COTS DOCUMENTATION

MakeMyTechnology
VTU Regional Center,1st Main Rd, RHCS Layout, Annapoorneshwari Nagar,
Naagarabhaavi, Bengaluru, Karnataka 560091

Contents

1. Introduction

Welcome to the MakeMyTechnology (MMT) COTS & Multi-Mobile Server User Manual. This manual is made to help you understand and use the COTS Platform easily and effectively. Whether you are using it for the first time or are already familiar with it, this guide will walk you through every part of the system from setup and operation to troubleshooting. It explains the main features, how to view live network data, and how to make the best use of COTS for your testing and research work.

1.1. About MakeMyTechnology (MMT)

MMT is a deep-tech startup specializing in Made-in-India 5G infrastructure and training solutions. We design and manufacture 5G cores, gNBs, multi-mobile testing tools (COTS), and LoRa-based connectivity devices, and partner with colleges and universities to set up fully functional 5G labs. Our programs are hands-on and project oriented, allowing students and faculty to gain practical exposure to 5G technologies, network testing, and lab operations, bridging the gap between theory and real world applications.

1.2. Purpose of this Manual

This manual provides comprehensive, step-by-step guidance for beginners including faculty members, students, and interns to effectively install, configure, and operate the **COTS and Multi-Mobile Server application suite** developed by **MakeMyTechnology (MMT)**. It is designed to help users understand key COTS concepts, perform system setup, manage server operations, monitor data flows, review logs, and troubleshoot common issues easily.

2. Overview

2.1. What is COTS?

COTS (Commercial Off-The-Shelf) is an Indian-made 5G/4G wireless logging and analysis tool. It allows you to gather network logs from standard Android smartphones and visualize key KPIs such as RSRP, RSRQ, SINR, throughput, and more. The tool supports automation for testing and exporting logs for analysis.

• Simple: One-click actions, built for beginners.

• Affordable: Suitable for educational institutions.

• Educational: Real data for learning wireless network concepts.

2.2. What is the Multi-Mobile Server?

The Multi-Mobile Server is a lightweight HTTP server running on Windows Laptop/PC.

It exposes a small set of HTTP end points that:

• Stream live device stats (signal strength, cell info, GPS).

• Responds to automation commands (start/stop ping, drive test logging, etc.).

• Allow your browser dashboard (EMS) to fetch and display data from one or many phones.

3. System Architecture

3.1. Components

• APK (Phone Side): Android app containing the Multi-Mobile Server (Ktor server).

• Dashboard (PC Side): EMS web dashboard based on Flask for metric display and log management.

3.2. Data Flow

Phones (with the APK and server) collect sensor/modem data and transmit it via Wi-Fi, hotspot, or USB tethering to a PC dashboard, where it is visualized and logged.

4. Features

• Live KPIs: View RSRP, RSRQ, SINR, PCI, bandwidth, TAC, and more.

• Location-aware testing: Metrics plotted against time and location for drive/walk exercises.

• Automation: Control ping/iperf tests for latency and throughput checking.

• Multi-device support: Monitor multiple phones simultaneously.

• Log export: Download PCAP, DLF, or CSV files for offline analysis.

5. System Requirements

Component	**Requirements**
Phone Side (APK)	● Android 14 ● Location and network state permissions ● Root access (for logs)
PC/Laptop Side (EMS Dashboard)	● Python 3.12+ ● Java JDK ● On same network as phones ● Browser (Chrome/Edge/Firefox)

6. Installation & Setup

6.1. Phone Side (APK)

- Install the APK on your Android test phone(s).
- Open the app and grant required permissions.
- Ensure phone and PC are on the same network
- Note the IP address of the mobile (UE) that you can see in the notification

6.2. PC/Laptop Side (EMS Dashboard)

- Download the Provided installer file.
- Open the file then it will open Cots EMS interface.
- In imsi_map section You need to add your testing phone ip address that you have saved earlier.
- Save the IP address than Start the Server.

7. Server Endpoints and Data Model

7.1. Health & Utility

Endpoint	Description
GET /ping	Returns: 'ping' (to verify server is running).
GET /device_id	Returns device ID.

7.2. RF & Telephony Info

Endpoint	Description
GET /mobile_stats	Returns RF stats (RSRP , RSRQ, SNR, etc.).
GET /cell_info	Returns Current cell information.
GET/neighbor_cell_info	Returns neighboring cell info.
GET/rf_info	Returns signal RF details (ssRsrp, ssRsrq, ssSinr).

7.3. Logging and Streaming

Endpoint	Description
GET /start_udp_log_stream? host_ip=IP&port=PORT	Starts log stream to specified IP/port.
GET /stop_udp_log_stream	Stops UDP log stream.
GET /write_logs_to_dlf_file	Begins saving logs to a DLF file.
GET/stop_write_logs_to_dlf_file	Stops DLF logging.
GET/download_dlf	Downloads the DLF log file.
POST /enable_logs	Enables logging for specified log types (hex codes).

7.4. Diagnostics

Endpoint	Description
POST /runAtCommand	Runs an AT command on the device.

7.5. Drive Test

Endpoint	Description
POST /start_drive_test	Starts a drive test with JSON body: { 'interval': ms }
GET /stop_drive_test	Stops the current drive test.
GET /performance_metrics	Returns collected performance samples.

7.6. iPerf Testing

Endpoint	Description
POST /runIperf	Starts iPerf with arguments in body JSON: {'args': ['-s', '-u'] }.
GET /stopIperf	Stops running iPerf process.

7.7. VoIP /Call Management

Endpoint	Description
POST /make_call	Starts an audio/video call. Body JSON: { 'type': 'audio video', 'number': '... }
GET /recv_call? action=answer|reject|status |audio|video	Handles incoming call control or media simulation.

7.8. Ping Utility

Endpoint	Description
POST /startPing	Starts ping to a target. Body JSON: { 'target': 'hostname' }
GET /stopPing	Stops ping process.

7.9. Device Control

Endpoint	Description
GET /airplane_mode? action=enable disable	Enables or disables airplane mode.
GET /reboot_device	Reboots the device.

7.10. APK Upgrade

Endpoint	Description
POST /upgrade_apk	Uploads and installs an APK.
POST /upload_license	Uploads and stores a license file.

8. Connecting Devices to EMS Dashboard

- Open the installer file.
- Now you will see the GUI (interface) page. You will see the (fig 11.1)
- Add phone server URLs (http://phone-ip:8080) to the dashboard.
- Confirm live KPIs and device data appear in the browser.

9. Common Workflows

9.1. Simple Drive/Walk Test

- Enable GPS on the phone and mount or carry it.
- Start the drive test from the COTS dashboard.
- Walk while monitoring KPIs in real time.

9.2. Side-by-Side Operator Comparison

- Use two phones with different carriers/SIMs.
- Add both to the EMS dashboard and compare KPIs live.

9.3. Throughput Testing

- Start iperf3 from the dashboard.
- In this we have given two types of options: iperf-client: By using this feature we can check the throughput.

10. Log Files & Formats

File Type	Description	Location
CSV	Time-series KPIs (timestamp, GPS, RSRP, RSRQ, SINR, etc.)	Phone storage or PC logs folder.
PCAP/DLF	Raw protocol capture (for Wireshark, offline analysis)	Dependent on app config/permissions.

11. User Interface Snapshots

11.1. GUI Launcher Page:

1. Here you can see the COTS EMS YAML page. On that page you need to modify the IP address with your mobile IP address and save the file.
2. There are 4 buttons you can see in the fig 11.1:
    - Start button
    - Stop button
    - Open in Browser
    - Clear console
3. After saving the YAML file, click on the start button then the multi mobile server will be started.
4. Next click on the open in Browser button then the multi-mobile screen will be shown in the browser fig 11.2

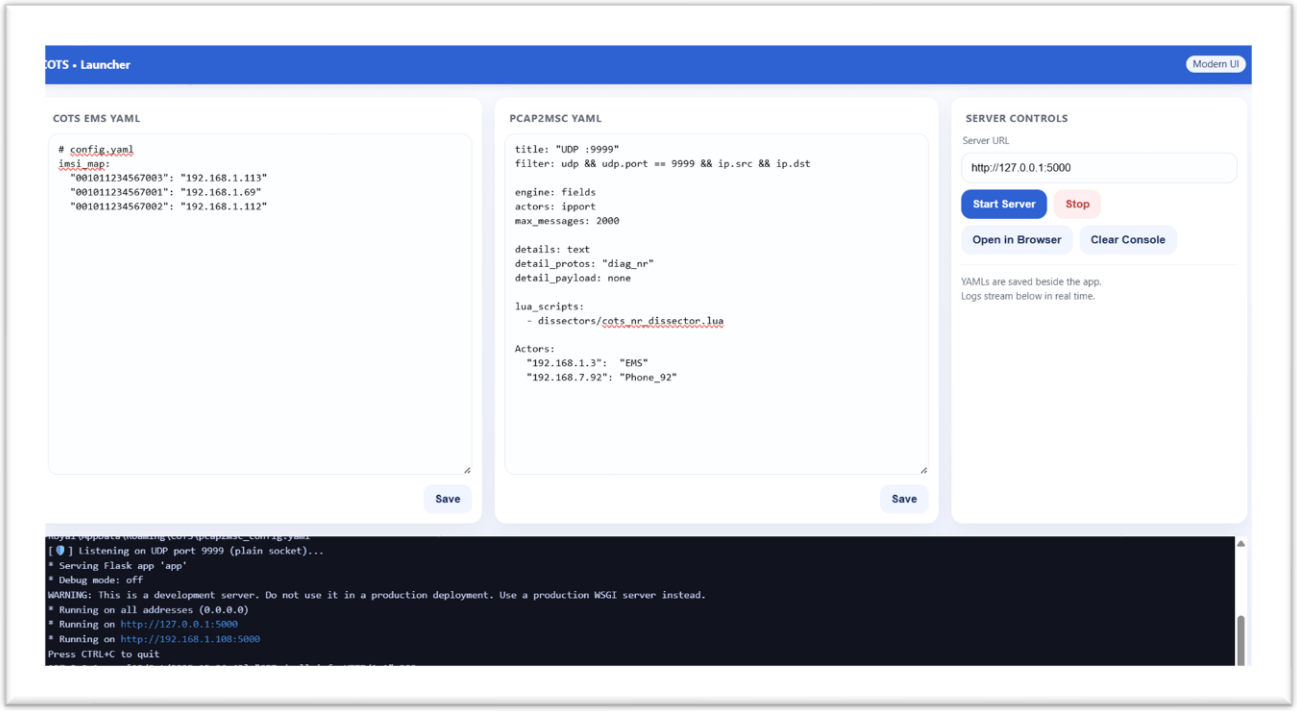


Fig 11.1

11.2. Multi-Mobile Server

- In this multi-mobile screen, you can see all mobiles information and we can also see how many mobiles are connected and there RSRP (Reference Signal Received Power), RSRQ (Reference Signal Received Quality) & SNR (Signal-to-Noise Ratio).
- Here you have 4 Features:
    - Cell information
    - Multi-Mobiles Dashboard
    - Robot Scripts
    - Log viewer
- Cell information: You can see the all serving cell information. Like PCI, ARFCN, TAC, Band, MCC, MNC, NCI, Operator Name.
- Mobiles Dashboard: Here you can see the values of the RSRP, RSRQ & SNR values of the multiple mobiles.
    - Click on the Mobile App open button then the mobile (UE) web server will be shown in the new page which is Fig 11.3
    - Then go to the 5g Nr Log Configuration page and select the log types which you want to see and click on the submit button.
    - Come back to the multi-mobile server page which is Fig 11.2
    - Click on the open Wireshark button then, you can see the logs in two ways:
        - In live MSC (message sequence chart): Opens in the new page which is Fig 11.5.1

        - In Wireshark: Opens in the Wireshark which is Fig 11.5.2
    - If you wanted to save the live logs then you can save them in the .dlf format for saving the logs you need to click on the start button which is in the log files (DLF) section in mobiles Dashboard

- Robot scripts: It is used for the test Automation so we have some set of commands to run the robot scripts for now we have given you the three-robot script file.
    - if you wanted to see the script files, click on the Open Robot Scripts form there if you wanted you can download them directly.
    - If you want to execute the Robot Scripts, please select the downloaded robot script file and click on the submit button then it will execute the robot script file and give you the output report.html, log.html & output.xml files which is 11.6. These files will be stored in the app directory.
- Log viewer: This feature is helpful to see the downloaded logs from here.
    - Select the downloaded dlf file from this log viewer and click on submit, this will show the logs in two ways:
        - In MSC (Message Sequence Chart)
        - In Wireshark
    - If you deselect the open call-flow diagram (MSC) option in the Log Viewer then you see the logs only in Wireshark.

Fig 11.2

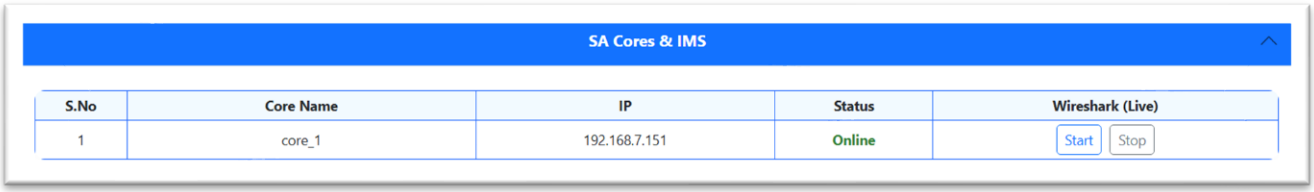


Fig 11.2.1

This image shows details related to the **5G Core Network** (also referred to as SA Core). It includes the core's name, the associated IP address, and the current status (which is "Online" here, meaning the core is active or connected). The image also displays the option to use **Wireshark** for live network monitoring. **Wireshark** is a powerful tool that allows network engineers to capture and inspect the data packets passing through the network, helping them diagnose and resolve connectivity or performance issues. By clicking on the "Start" button, engineers can begin monitoring the network traffic associated with the core. When the core is online, real-time network diagnostics can be conducted to ensure smooth operations.

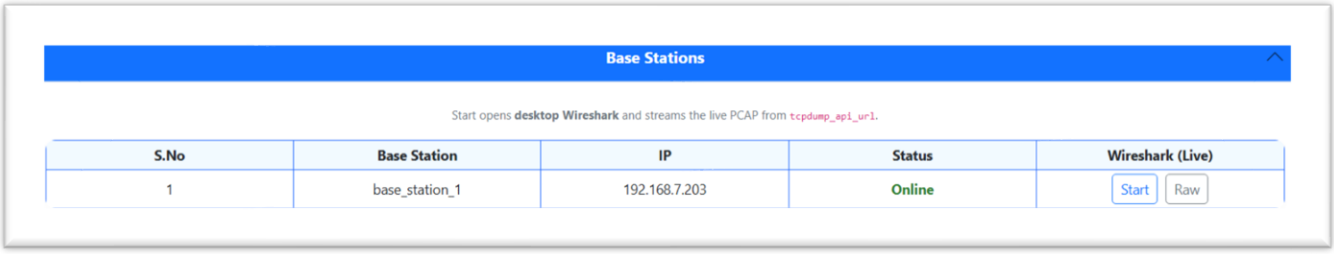


Fig 11.2.2

This image displays important details about a **Base Station** in the network. The **Base Station** is named **base_station_1** and has an assigned **IP address** (192.168.7.203). The **Status of Base Station** is currently **Online**, meaning it is actively serving users or processing network data. The image also includes the option to use **Wireshark** for live packet capture from the base station. **Wireshark** is a tool used by network engineers to capture, inspect, and analyze network traffic in real-time. By clicking the **Start** button, engineers can begin monitoring the data flowing through the base station, which helps in troubleshooting and performance analysis. The **Raw** button provides access to the detailed packet data for advanced users looking for deeper insights into the traffic.

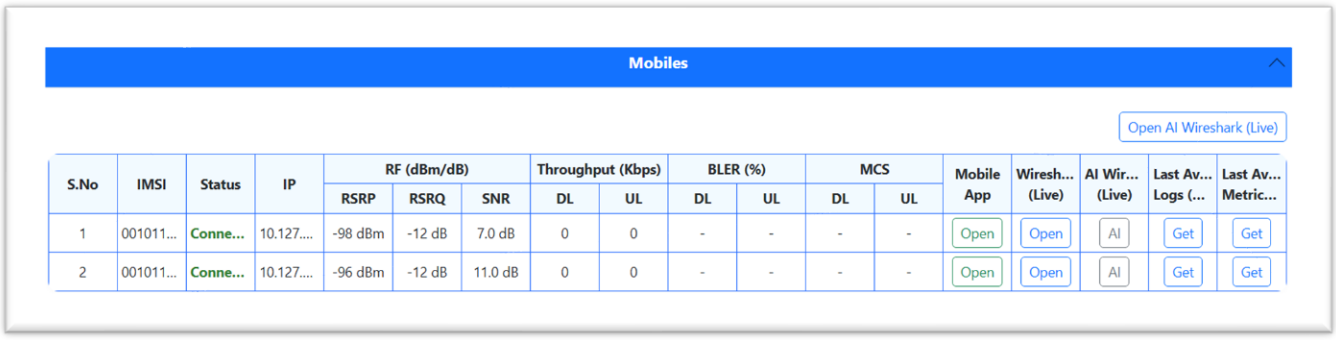


Fig 11.2.3

This image displays the current network performance of multiple mobile devices. It provides details such as the **IMSI**, which uniquely identifies each device, and their connection **status** (both are **Connected**). The **IP** address helps identify each device in the network. The **RF (Radio Frequency)** parameters show the signal quality for both devices, including **RSRP**, **RSRQ**, and **SNR**, which give insight into signal strength and quality. **Throughput** indicates the download and upload speeds, and **BLER** shows the error rate in the transmission. **MCS** values provide information about the data transmission efficiency. The image also allows users to interact with the **Wireshark (Live)** and **AI Wireshark (Live)** buttons to monitor real-time traffic from the devices, and access the last average logs and metrics for performance analysis.

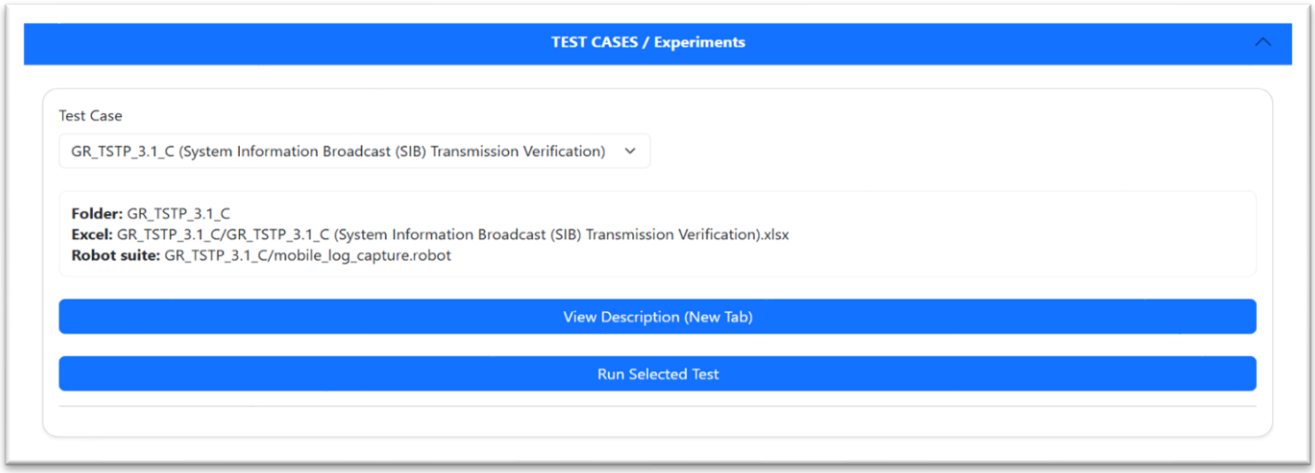
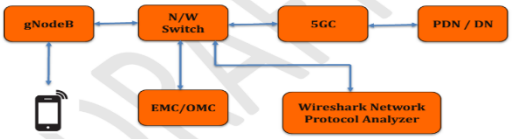
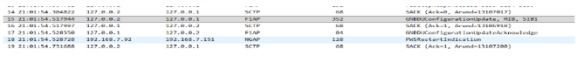


Fig 11.2.4

This image shows the interface for users to select and run 5G-related test cases for verifying system information broadcasts (SIB) in mobile networks. The selected test case will use a corresponding Excel file and Robot Framework script to automate the testing process. Users can either view more details about the test case or run the test directly by clicking the appropriate buttons. This tool is essential for automating and streamlining network verification processes during testing.

Field	Content
Test No.	GR_TSTP_3.1_C (System Information Broadcast (SIB) Transmission Verification)
Test Details	Verify that gNB shall support System Information Broadcast (SIB).
Test Instruments Required	Test gNodeB, RF cables and attenuators, 5GC/ 5GC Emulator, UE /UE Instruments simulator, Network Performance Measurement Tool, Wireshark Required Network Protocol Analyser
Test Setup	TEST SETUP 2
Test Procedure	1. Configure gNodeB with system information Broadcast parameters. Ensure that the gNodeB is operationally enabled. 2. Power on a user equipment 3. Verify through UE's reception of System Information (SIB). 4. Capture the control plane signaling using a tool like Wireshark to analyze the broadcast of SIBs that the gNodeB is transmitting configured system information.
Test Limits	NA
Expected Results	The gNodeB shall broadcast system information as per
STP_3.1_C (System Information Broadcast (SIB) Transmission Verification)	
Test Setup/Architecture	
Explanation for Test setup/Architecture	The gNodeB is configured with SIB broadcast parameters and connected to the 5GC via the N/W switch, while OMC and Wireshark monitor signaling. A UE powers on and receives MIB/SIB messages from the gNodeB, verifying that broadcast system information is transmitted correctly. Wireshark captures the control-plane signaling (NGAP/SCTP/F1AP) to confirm that the gNodeB is broadcasting the configured system information.
Sample Logs	
Explanation for Wireshark Sample log	The logs show MIB/SIB related signaling such as GNB-DU Configuration Update, Acknowledge, and NGAP Restart Indication, confirming system information configuration. SCTP exchanges demonstrate successful transport of broadcast-related control messages. These traces validate that the gNodeB is actively transmitting the configured SIB information as part of its system broadcast procedure.

Sample Test Case Description View

11.3. Mobile Side View

- Displays the all information of the mobile phone information, there are so many features in this mobile webserver they are:
    - **Serving cell information**: Shows all the cellular information of the mobile.
    - **RF information:** Shows the all RSRP, RSRQ & SNR Values of the mobile phone.
    - **NR 5G Log Code Configuration:** In this you can see all the Different types of 5G logs form the L1, L2 and L3 Layers.
    - **Iperf3:** Here you have the features for the iperf.
        - Iperf Client: Here you need to enter the server IP Address so check the throughput between the Client and server. Ex:- “-c 192.168.xx.xx -t 10 -i 1” (Fig 11.5.1)
        - Iperf Server: Here you can start a server from a mobile to start the server you need to enter the commands Ex:- “-s -p 5201” (Fig 11.5.2)
    - **Ping Test:** This feature is just for check the ping weather the destination is active or not and what is the speed of the destination address Ex:- “google.com” (Fig 11.6)
    - **Drive Test:** This feature is used for testing all cell and rf information throughout the area where we are going. Which is Fig 11.7
    - **AT command:** This feature is giving the access of the full phone to the users. For example, if any one wants to take full access to the mobile phone, then he will type the AT commands and take that permissions form the mobile phone. (Fig 11.8)
    - **RF Signal Quality chart:** In this you can see all the RF information like RSRP, RSRQ & SNR values in the chart format. Which is Fig 11.9

Fig 11.3

11.4. Types of 5G Logs View

- We can see the logs in 2 types:
    - In MSC (Message Sequence Chart)
    - In Wireshark

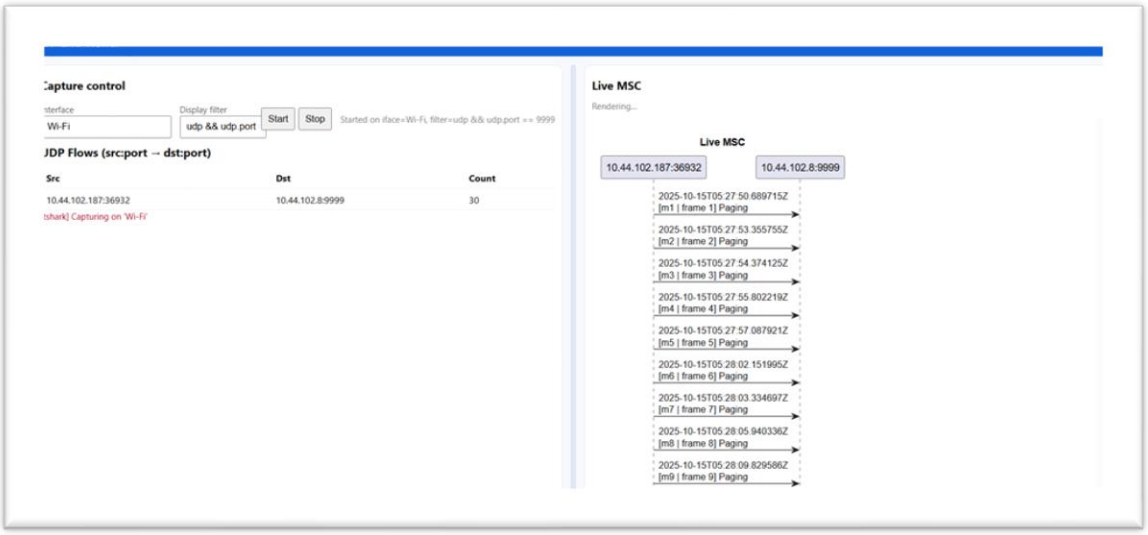


Fig 11.4.1 (MSC)

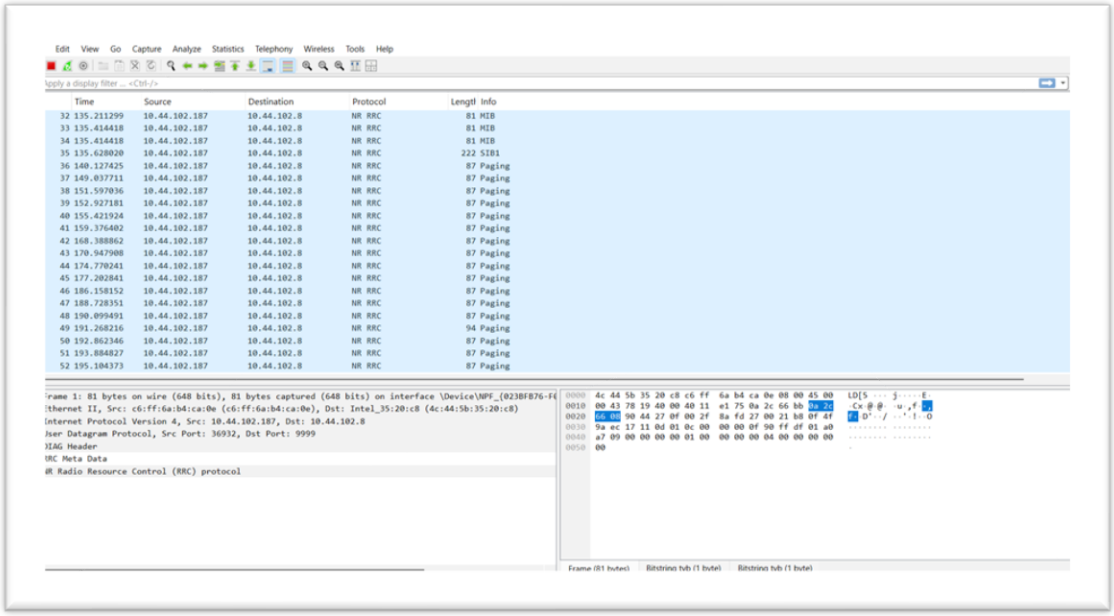


Fig 11.4.2 (Wireshark)

11.5. iPerf Client & Server Views

- Run and monitor network throughput using integrated iPerf controls from the UE Desktop.

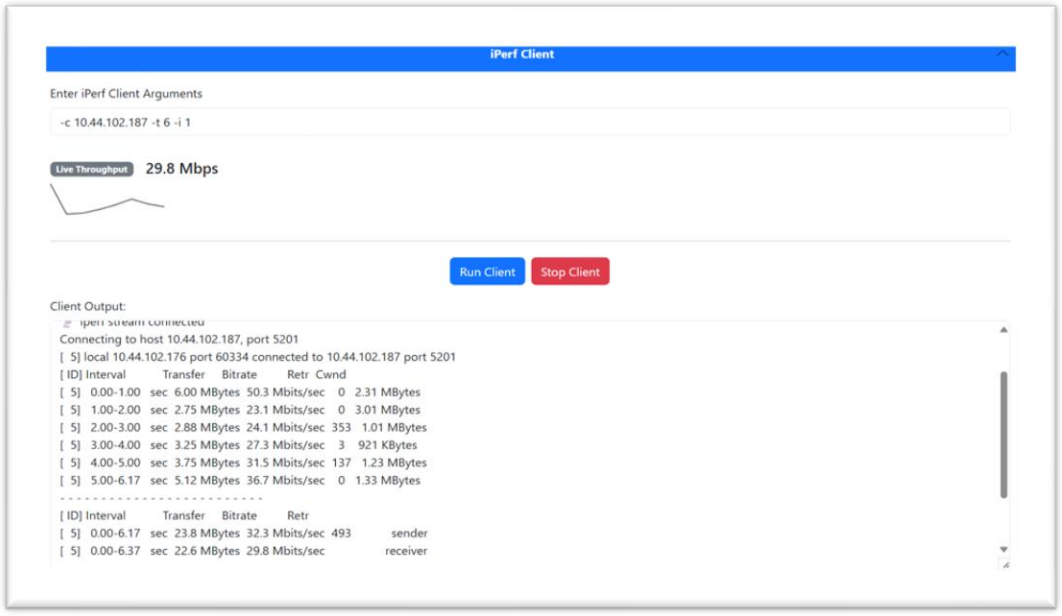


Fig 11.5.1 (Iperf Client)

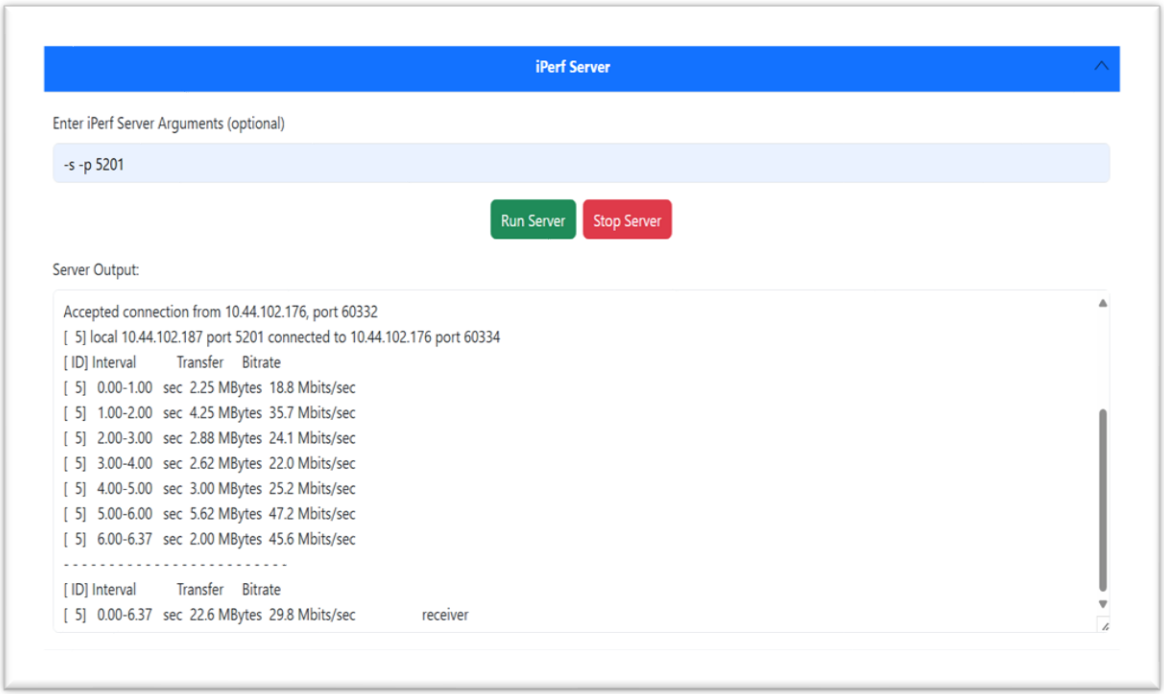


Fig 11.5.2 (Iperf Server)

11.6. Ping Test Views

- Quickly start, stop, and monitor ping test. It checks weather the host is available or not.
- You can ping for any hosts. Ex:- google.com (or) yahoo.com

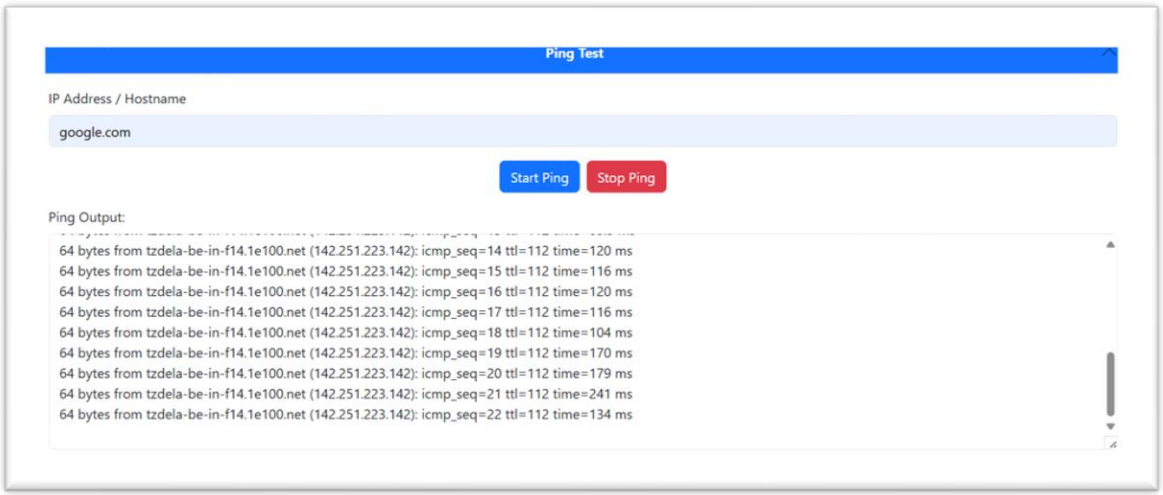


Fig 11.6 (Ping Test)

11.7. Drive Test Views

- It shows all the Serving cell and the RF cell information for the particular location.
- It also shows the latitude, longitude and the speed.
- Here you can see the pointer is located on the map that is you location it will move according to your location.
- This drive test is mainly used to see the cell and rf signals for different locations across the towers.

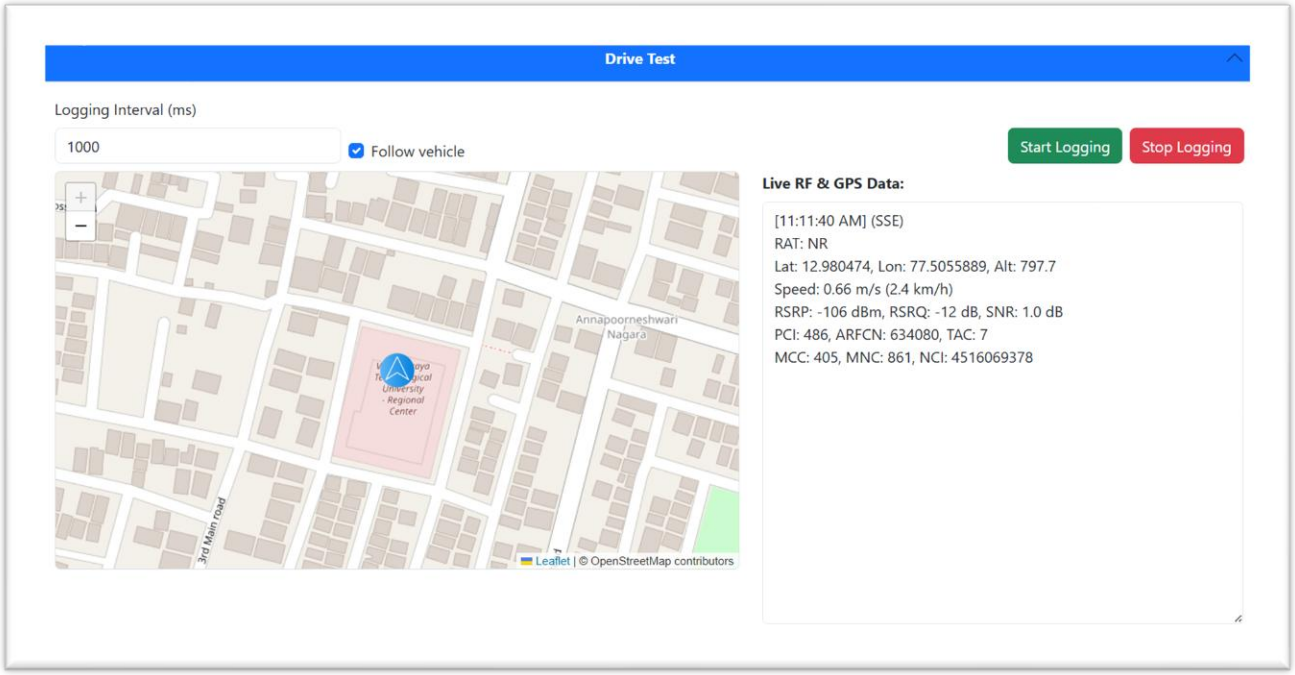


Fig 11.7 (Drive Test)

11.8. AT Commands:

- By using these commands, we can control the mobile phone.
- The general command for all is “AT”.
- Aero plane mode on in the mobile, we use this command “AT+CFUN=4”

- Aero plane mode off in the mobile, we use this command “AT+CFUN=1”
- For checking the Signal Quality, we will use this command “AT+CSQ”

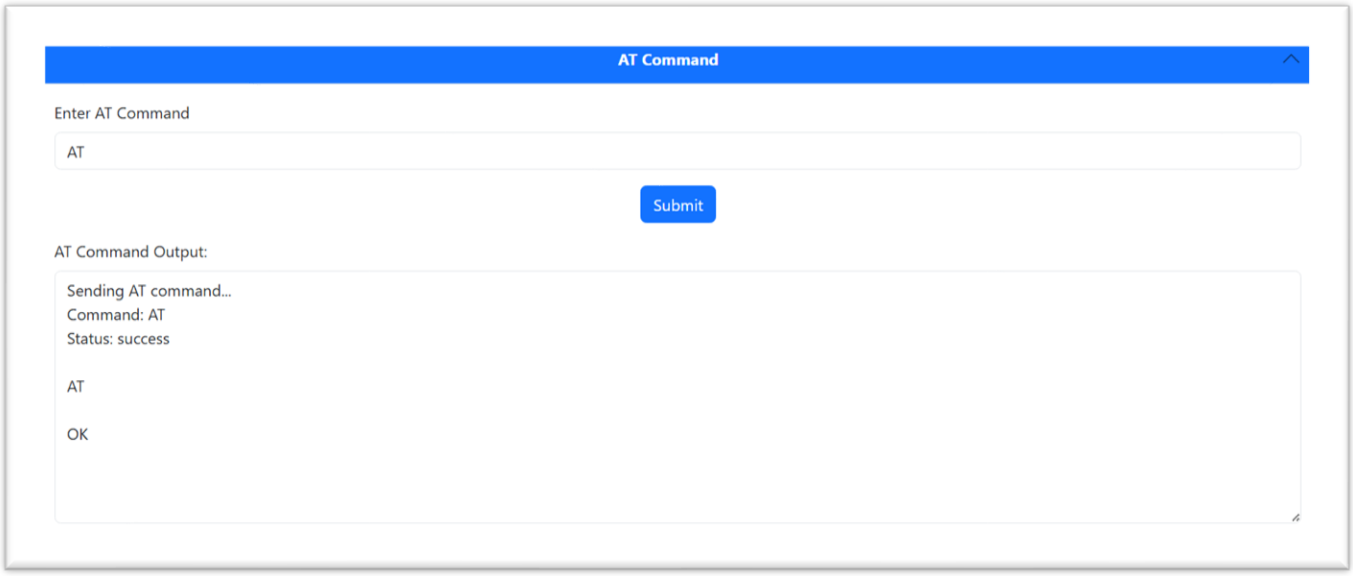


Fig 11.8 (AT Commands)

11.9. RF Signal Quality Chart:

- It will represent all the RF values in the chart format. Here we can set the time interval according to our requirement.

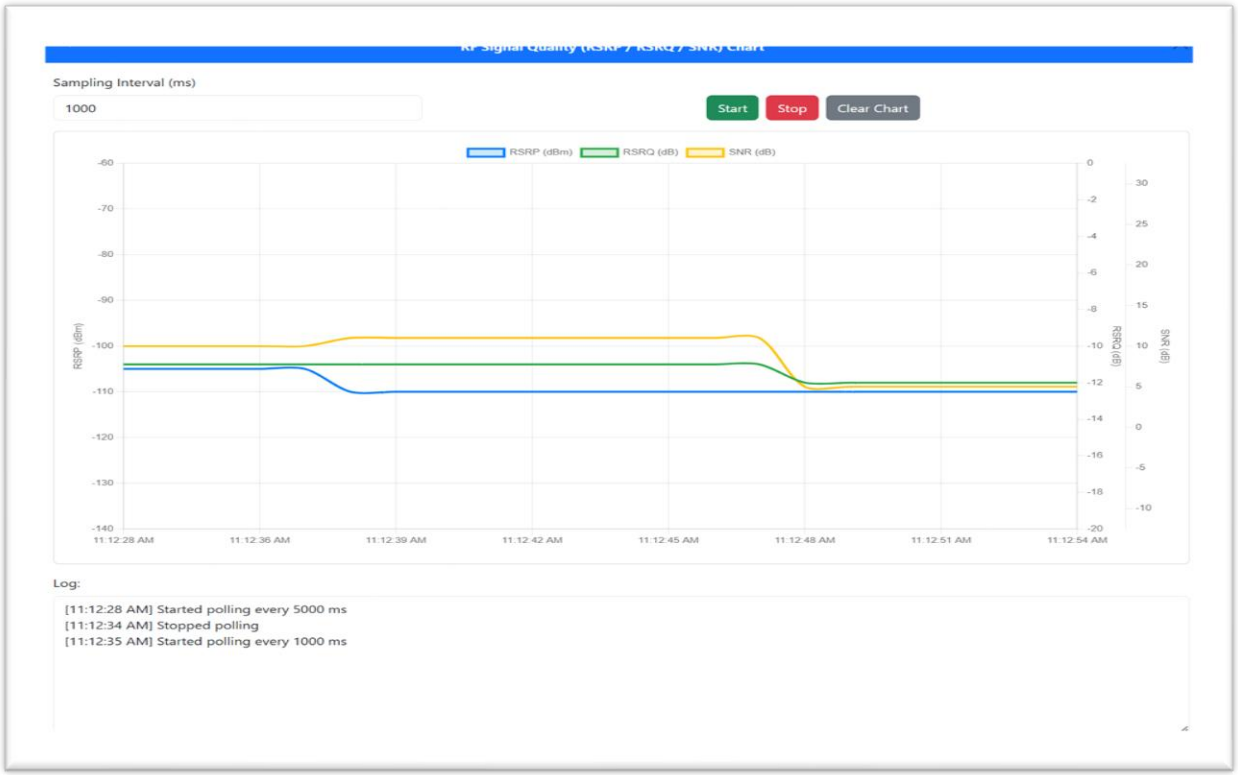


Fig 11.9 (RF Signal Chart)

12. Quick Start Checklist

- Install APK and grant permissions on all phones.
- Confirm http://phone-ip:8080/ loads on your PC browser.
- Ensure Wireshark is installed (to see the logs in Wireshark)
- Ensure java jdk is installed (to see the logs in MSC)
- Start Flask Multi-mobile server on PC and open the dashboard.
- Add each phone's IP address in the GUI page.
- Initiate a drive/walk test and verify location is on (or) not.

13. Troubleshooting & FAQs

- **Cannot connect to phone server:** Ensure phone and PC are on the same network. Check firewall settings.
- **APIs missing or null:** Ensure phone is rooted and all permissions are granted.
- **Logs not downloading:** Verify storage permissions on phone and PC.
- **No data shown:** Check network connectivity and correct server URL .

For advanced troubleshooting, see detailed error messages on the Console which is in the cots GUI page and refer to the user community or support contact for your deployment.