ACK] Seq=9525 Ack=15 Win=65280 Len=1228
48 5.636807	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=15 Ack=9525 Win=86016 Len=0
49 5.636807	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=15 Ack=10753 Win=89088 Len=0
50 5.665207	192.168.1.20	192.168.1.5	TCP	1514 54637 → 50020 [PSH, ACK] Seq=10753 Ack=15 Win=65280 Len=1460
51 5.665288	192.168.1.20	192.168.1.5	EMTP	1282 [openfor message]

Sequence Number (raw): 2107765737
[Next Sequence Number: 15 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 932061568
0101 = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window: 128
[Calculated window size: 65536]
[Window size scaling factor: 512]
Checksum: 0xd268 [Unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[[Timestamps]]
[[SQ/ACK analysis]]
TCP payload (14 bytes)

0000 c4 75 ab ba cc ad 62 43 bb 52 b8 76 08 04 00 u... bC R v E.
0010 00 36 14 e8 40 00 40 06 20 c0 a8 01 05 c0 a8 6 ..@. p
0020 01 14 c3 d4 d5 6d 7d a1 f3 e9 37 Be 21 78 50 18 ...d m... -7 !xp.
0030 00 80 d2 60 00 00 43 54 52 4c 3a 43 4f 4e 45 CT RL:CONN
0040 43 54 45 44 CTE

2458 17.704582	192.168.1.5	192.168.1.20	TCP	1514 50020 → 54637 [ACK] Seq=1055171 Ack=1072513 Win=453632 Len=1460
2459 17.704582	192.168.1.5	192.168.1.20	TCP	718 50020 → 54637 [PSH, ACK] Seq=1056631 Ack=1072513 Win=453632 Len=64
2460 17.704572	192.168.1.20	192.168.1.5	TCP	54 54637 → 50020 [ACK] Seq=1072513 Ack=1057295 Win=65280 Len=0
2461 17.705563	192.168.1.20	192.168.1.5	TCP	1514 54637 → 50020 [PSH, ACK] Seq=1072513 Ack=1057295 Win=65280 Len=1460
2462 17.705722	192.168.1.20	192.168.1.5	TCP	1282 54637 → 50020 [PSH, ACK] Seq=1073973 Ack=1057295 Win=65280 Len=1228
2463 17.716354	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=1057295 Ack=1075201 Win=453632 Len=0
2464 17.733664	192.168.1.20	192.168.1.5	TCP	1514 54637 → 50020 [PSH, ACK] Seq=1075201 Ack=1057295 Win=65280 Len=1460
2465 17.733984	192.168.1.20	192.168.1.5	TCP	1282 54637 → 50020 [PSH, ACK] Seq=1076661 Ack=1057295 Win=65280 Len=1228
2466 17.734789	192.168.1.20	192.168.1.5	TCP	69 54637 → 50020 [PSH, ACK] Seq=1077889 Ack=1057295 Win=65280 Len=15
2467 17.736411	192.168.1.20	192.168.1.5	TCP	54 54637 → 50020 [PSH, ACK] Seq=1077904 Ack=1087205 Win=453632 Len=0
2468 17.737548	192.168.1.5	192.168.1.20	TCP	54 50020 → 54637 [ACK] Seq=1057295 Ack=1077889 Win=453632 Len=0
2469 17.748478	192.168.1.5	192.168.1.20	TCP	1514 50020 → 54637 [ACK] Seq=1057295 Ack=1077905 Win=453632 Len=1460
2470 17.748478	192.168.1.5	192.168.1.20	TCP	1514 50020 → 54637 [ACK] Seq=1058755 Ack=1077905 Win=453632 Len=1460
2471 17.748478	192.168.1.5	192.168.1.20	TCP	718 50020 → 54637 [ACK] Seq=1060215 Ack=1077905 Win=453632 Len=64
2472 17.748561	192.168.1.20	192.168.1.5	TCP	54 54637 → 50020 [ACK] Seq=1077905 Ack=1060879 Win=65280 Len=0
2473 17.790078	192.168.1.5	192.168.1.20	TCP	1514 50020 → 54637 [ACK] Seq=1062339 Ack=1077905 Win=453632 Len=1460
2474 17.790078	192.168.1.5	192.168.1.20	TCP	718 50020 → 54637 [ACK] Seq=1063799 Ack=1077905 Win=453632 Len=64
2475 17.790078	192.168.1.5	192.168.1.20	TCP	54 54637 → 50020 [PSH, ACK] Seq=1077905 Ack=1062339 Win=0 Len=0
2476 17.790158	192.168.1.20	192.168.1.5	TCP	END

Sequence Number: 1077889 (relative sequence number)
Sequence Number (raw): 933139448
[Next Sequence Number: 1077904 (relative sequence number)]
Acknowledgment Number: 1057295 (relative ack number)
Acknowledgment number (raw): 2108823031
0101 = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window: 255
[Calculated window size: 65280]
[Window size scaling factor: 256]
Checksum: 0x8393 [Unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
TCP payload (15 bytes)
TCP segment data (15 bytes)

Reasons why RTP doesn't appear:

SIP and RTP protocols are not being used explicitly; instead, the application relies on custom communication over TCP (port 50020). Below are the reasons why Wireshark does not show SIP and RTP packets:

Absence of SIP Implementation

SIP (Session Initiation Protocol) is a signaling protocol used to establish, manage, and terminate VoIP calls. In the current implementation, SIP is not utilized for signaling purposes. Instead, custom control messages (e.g., CTRL:CONNECTED) are sent over TCP.

Typically, SIP messages are transmitted on ports like 5060 (default for SIP), but since SIP protocols or libraries are not incorporated, Wireshark cannot detect SIP packets.

Lack of RTP for Media Transmission

RTP (Real-time Transport Protocol) is the standard for transmitting audio and video streams in VoIP communications. The application uses raw socket communication and Android's Audio Record and Audio Track APIs to handle audio data.

Unlike RTP, which operates over UDP, this implementation transmits audio over TCP. Since it does not adhere to RTP standards, Wireshark does not recognize these packets as RTP.

Packet Flow During the VoIP Session:

1. Connection Setup

- The server starts by listening for incoming TCP connections on port 50020.**
When the client initiates a connection using the server's IP address and port, a TCP handshake occurs, establishing the communication channel.

2. Control Messages

- After the TCP connection is established, the server sends a control message (CTRL:CONNECTED) to the client to confirm the connection.**
- Throughout the session, additional control messages such as CTRL:CALL_ENDED may be exchanged to signal events like call termination whether the user choose to terminate it or any other reason that lead to call termination.**

3. Audio Data Transmission

- Capture and Transmission:**
The client's microphone captures audio using the Audio Record API, and the data is read into a buffer. The buffered audio data is then sent over the established TCP connection to the server in real-time.
- Reception and Playback:**
On the server side, the audio data is received from the TCP connection, buffered, and played back using the Audio Track API.

4. Unidirectional/Bidirectional Flow

- The packet flow for audio data is bidirectional: both the client and server send and receive audio packets simultaneously to enable real-time communication.**
- However, these packets are transmitted as raw byte streams over TCP, without using a standardized media transport protocol like RTP.**

5. Call Termination

- When either the client or server decides to end the session, a control message (CTRL:CALL_ENDED) is sent to the other party, signaling the termination of the call.**
- The TCP connection is then gracefully closed by both sides.**