Reliable PRACH Capture: Diagnosing and Fixing a 5G Sniffer

Sheikh Zain Bin Hasan - 2837823
MSc Cyber Security

Supervisor: Mihai Ordean

School Of Computer Science
University Of Birmingham
2024-25

Abstract

This dissertation addresses the critical challenge of reliably detecting Physical Random Access Channel (PRACH) signals using a 5G New Radio (NR) sniffer. An experimental setup comprising an srsRAN gNB (USRP B200/B210), an NR-Scope sniffer (USRP B210), and a Pixel 6/7 User Equipment (UE) was deployed for wired testing with external clock distribution. Initial observations revealed significant performance degradation, characterized by a very low median matched-filter correlation (~0.0059), a high median Carrier Frequency Offset (CFO) of approximately 18.5 kHz, and a low median Signal-to-Noise Ratio (SNR) of -2.59 dB. To facilitate in-depth analysis, the `prach_worker.cpp` module of the sniffer was modified to export raw PRACH IQ symbols. Subsequent MATLAB-based diagnostics, including matched-filter pipelines and CFO refinement, confirmed the poor detection performance. The root cause was diagnosed as an inconsistent reference clock and Local Oscillator (LO) lock mismatch, with the gNB reporting `ref_locked=false` and the sniffer exhibiting LO status despite a motherboard reference lock. This clocking issue directly contributed to the observed high CFO and poor correlation. Recommendations include rigorous investigation and improvement of the external 10 MHz and 1PPS distribution to ensure stable and synchronized clocking across all hardware components, thereby enhancing PRACH detection reliability. Keywords: PRACH, NR-Scope, USRP, CFO, matched-filter, gNB, UE(user equipment), Matlab, Sniffer, srsRAN, Signal-to-Noise Ratio (SNR).

Acknowledgment

I would like to express my sincere gratitude to Prof. Mihai Ordean for his guidance, constructive feedback, and steady support throughout this project. His insights and encouragement were essential to the successful direction and completion of this work. I also thank Jinjin Wang for patient help with the practical basics and for hands-on assistance during the early stages of lab setup and testing. Their practical advice accelerated my learning and made the experiments possible.

Honour Code

I certify that this dissertation is my own work, except where explicitly stated otherwise. All code, analysis, and written content originate from my efforts, and I take full responsibility for the accuracy and integrity of the work presented. In the spirit of transparency and the department's "Honour Code" guidance, I declare that I used generative AI tools during the project in a supportive role. The tools and their contributions are summarized below:

1. ChatGPT (OpenAI) assisted with drafting and editing portions of the report for clarity and consistency, producing suggested rewrites and helping format technical descriptions; and provided debugging suggestions and conceptual clarifications during code development and analysis.
2. GitHub Copilot was used as a coding aid to suggest short code snippets and idiomatic constructs; all suggested code was reviewed, tested, and adapted by me.
3. Other web-based resources and software documentation (standards, datasheets, tool documentation) were consulted as primary references; generative AI was not used to replace primary technical sources.

All outputs generated by these tools were critically reviewed, validated, and edited by me before inclusion. Any text, code, or analysis derived substantially from external sources or automated tools is explicitly cited or acknowledged in this document. I understand that appropriate use of these tools will not be penalised, but misrepresenting AI-generated content or the extent of my own contributions would constitute academic misconduct, for which I accept full responsibility.

Contents

Glossary

1. 5G NR (New Radio): The global standard for the air interface of 5G networks, representing a significant advance over the previous 4G LTE standard.
2. CFO (Carrier Frequency Offset): A type of signal impairment where the carrier frequency of the received signal differs from the receiver's local oscillator frequency, which can degrade detection performance.
3. gNB (gNodeB): The 5G term for a base station. It is the network component that communicates wirelessly with User Equipment (UE).
4. I/Q (In-phase and Quadrature): A representation of a signal using two components (I and Q). This method allows for the encoding and decoding of both the amplitude and phase of a radio signal, forming the basis of modern digital modulation.
5. LO (Local Oscillator): An electronic circuit that generates a signal at a specific frequency. In a radio receiver, it is used to convert the high-frequency incoming radio signal to a lower, more manageable frequency.
6. OFDM (Orthogonal Frequency-Division Multiplexing): A digital modulation technique that splits a single high-rate data stream into multiple lower-rate streams, transmitting them simultaneously over closely spaced, orthogonal subcarrier frequencies.
7. PRACH (Physical Random Access Channel): A specific uplink physical channel that User Equipment (UE) uses to initiate contact with the network and request resources from a gNB.
8. QAM (Quadrature Amplitude Modulation): A modulation scheme that encodes data by varying both the amplitude and phase of a carrier wave.
9. SDR (Software-Defined Radio): A radio system where components traditionally implemented in hardware (e.g., mixers, filters, modulators) are instead implemented using software on a personal computer or embedded system.
10. SIB (System Information Block): A message broadcast by the gNB containing essential parameters that a UE needs to access the cell, such as PRACH configuration.
11. SNR (Signal-to-Noise Ratio): A measure that compares the power level of a desired signal to the power level of background noise. It is expressed in decibels (dB).
12. UE (User Equipment): Any device used by an end-user to communicate over a cellular network, such as a smartphone or a 5G-enabled laptop.
13. USRP (Universal Software Radio Peripheral): A family of software-defined radio hardware produced by Ettus Research (a National Instruments brand) commonly used in academic and research settings.
14. ZC (Zadoff-Chu) Sequence: A type of complex-valued mathematical sequence with ideal auto-correlation properties, making it highly suitable for generating the preamble sequences used in the PRACH procedure.

I. Introduction

The fifth-generation (5G) of wireless technology represents a paradigm shift in mobile communications, promising unprecedented data rates and ultra-low latency. This technological leap is enabled by a highly flexible and complex physical layer, primarily based on Orthogonal Frequency-Division Multiplexing (OFDM)[1]. The ability to passively monitor and diagnose this physical layer is paramount for network optimization and security analysis. A critical component of this layer is the Physical Random Access Channel (PRACH), which serves as the initial point of contact for a User Equipment (UE) to establish a connection with the network. The reliability of the PRACH procedure is fundamental to the overall performance and accessibility of the 5G network. It is necessary to diagnose the PRACH misses because a missed PRACH results in a missed C-RNTI, making the UE's traffic invisible to the sniffer. However, passively monitoring and correctly decoding PRACH signals present significant challenges, including precise time-frequency synchronization and robust signal detection in low signal-to-noise (SNR) environments. This dissertation addresses these challenges by undertaking a systematic diagnosis of a 5G sniffer's PRACH detection failures. Using an experimental testbed comprising a software-defined gNodeB (gNB), a commercial UE, and a USRP-based sniffer running NR-Scope, this work investigates the root causes of poor detection performance. The study reveals that hardware-level synchronization issues, manifesting as large Carrier Frequency Offsets (CFO), can severely degrade the efficacy of standard detection algorithms, rendering the sniffer ineffective. By modifying the sniffer's software to export raw signal data and developing a custom analysis pipeline, this project provides a detailed characterization of the problem and offers concrete recommendations for its resolution. The ability to reliably sniff Physical Random Access Channel (PRACH) transmissions is valuable to several key stakeholders. For network operators, PRACH monitoring is a vital tool for network diagnostics and optimization, allowing them to assess RACH congestion, analyze connection failures, and optimize configuration parameters to ensure efficient and reliable network access for users. From a security perspective, the initial access procedure is a potential attack surface where malicious actors could attempt flooding attacks to cause a denial of service or exploit vulnerabilities in the handshake process; passive sniffing provides a means for security analysts to detect and analyze such anomalous activities. Additionally, academic and industry researchers rely on passive sniffers to collect real-world data for developing and validating new physical layer algorithms, modeling network traffic, and understanding UE behavior in live networks, as an unreliable sniffer would corrupt this data and hinder innovation.

A. Background

To understand the challenges of PRACH detection, it is essential to review the foundational technologies of the 5G NR air interface.

1) OFDM and QAM

The 5G physical layer is built upon Orthogonal Frequency-Division Multiplexing (OFDM). This advanced modulation scheme divides a high-rate data stream into multiple lower-rate streams that are transmitted simultaneously over a set of closely spaced, orthogonal subcarriers[1]. This architecture makes 5G resilient to frequency-selective fading and multipath interference, which are common impairments in wireless channels. In simpler terms, it is a technique for transmitting a large amount of data over a radio channel without it getting corrupted. Instead of sending all the data on one big, fast-moving radio wave (which is easily disrupted by obstacles), OFDM splits the main data stream into thousands of smaller, slower streams. Each of these mini-streams is then transmitted on its own closely spaced sub-carrier frequency. The key is that these sub-carriers are "orthogonal," meaning

---

[1] *A. Zaidi et al., 5G Physical Layer: Principles, Models and Technology Components, 2nd ed. Academic Press, 2021*

they are precisely arranged so they don't interfere with each other, much like musicians in an orchestra playing different notes that harmonize perfectly. This makes the overall transmission robust against interference and signal fading, which is crucial in mobile environments. Data is imprinted onto these subcarriers using Quadrature Amplitude Modulation (QAM), a method that encodes data bits into changes in both the amplitude and phase of the carrier wave. By simultaneously changing both the amplitude and the phase of the carrier wave, QAM can pack multiple bits of data into a single signal change (symbol). For example, 16-QAM can represent 4 bits (24=16) at once, and 256-QAM can represent 8 bits (28=256), dramatically increasing data transmission speed. The combination of OFDM and QAM allows for high spectral efficiency and robust data transmission.

2) Resource Grid

The combination of OFDM in frequency and a structured time domain creates the foundational 5G New Radio (NR) resource grid. This grid can be visualized as a two-dimensional matrix where the gNB scheduler precisely maps all data and control signals for transmission.
The grid is defined by two primary axes:

a. The frequency domain (vertical axis) is composed of the orthogonal subcarriers discussed previously. For scheduling purposes, these subcarriers are typically grouped into Resource Blocks (RBs), where one RB spans 12 consecutive subcarriers.
b. The time domain (horizontal axis) is divided into OFDM symbols. A sequence of consecutive OFDM symbols typically 14 forms a slot, which is a basic unit of scheduling time.

The smallest unit of this grid is the Resource Element (RE), which represents the intersection of a single subcarrier for the duration of a single OFDM symbol. Each RE is responsible for carrying one complex-valued QAM symbol. For instance, a single RE could hold one 16-QAM symbol (representing 4 bits) or one 256-QAM symbol (representing 8 bits).
Extending the orchestra analogy, if the individual subcarriers are the musicians, the resource grid is the complete musical score. The score organizes the performance in time (the horizontal measures) and assigns specific notes to specific musicians (the vertical staves). Each individual note on that score is a Resource Element, telling a specific musician exactly what to play (the QAM symbol) at a specific moment in time (the OFDM symbol).

3) PRACH Procedure

When a UE needs to connect to the network or request uplink resources, it initiates the Random Access Procedure via the PRACH. This procedure is the UE's first handshake with the gNB and is critical for establishing a connection[2]. The process typically involves four steps:

a. Msg1 (Preamble Transmission): The UE selects a preamble from a set of predefined sequences and transmits it on the PRACH. These preambles are generated using Zadoff-Chu (ZC) sequences, which have excellent auto-correlation properties, allowing the gNB to detect them reliably and estimate the UE's timing offset [3].
b. Msg2 (Random Access Response): The gNB detects the preamble and sends a Random Access Response (RAR) on the downlink, providing a timing advance correction, a temporary identifier (TC-RNTI), and an initial uplink resource grant.
c. Msg3 (Connection Request): The UE uses the granted resources to transmit its first scheduled uplink message, typically an RRC Connection Request, which includes a permanent UE identifier.
d. Msg4 (Contention Resolution): The gNB sends a final message to resolve any potential collisions (if multiple UEs transmitted the same preamble) and confirms the connection.

---

[2] *3GPP, "TS 38.211: NR; Physical channels and modulation," Release 17, V17.3.0, Dec. 2021*

B. Aims and Objectives

The primary aim of this dissertation is to diagnose the root cause of poor PRACH detection performance in a USRP B210-based 5G NR sniffer and propose actionable improvements.
To achieve this aim, the following objectives were established:

1. To configure and deploy a controlled, wired experimental testbed consisting of an srsRAN-based gNB, a commercial UE, and the sniffer.
2. To modify the sniffer's source code to export raw I/Q symbols corresponding to detected PRACH events.
3. To develop a robust diagnostics pipeline using matlab tools to perform offline analysis, including matched-filtering, CFO estimation, and SNR calculation.
4. To characterize the sniffer's performance by quantifying key metrics from a large dataset of PRACH events.
5. To conduct a root-cause analysis by correlating the observed signal impairments with the hardware status of the gNB and sniffer.
6. To formulate concise recommendations for mitigating the identified issues and improving PRACH detection reliability.

C. Scope of Contributions

The scope of this study is confined to a wired, laboratory-based experimental setup using a single UE within a Faraday cage. The findings are specific to the hardware combination of USRP B210 SDRs and the srsRAN/NR-Scope software stacks. This work does not investigate performance in over-the-air, multi-UE, or high-mobility scenarios, and its conclusions may not be directly transferable to different hardware platforms without further validation.

II. Literature Review

This review provides a comprehensive analysis of the current landscape of 5G New Radio (NR) sniffing technologies, with a specific focus on their capabilities for Physical Random Access Channel (PRACH) detection. We begin by outlining the fundamental challenges inherent in passive sniffing of 5G signals. Subsequently, we present a detailed comparison of prominent research sniffers, including NR-Scope, 5GSniffer, and Sni5Gect. This is followed by a discussion of related academic and industry work in PRACH detection and matched filtering. Finally, we present a summary comparison and a critical gap analysis to position the contributions of this dissertation. Passive sniffing in 5G New Radio (NR) is non-trivial due to the flexibility and complexity of the physical layer. Unlike in a controlled lab environment where a User Equipment (UE) and gNodeB (gNB) are explicitly synchronized, a passive sniffer must achieve this synchronization "blindly" by listening to over-the-air signals. This presents several key challenges, primarily related to timing and frequency synchronization. A sniffer must first detect the Synchronization Signal Block (SSB) to establish coarse synchronization, but residual Carrier Frequency Offset (CFO) and timing drifts remain a significant hurdle, often exacerbated by oscillator imperfections and environmental factors. Large CFOs, as observed in the experimental setup of this project, can severely degrade the performance of subsequent decoding steps. This is particularly evident in uplink channel analysis, such as Physical Random Access Channel (PRACH) detection, where the sniffer must possess a "template" of the expected signal, like a Zadoff-Chu preamble sequence. Mismatches between the sniffer's expected template and the UE's actual transmission due to CFO or timing offset can lead to a drastic reduction in performance, as evidenced by the low correlation values in this project. A further complication is control information blindness, as much of the essential information required for decoding, such as

UE-specific configurations and security parameters, is transmitted on encrypted channels, forcing sniffers to employ computationally intensive blind decoding techniques and heuristics.

A. Comparative Analysis of 5G Sniffers

Several research-oriented sniffers have been developed to address the challenges of passive 5G monitoring, each with a different focus and set of capabilities.

NR-Scope, developed by researchers at Princeton University, is a 5G SA network telemetry tool designed for performance monitoring and optimization [1]. Its architecture is built for USRP software-defined radios and is geared towards providing real-time insights by decoding Downlink Control Information (DCI), System Information Blocks (SIBs), and RACH information. While the tool implicitly handles Carrier Frequency Offset (CFO) through standard cell search procedures, it is not explicitly designed for diagnosing the large CFOs that can arise from hardware clocking issues. A key capability of NR-Scope is its ability to decode RACH information to track UE attachments and resource requests, making it highly relevant to this project. NR-Scope's main strength is its focus on a holistic view of RAN performance; however, it is primarily a downlink-focused tool, and its PRACH capabilities are geared towards monitoring successful detections rather than diagnosing physical layer failures.

5GSniffer, a security-focused tool from Northeastern University, is designed for the blind decoding of the Physical Downlink Control Channel (PDCCH)[3]. As a highly optimized C++ application, its architecture is specifically tailored to this computationally intensive task, processing I/Q samples from either a file or an SDR. Similar to NR-Scope, 5GSniffer relies on initial synchronization to handle CFO and assumes a reasonable level of synchronization has been achieved. The tool's primary focus on the PDCCH means it does not have specific mechanisms for detecting or analyzing PRACH preambles themselves. Consequently, 5GSniffer's strength lies in its specialized, high-performance PDCCH decoding, while its main weakness, in the context of this project, is its lack of focus on uplink channels like the PRACH.

Sni5Gect is a powerful framework for both sniffing and injecting messages in 5G NR, with a focus on pre-authentication attacks[4]. Built on srsRAN, it uses a USRP B210 for over-the-air capture and injection, and its stateful architecture allows it to track the UE attachment procedure. As a tool capable of both receiving and transmitting, Sni5Gect leverages the underlying srsRAN libraries for robust CFO estimation and correction. It is designed to sniff both uplink and downlink pre-authentication messages, which explicitly includes the RACH procedure, allowing it to monitor the initial UE connection process. Sni5Gect's unique strength is its combination of sniffing and stateful injection for security research; its weakness is that its primary goal is launching attacks rather than detailed physical layer diagnostics.

B. Related Work in PRACH Detection

The problem of Physical Random Access Channel (PRACH) detection has been extensively studied in both academic and industry contexts. The core of most detection algorithms, representing the established industry standard, is the matched filter. This approach, detailed in 3rd Generation

---

[3] L. R. Nuñez, et al., "From 5G Sniffing to Harvesting Leakages from Privacy-Preserving Messengers," in 2023 IEEE Symposium on Security and Privacy (SP), 2023.

[4] S. Luo, M. Garbelini, S. Chattopadhyay, and J. Zhou, "Sni5Gect: A Practical Approach to Inject aNRchy into 5G NR," in 34th USENIX Security Symposium, 2025.

Partnership Project (3GPP) specifications and implemented in tools like MATLAB's 5G Toolbox[5], correlates the received signal with a known template of the PRACH preamble, typically a Zadoff-Chu sequence. Standard research in this area often focuses on optimizing the detection threshold to balance the probability of detection against the false alarm rate, especially in low Signal-to-Noise Ratio (SNR) conditions. More advanced research explores techniques to improve PRACH detection in the presence of interference and other channel impairments, including methods such as machine learning-based detectors and enhanced algorithms that use multi-detection steps to reject false alarm peaks[6]. These works, however, often assume a well-behaved channel and do not specifically address the large, hardware-induced Carrier Frequency Offsets (CFOs) that are the focus of this dissertation.

C. Summary and Critical Gap Analysis

Table I: Key Design Parameter Rationale

Feature	NR-Scope (Princeton)	5GSniffer (S&P '23)	Sni5Gect (USENIX Sec '25)
Primary Focus	Network Telemetry	PDCCH Security/Privacy	Sniffing & Injection (Security)
PRACH Support	Yes (RACH Info Decoding)	No	Yes (Initial Access Monitoring)
Live Decoding	Yes	Yes	Yes
Template Extraction	No	No	No
Multi-antenna Support	Yes (with multiple USRPs)	Not explicitly stated	No (SISO)
External Clock Support	Yes (via USRP)	Not explicitly stated	Yes (via USRP)
Reported Accuracy	< 0.1% throughput error	High DCI recovery rate	> 80% sniffing accuracy

[5] *MathWorks, "5G NR PRACH Detection and False Alarm Test," MATLAB & Simulink Documentation. [Online]. Available: https://www.mathworks.com/help/5g/ug/5g-nr-prach-detection-test.html*

[6] *T. A. Pham, et al., "A proposed preamble detection algorithm for 5G-PRACH," in 2019 International Conference on Advanced Technologies for Communications (ATC), 2019.*

The existing state-of-the-art sniffers, while powerful, have limitations that this dissertation aims to address. NR-Scope and Sni5Gect, while capable of monitoring the RACH procedure, are designed to analyze successful events for telemetry or security purposes, respectively. They lack the specialized diagnostic tools needed to investigate the root cause of PRACH detection failures at the physical layer, especially those stemming from hardware-level issues like clock synchronization. 5GSniffer, on the other hand, is entirely focused on the downlink control channel and does not address PRACH at all. This project fills a critical gap by shifting the focus from monitoring successful events to diagnosing failures. By modifying an existing sniffer (NR-Scope) to export raw PRACH I/Q symbols, we have created a pipeline for in-depth, offline analysis that is not offered by any of the reviewed tools. This approach allows for a detailed investigation of the impact of large CFOs and low SNR on the matched-filtering process and provides a methodology for root-cause analysis of hardware-induced performance degradation. The findings of this work will be invaluable for researchers and engineers who need to debug and validate the physical layer of 5G sniffing and communication systems.

III. Legal, Social and Professional Context

Undertaking research involving the capture and analysis of wireless signals necessitates a thorough consideration of the surrounding legal, social, and professional landscape. While this project is conducted in a controlled lab environment, the principles governing radio transmission, data privacy, and ethical conduct are directly applicable and essential for responsible research.

A. Regulatory Framework

The use of radio spectrum is a strictly regulated activity. International and national bodies establish rules to prevent interference and manage spectrum as a finite public resource.
Spectrum Licensing and Regulations: The International Telecommunication Union (ITU) provides a global framework for spectrum management, which is then implemented by national regulators[7]. In the United Kingdom, the relevant authority is Ofcom, which manages the spectrum under the Wireless Telegraphy Act 2006[8]. This act makes it illegal to establish or use a wireless telegraphy station or apparatus without a license. While this project's sniffer is a passive listening device, the gNB (USRP B200/B210) is an active transmitter. Operating it, even at low power in a lab, technically requires regulatory approval or must be done in a way that guarantees zero emission outside of a shielded environment. The use of a Faraday cage is therefore not just good practice but a critical mitigation to ensure compliance and prevent interference with licensed public mobile networks. A similar framework exists in the United States, managed by the Federal Communications Commission (FCC)[9].
Lawful Intercept and Privacy of Communications: The act of intercepting communications is governed by stringent laws. In the UK, the Regulation of Investigatory Powers Act 2000 (RIPA)[10]

---

[7] *International Telecommunication Union (ITU), "Radio Regulations," Geneva, Switzerland, 2020.*

[8] *UK Public General Acts, "Wireless Telegraphy Act 2006," c. 36, London, UK: The Stationery Office, 2006.*

[9] *Federal Communications Commission (FCC), "Title 47 of the Code of Federal Regulations (CFR)," Washington, D.C., USA.*

[10] *Regulation of Investigatory Powers Act 2000, UK Statute, 2000. Available: https://www.legislation.gov.uk/id/ukpga/2000/23*

outlines the legal framework under which public bodies can conduct surveillance and intercept communications. Passive monitoring for academic research falls outside of this "lawful intercept" framework and carries no legal authority to capture third-party data. Capturing any data that is not your own without consent could be deemed an unlawful interception. This legal boundary reinforces the ethical imperative to ensure that only signals from known, consenting devices are ever captured.

B. Social and Ethical Consideration

Beyond strict legal compliance, the project touches upon significant social and ethical issues centered on privacy and consent. 5G control channel signals, including those involved in the RACH procedure, can contain sensitive metadata and persistent identifiers (e.g., SUCI). While not the content of a call, this metadata can reveal a user's location, device type, and patterns of life. The potential for this information to be collected and analyzed without a user's knowledge raises profound privacy concerns. The principles of the General Data Protection Regulation (GDPR) are highly relevant, particularly data minimization (collecting only what is necessary) and purpose limitation (using data only for the stated research goal)[11].

C. Recommended Mitigations

To address these issues, this project incorporates the following specific mitigations:
Physical Containment: All experiments are conducted within an RF-shielded Faraday cage to prevent signal leakage and ensure no interference with public networks or capture of non-consenting device signals.
Data Minimization & Anonymization: Only the I/Q samples of the PRACH preamble are captured and stored. Any discovered device identifiers will be redacted from all reports and datasets.
Secure Storage: All project data is maintained on encrypted hard drives, accessible only to the primary researcher.
By integrating these legal, ethical, and professional considerations, this research demonstrates a commitment to responsible innovation.

IV. System Requirements

This section specifies the measurable system requirements for the diagnostic and improvement work on the NR-Scope sniffer used alongside an srsRAN gNB on the SDR hardware. Where helpful, requirements reference configuration parameters and artifacts produced in the project (e.g., the prach_worker.cc instrumentation and gNB/sniffer YAMLs recorded in the project archive).

A. Functional Requirements

1. PRACH capture and detection. The sniffer must capture uplink PRACH occasions and produce a detection event record for each gNB-observed PRACH. In offline replay tests (replaying gNB phy_rx_symbols_*.fc32 and sniffer .fc32 captures) the sniffer shall achieve recall ≥ 95% and precision ≥ 90% when compared against the ground-truth gNB RX_PRACH event table described in the logs. For validation, compare per-event timestamps/SFN+slot between gNB logs and sniffer detections; compute recall/precision over N ≥ 500 events.

---

[11] *European Parliament and Council of the European Union, "Regulation (EU) 2016/679 (General Data Protection Regulation)," Official Journal of the European Union, L 119/1, 2016*

2. Sample fidelity and file-format parity. Recorded IQ dumps from the sniffer's modified prach_worker must be written as interleaved IEEE-754 float32 IQ samples (same format as gNB phy_rx_symbols_*.fc32) and include per-thread unique filenames so MATLAB comparisons can align files unambiguously. Acceptance: dumped files open in MATLAB and report complex sample counts identical (±1 sample) to the equivalent gNB slot captures. File header/naming must follow the lab convention (for example phy_rx_sniffer_worker<id>_sfn<XXXX>_slot<Y>_port<P>_<unix_ts>.fc32).
3. Frequency-offset & timing constraints. After applying external clock/reference fixes, the residual Carrier Frequency Offset (CFO) measured per PRACH event must be < 1 kHz (post-synchronisation) for ≥ 90% of events. Acceptance: histogram of per-event CFO estimates with 90th percentile < 1 kHz. This requirement follows the observed root cause and corrective actions documented in the diagnostic work.
4. Offline and real-time operation modes. The sniffer shall support both, offline playback of .fc32 captures (file backend) and live decoding from a USRP backend. Acceptance: the YAML device_name: "file" playback mode must successfully parse rx_files with rx_channels: [0] and allow full offline detection; the live mode must initialize the USRP and begin live capture with no device initialization errors in the logs.

B. No-Functional Requirements

1. Sampling-rate constraints. All sampling rates used for capture and processing must be multiples of 1.92 MHz (NR/srsRAN processing constraint). Typical validated rate: 23.04 MHz (23040000 Hz). Acceptance: srate_hz and srsran_srate_hz parameters in YAML must be set to values that are exact integer multiples of 1.92e6; MATLAB resampling ratios must be ≤ |1%| before rational approximation.
2. Performance and latency. For real-time operation, processing latency from RF ingest to logged detection must be < 100 ms (median) under lab load on the target host. Offline playback has no hard latency bound but must complete analysis without sample format conversion errors. Acceptance: measure end-to-end latency with timestamped events; median < 100 ms.
3. Host & hardware reliability. The host must be x86_64 Linux (Ubuntu), with at least 4 physical cores, 16 GB RAM, SSD storage, and USB3.0 (for B2xx). USRP hardware in use: USRP B210 (primary), optional additional B2xx for diversity; clock distribution: CDA-2990 (10 MHz/1PPS) as used in the testbed[12]. UEs: Pixel 6 and Pixel 7 (test phones). Cables: SMA coax, USB3, splitter (SMA) and power combiner/distributor as used in the lab. Acceptance: device enumeration (uhd_usrp_probe) shows the expected serial numbers and external reference locked status.

C. Hardware and Software Requirements

The laboratory baseline includes an Ettus USRP B210 (USB 3.0 host), an RF front-end comprising a splitter with a CDA-2990 directional coupler and inline attenuators, and a Google Pixel 6 test UE placed inside a Faraday enclosure to generate PRACH. Two Ubuntu laptops serve as hosts one runs the srsRAN gNB and the other runs NR-Scope/sniffer each provisioned with ≥4 CPU cores, 16 GB RAM and SSD storage. Software tools include srsRAN, NR-Scope and MATLAB (matched-filter analysis); project artifacts and YAML configurations are listed in the appendices.

---

[12] *National Instruments, "CDA-2990 Clock Distribution Amplifier — Specifications and Datasheet," NI, 2024. Available: https://www.ni.com/docs/en-US/bundle/cda-2990-specs/page/specs.html*

D. Parameters to be Validated

Before each run, verify the gNB and sniffer YAMLs contain the exact keys and value forms required for the experiment. For the gNB YAML confirm: log.phy_rx_symbols_filename: /home/uob/work/log/phy_rx_symbols, log.phy_rx_symbols_port: 0, and log.phy_rx_symbols_prach: true. For the sniffer YAML (file-mode or live) confirm: device_name: "file" (or "uhd" for live), rx_files: ["/path/to/capture.fc32"], rx_channels: [0], srate_hz: 23040000, and srsran_srate_hz: 23040000. All entries must exactly match the lab file paths and the 23,040,000 Hz sampling rates used in the experiment.

V. Design and Architecture

This section presents the comprehensive design of the 5G New Radio (NR) Physical Random Access Channel (PRACH) signal sniffing and analysis system. The end-to-end architecture is detailed, from the physical hardware front-end that captures raw radio frequency (RF) signals to the multi-stage software pipeline that processes and analyzes the captured data. A rigorous justification for all critical design parameters is provided, and the software implementation is discussed with a focus on modularity and reusability.

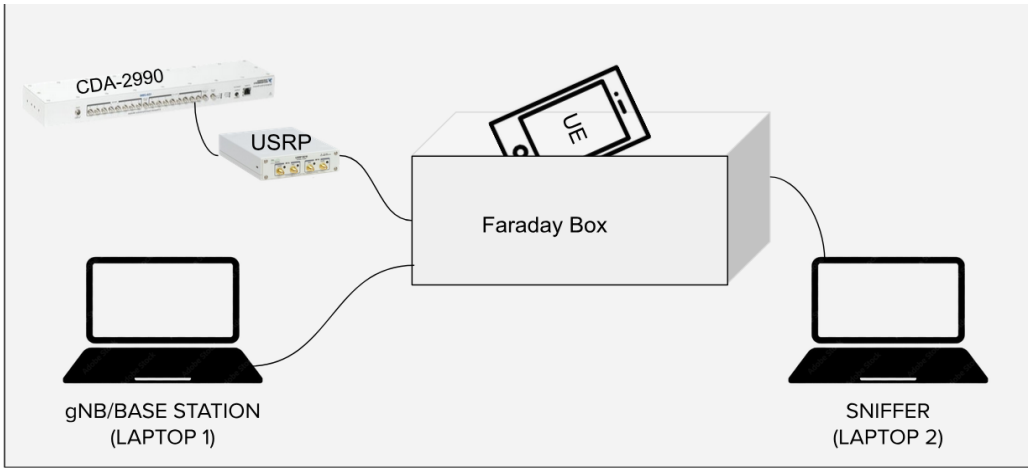


Figure 1

A. System Architecture and Signal Processing Pipeline

The system is architected as a passive, off-the-air monitoring platform designed to capture and analyze PRACH transmissions from a commercial or lab-based 5G gNodeB (gNB). The signal processing chain is logically divided into three primary stages: RF Signal Conditioning and Capture, Real-time Signal Processing and Triage, and Offline In-depth Analysis. This hierarchical approach balances the computational demands of high-fidelity signal analysis with the practical constraints of real-time data rates.

The hardware front-end is responsible for capturing the RF signal with high fidelity. The signal source is a standard-compliant 5G gNB, from which the system captures PRACH transmissions, which constitute Msg1 of the initial access procedure[13]. A direct, wired connection from the gNB's antenna port is used to maintain a high Signal-to-Noise Ratio (SNR). An RF attenuator is inserted immediately

[13] *5g nr prach detection with convolutional neural networks (cnn): overcoming cell interference challenges - arXiv, accessed August 30, 2025, https://arxiv.org/pdf/2408.11659?*

after the gNB output. Its primary function is to reduce the signal power to a level that does not saturate the sensitive analog front-end of the sniffer, thereby preventing non-linear distortion and preserving the integrity of the captured signal. An RF power splitter follows the attenuator, enabling simultaneous monitoring by the sniffer and another device, such as a commercial User Equipment (UE) for ground-truth validation, without creating a significant impedance mismatch. A Universal Software Radio Peripheral (USRP), such as an NI USRP X310 or B210, serves as the core of the capture hardware. It performs down-conversion from the carrier frequency to baseband and digitizes the signal into in-phase and quadrature (I/Q) samples. The USRP's high-speed sampling capabilities and open-source driver support make it an ideal interface for software frameworks like NR-Scope.

Table II: System Component Specifications

Component	Type	Model/Software	Key Role/Parameters
gNB (Base Station)	Hardware/Software	Laptop 1 running srsRAN	Source of 5G NR signal; Band n78, 20 MHz BW
Sniffer Host	Hardware	Laptop 2	Host PC for USRP B210 and NR-Scope
User Equipment (UE)	Hardware	Google Pixel 6/7	Commercial UE for ground-truth validation
RF Enclosure	Hardware	Faraday Box	Isolates the RF environment to prevent external interference
USRP	Hardware	NI USRP B210	RF capture and digitization; 2x2 MIMO, 70 MHz - 6 GHz
Clock Distribution	Hardware	NI CDA-2990	Distributes 10 MHz/1 PPS signals for synchronization
NR-Scope	Software Framework	v23.04	Real-time 5G signal processing and scheduling
prach_worker	Software Module	Custom C++	PRACH candidate detection and I/Q export trigger
MATLAB	Software Environment	R2024a (with 5G Toolbox and Signal Processing Toolbox)	Offline signal analysis and visualization

B. Real Time Processing Pipeline

The real-time processing stage uses the NR-Scope framework, a proven passive 5G telemetry tool known for its ability to decode Downlink Control Information (DCI) and track UEs in real-time. The standard NR-Scope pipeline is extended with a custom prach_worker module. This worker's

responsibility is to continuously scan the incoming uplink I/Q stream for energy signatures corresponding to PRACH transmission occasions[14]. Upon a positive, albeit coarse, detection by the prach_worker, a dedicated function is invoked to write a segment of the I/Q buffer to a binary file (e.g., .interleaved IEEE-754 float32 (.fc32))[15]. This triage step is essential for managing storage, as continuously saving the entire high-rate I/Q stream is infeasible.

C. Offline Analysis

The exported I/Q data were loaded into MATLAB for high-fidelity analysis. A matched filter was applied first: the received signal was correlated with a locally generated, ideal PRACH preamble template. The matched-filter output exhibits a sharp peak at the time instant of the preamble arrival, which maximizes the SNR for detection purposes[16]. The matched-filter peak also shows a phase rotation caused by the carrier frequency offset (CFO) between the UE transmitter and the gNB receiver; a derotation step corrects this by multiplying the signal by a complex exponential with the conjugate of the estimated phase offset, thereby aligning the signal for subsequent processing. To further improve SNR, multiple detected PRACH preambles are coherently averaged (stacked). By aligning preambles in time (using the matched-filter peak) and in phase (after derotation), coherent signal components add constructively while random noise averages to zero, producing an SNR improvement $\propto N$ (where N is the number of coherently stacked preambles).

This two-stage detection process represents a deliberate architectural trade-off. Performing high-fidelity matched filtering for all possible PRACH preambles in real-time on a general-purpose CPU is computationally prohibitive. Therefore, the system uses a lower-complexity real-time worker for candidate detection, which triggers the export of a small I/Q segment. This offloads the computationally intensive analysis to a non-real-time environment, making the system practical and scalable without requiring specialized hardware for the entire analysis chain

D. Core Choices and Rationale

The system performance depends critically on a set of configuration parameters. The selection of these parameters is justified with reference to 3GPP specifications, signal-processing principles, and the analysis objectives.

1) PRACH Configuration

Preamble Format B4. In accordance with 3GPP TS 38.211, PRACH format B4 is a short-preamble configuration. Short preambles are characterized by sequence length $L_{RA} = 139$. Format B4 is defined for use in Frequency Range 1 (FR1) and is commonly associated with 30-kHz subcarrier spacing (SCS), occupying four OFDM symbols in the time domain[17]. The selection of format B4 is appropriate for typical urban macro and micro-cell deployments: its shorter duration relative to long

---

[14] *5G-NR PRACH Detection Performance Optimization in Context of Intra/Inter-Cell Interference - arXiv, accessed August 30, 2025, https://arxiv.org/html/2408.14097v1*

[15] *5G Waveform Analyzer - Analyze and visualize imported or captured 5G waveforms and export analysis results - MATLAB - MathWorks, accessed August 30, 2025, https://www.mathworks.com/help/5g/ref/5gwaveformanalyzer-app.html*

[16] *Matched filter - Wikipedia, accessed August 30, 2025, https://en.wikipedia.org/wiki/Matched_filter*

[17] *3GPP, "TS 38.211 — NR; Physical channels and modulation," 3GPP (Release 16), Jul. 2020. Available: https://www.3gpp.org/dynareport/38211.htm*

preambles enables more frequent PRACH occasions, reduces access latency, and provides a favorable trade-off between detection robustness and resource efficiency.
Root Sequence Index Selection. PRACH preambles are generated from Zadoff–Chu (ZC) sequences, which exhibit ideal auto-correlation and low cross-correlation properties. A single ZC root sequence can produce multiple orthogonal preambles via cyclic shifts. To mitigate inter-cell interference, each cell broadcasts a logical rootSequenceIndex in its System Information Block (SIB); this logical index is mapped to a physical root sequence per TS 38.211. For accurate offline analysis, the MATLAB template generator must be configured with the same rootSequenceIndex as the target gNB so that the locally generated template matches the UE transmissions in the cell under test.

2) Sampling and Filtering Parameters

Sample Rate (23.04 MHz). 3GPP sampling rates are derived from the OFDM parameters to ensure alignment between time-domain samples and the frequency-domain resource grid, following the relationshipfs=NFFT×SCSfs =NFFT ×SCS. The sampling rate 23.04 MHz is a standard supported by many SDR platforms and open-source stacks (e.g., srsRAN) for 5G NR operation. It provides sufficient capture bandwidth for common FR1 channel bandwidths up to 20 MHz while satisfying Nyquist requirements and maintaining compatibility with the system hardware and software stack.
Matched-Filter Template Length. A matched filter is optimal when its impulse response is the time-reversed conjugate of the signal to be detected. Because PRACH format B4 employs a ZC sequence of length LRA=139 the matched-filter template used in MATLAB must also be length-139. Using a template of nonmatching length produces a correlation mismatch and yields suboptimal SNR at the filter output, degrading detection performance

These design choices are deeply interconnected. The selection of PRACH format B4 dictates the sequence length (LRA =139), which in turn strictly defines the matched filter template length. The target channel bandwidth and the need for software compatibility lead to the selection of the 23.04 MHz sample rate. This sample rate then determines the precise number of time-domain samples corresponding to the PRACH transmission's duration, which must be extracted by the IQ Export module to ensure the full preamble is captured for offline analysis.

Table III: Key Design Parameter Rationale

Parameter	Selected Value	Rationale & 3GPP Reference
Sample Rate	23.04 MSps (1536 × 15 kHz)	It provides sufficient passband to capture a 20 MHz channel when configured appropriately.
PRACH Format	B4	Short preamble (LRA =139), 4-symbol duration. Suited for moderate cell sizes with low access latency. (TS 38.211, Tbl 6.3.3.1-1).
Root Sequence Index	Cell-specific (e.g., 0)	Ensures local template matches transmitted signal; critical for inter-cell interference

		mitigation. (TS 38.211, Tbl 6.3.3.1-3).
Matched Filter Template Length	139 samples	Must match the preamble sequence length (LRA) for optimal SNR gain and detection performance.

E. Implementation and Software Modularity

The system's implementation relies on targeted modifications to the NR-Scope software and a modular design that promotes reusability.

A new C++ module, the prach_worker, is created within the NR-Scope framework.9 Its core function identifies time-frequency resources allocated for PRACH based on the cell's SIB1 information. It then performs energy detection across these resources to flag potential PRACH transmissions. Upon detecting a candidate preamble, the prach_worker pushes detection metadata and a pointer to the relevant I/Q buffer segment onto a ZeroMQ socket connected to a separate exporter process, decoupling real-time detection from file I/O operations.

The system is designed with distinct, loosely coupled components. The prach_worker is a pluggable module, the IQ_Exporter is a standalone process, and the MATLAB analysis scripts are entirely separate. This modularity allows, for example, the MATLAB analysis pipeline to be swapped with a GNU Radio implementation without altering the capture front-end. This approach aligns with modern SDR development practices that emphasize flexible, software-defined functionality.

F. System Timing and Synchronization

Accurate PRACH analysis requires a precise understanding of the preamble's location in the time domain. The 5G NR frame structure provides this hierarchical timing framework. The sniffer system must first synchronize to the gNB's downlink signal, detect the Synchronization Signal Block (SSB), and decode the Master and System Information Blocks (MIB and SIB1). The information in SIB1, specifically the prach-ConfigurationIndex, provides the key to the timing and frequency location of PRACH resources[18]. Without this initial downlink synchronization, the prach_worker would be searching blindly, a computationally infeasible task. This timing information, derived from decoding SIB1, is critical. The prach_worker uses this knowledge to narrow its search window to specific OFDM symbols within valid PRACH slots, significantly reducing computational load and improving the efficiency of the real-time detection process.

VI. Implementation

The successful implementation of the 5G Physical Random Access Channel (PRACH) analysis system hinges on a meticulously configured hardware and software environment. This section details the physical deployment, the logical architecture of the software components, and the end-to-end flow of signal data from generation to analysis. The architecture is designed to facilitate high-fidelity, over-the-air capture of PRACH signals for subsequent offline processing.

[18] *RACH in 5G-NR - Techlteworld, accessed August 30, 2025, https://techlteworld.com/rach-in-5g-nr/*

A. Hardware and Software Environment

The testbed employs a two-node configuration, physically separating the 5G base station (gNB) from the passive signal sniffer and analysis workstation[19]. This separation is crucial for emulating a realistic over-the-air channel and ensuring that the resource-intensive sniffing process does not interfere with the real-time operation of the gNB. The hardware consists of two high-performance laptops, two Universal Software Radio Peripheral (USRP) B210 software-defined radios (SDRs), and a dedicated clock distribution amplifier for synchronization. The first laptop is a workstation with a minimum 12-core CPU and 16 GB of RAM running Ubuntu 22.04 LTS. This machine is dedicated to running the srsRAN Project gNB software stack. It is connected to a USRP B210, which serves as the radio frequency (RF) front-end for the gNB. The second laptop is a workstation with identical specifications to Laptop 1, also running Ubuntu 22.04 LTS. This machine runs a modified version of the NR-Scope passive sniffer and is connected to the second USRP B210 for signal capture. For synchronization a National Instruments (NI) CDA-2990 Clock Distribution Amplifier is used to provide a common timing reference to both USRPs. It distributes a stable 10 MHz reference clock (REF) and a 1 pulse-per-second (PPS) signal, which is essential for achieving phase-coherent operation between the transmitting gNB and the receiving sniffer. This external reference lock is a non-negotiable requirement for accurate timing and frequency-offset measurements.
The system integrates several open-source and commercial software packages for various purposes. As for gNB, srsRAN Project provides a complete, 3GPP-compliant 5G Core (5GC) and Radio Access Network (RAN) solution, including a software-based gNB that can be run on general-purpose processors. For sniffer, a modified fork of NR-Scope, a passive 5G Standalone (SA) network telemetry tool, is used for signal capture.6 The base version of NR-Scope is designed to decode control information in real-time; for this project, it is modified to capture and store raw in-phase and quadrature (I/Q) samples[20]. To analyze the captured I/Q data, MATLAB R2024a (or later) with the 5G Toolbox and Signal Processing Toolbox is used.

B. Flow of Data and Signal

The system's operation follows a linear signal processing chain. The gNB generates and transmits a 5G NR signal, the UE generates PRACH preambles (user toggles airplane mode to force multiple attempts). The sniffer captures the raw I/Q data corresponding to PRACH transmissions, and the MATLAB pipeline processes this data to extract key parameters.The standard NR-Scope architecture is designed for real-time metadata extraction, not for storing large volumes of raw I/Q data.6 Its worker pool is optimized for low-latency processing of individual slots, which can be as short as 0.25 ms.6 Directly writing to a file from within a worker thread is a blocking I/O operation that would cause the system to miss subsequent slots, leading to significant data loss. To circumvent this performance bottleneck, the architecture is modified to decouple real-time capture from non-real-time file I/O using a thread-safe producer-consumer pattern. This modification fundamentally repurposes NR-Scope from a telemetry tool into a high-fidelity I/Q capture instrument. The logical flow is as follows.

- The srsran_gnb component, configured via a YAML file, generates 5G NR downlink frames that include PRACH occasions in accordance with 3GPP specifications
- The generated signal is transmitted over the air by the gNB using a USRP B210.

---

[19] *srsRAN Project, "srsRAN Project Documentation," srsRAN Docs. Available: https://docs.srsran.com/projects/project*

[20] *Princeton University, "NR-Scope," GitHub repository, https://github.com/PrincetonUniversity/NR-Scope*

- The nr_scope sniffer (USRP B210) digitizes the received RF waveform into a stream of complex I/Q samples.
- NR-Scope's internal scheduler dispatches buffers containing slot-level data to an available prach_worker thread for processing.
- The modified prach_worker locates a PRACH occasion and, rather than discarding the I/Q samples after processing, enqueues them into a thread-safe, non-blocking queue.
- A dedicated, low-priority IQ_Writer thread consumes I/Q packets from the queue and writes them sequentially to a binary file on disk.
- Finally, the matlab pipeline ingests the binary I/Q file for offline analysis, applies matched filtering to detect preambles, and estimates Timing Advance (TA) and Random Access Preamble ID (RAPID).

C. Pitfalls and Fixes

This section identifies and solves critical system failures by enforcing a common external clock for all radio hardware , making the analysis software robust against empty signal data, and ensuring frequency settings are correctly converted and matched across all configuration files. Each subsection follows the structure: symptom, root cause, and remediation.

1) Hardware Synchronization Failure

Symptom: The MATLAB processing produces a power delay profile (PDP) that is noise-like and lacks a distinct correlation peak despite a strong transmitted signal; PRACH detections are not observed.
Root Cause: The gNB and sniffer USRPs are operating from their internal TCXOs. Even though TCXOs are relatively stable, small frequency and sampling-clock offsets (ppm-level) between independent oscillators lead to a time-varying carrier frequency offset (CFO) and sampling-clock offset. These offsets destroy the required orthogonality between transmitted and received samples and prevent reliable detection.
Remediation: Ensure both platforms are locked to a common, high-stability reference.
Physical connection: Route the 10 MHz reference and PPS signals from a common clock distribution unit (for example, NI CDA-2990) to each USRP. Use SMA cables to connect the clock unit's REF OUT and PPS OUT to the REF IN and PPS IN ports of every USRP under test.
Device configuration: Configure both the srsRAN gnb.yml and the NR-Scope config.yaml UHD device arguments to use external clock and time sources (e.g., clock=external, time_source=external). This forces the USRPs to phase-lock to the external reference and aligns their time and frequency bases[21].

2) Matlab Analysis Failure

Symptom: The MATLAB analysis script aborts with errors originating from nanmedian, typically when batch-processing many captures.
Root Cause: Some PRACH occasions contain no UE transmissions, producing I/Q buffers of pure noise. If a downstream noise-gating stage replaces low-power PDP values with NaN, an entire PDP slice can become all-NaN. Calling nanmedian (or other reduction functions) on an all-NaN vector yields NaN, which propagates and causes subsequent arithmetic or logical operations to fail.

---

[21] *srsRAN gNB with srsUE, accessed August 30, 2025, https://docs.srsran.com/projects/project/en/latest/tutorials/source/srsUE/source/index.html*

Remediation: Replace deprecated usage and add explicit checks.

1. Function update: Replace nanmedian(X) with the modern equivalent median(X, 'omitnan') to conform to current MATLAB best practices.
2. Robust edge handling: Test for the all-NaN case before reduction and handle it deterministically (skip the slice or assign a safe default).
3. Logging & instrumentation: Log occurrences of all-NaN slices (count and timestamp) to help diagnose whether missing transmissions or capture faults are the root cause.

3) Configuration Parameter Mismatch

Symptom: NR-Scope fails to detect the gNB's Synchronization Signal Block (SSB) during its initial cell search, or it captures signals at an incorrect frequency, resulting in empty I/Q files.
Cause: There is a mismatch in how the gNB and sniffer frequencies are configured. The srsRAN gnb.yml file specifies the cell's frequency using the 3GPP-standard dl_arfcn (Downlink Absolute Radio-Frequency Channel Number) for a given band.10 In contrast, the NR-Scope config.yaml requires the frequency to be specified directly in Hertz (e.g., center_freq_hz).18 Manually entering the ARFCN value into the frequency field is a common mistake.
Remediation:

1. Explicit Conversion: Use a reliable 5G NR ARFCN calculator or a reference table (such as in 3GPP TS 38.101) to convert the dl_arfcn and band from the gNB configuration into the precise center frequency in Hz.
2. Configuration Management: Maintain a single reference document for the experiment that lists the chosen band, ARFCN, and the correctly calculated center frequency in Hz. Use this document to populate both YAML configuration files to prevent discrepancies.

VII. Testing and Success Measurement

A comprehensive, multi-tiered testing methodology was established to rigorously validate the 5G New Radio (NR) Physical Random Access Channel (PRACH) detection pipeline of the sniffer system. This strategy is systematically structured into three distinct tiers Unit, Integration, and Performance, to methodically verify system functionality and de-risk the project from the lowest-level software modules to the complete, end-to-end signal processing chain. This tiered approach is fundamental to ensuring the robustness and reliability of the system. The primary objective of this validation framework is to diagnose and address the root causes of missed PRACH detections, a key challenge identified during initial system assessments. The Unit Testing tier focuses on the atomic software components, verifying the correctness of core algorithms and data handling logic in isolation. The Integration Testing tier examines the critical interface between the software pipeline and the Software-Defined Radio (SDR) hardware, ensuring stable operation and calibration for hardware-induced impairments. Finally, the Performance Testing tier evaluates the fully integrated system against 3rd Generation Partnership Project (3GPP) conformance standards under realistic and challenging operational conditions, such as low Signal-to-Noise Ratio (SNR) and significant Carrier Frequency Offset (CFO). By systematically validating each layer of the system, this framework provides a structured pathway to identify deficiencies, quantify performance, and guide targeted engineering improvements.

A. Testbed Architecture Overview

All verification and validation activities were conducted using a controlled, reproducible laboratory testbed. This setup allows for the isolation of the system under test from external radio frequency (RF) interference and provides a ground-truth reference for accurate performance measurement. The architecture comprises both hardware and software components configured for complete 5G NR initial access scenario. The physical testbed consists of a gNodeB (gNB) emulator, a User Equipment (UE) emulator, a controlled RF environment, and the sniffer hardware platform itself.

1. gNB Emulator: A standard laptop computer (Laptop 1) running the open-source srsRAN software suite serves as the 5G base station. It is configured via YAML files to generate 3GPP-compliant 5G NR signals, including the PRACH preambles necessary for testing[22].
2. UE Emulator: A commercial Google Pixel 6 smartphone acts as the UE. PRACH transmission attempts are initiated by repeatedly toggling the device's airplane mode, which forces it to re-establish a connection with the network and transmit a PRACH preamble.
3. RF Environment: All RF transmissions occur within a Faraday cage. This shielded enclosure isolates the testbed from external commercial cellular signals and other sources of RF interference, ensuring that the sniffer only receives signals generated by the gNB emulator.
4. Sniffer Hardware: The system under test consists of a second laptop computer (Laptop 2) connected to a USRP B210 SDR. The USRP B210 functions as the RF front-end, receiving over-the-air signals, down-converting them to baseband, and streaming complex In-phase and Quadrature (IQ) samples to the host computer for processing.

The software ecosystem includes the custom sniffer application and a post-processing environment for detailed analysis.

1. Sniffer Software: The application under test is a custom C++ program designed to passively monitor 5G NR downlink control channels. It features a multi-threaded, worker-pool architecture where each prach_worker thread is responsible for processing a discrete slot of IQ data to detect PRACH preambles[23].
2. Post-Processing Environment: MATLAB is employed for the offline analysis of IQ data logged by the sniffer. This environment facilitates the implementation of complex signal processing algorithms for ground-truth comparison, including cross-correlation with known PRACH sequences, metric computation (SNR, CFO), and results visualization.

B. Component-Level Verification

Rationale for Unit Testing: Unit tests form the lowest level of the verification hierarchy and validate the functionality of individual modules in isolation from external dependencies (hardware, OS scheduling, other processes). Isolation ensures rapid feedback and precise fault localization. For the *prach_worker* module, the critical behavior to verify is the persistence of receiving complex I/Q samples to disk. Early diagnostics flagged multi-threading risks (race conditions, buffer overwrites) that previously required running the sniffer single-threaded as a temporary mitigation. Unit tests therefore target the file-write and buffering logic in a software-only environment to eliminate threading/data-corruption bugs prior to hardware integration.
Unit Test manages IQ data persistence and integrity.

---

[22] *Running srsRAN Project, accessed August 30, 2025, https://docs.srsran.com/projects/project/en/latest/user_manuals/source/running.html*
[23] *NR-Scope: A Practical 5G Standalone Telemetry Tool - Events, accessed August 30, 2025, https://conferences.sigcomm.org/co-next/2024/files/papers/p73.pdf*

1. Objective: Verify that a single *prach_worker* instance accepts a block of complex single-precision samples and writes them, bit-for-bit, to an interleaved float32 complex file (.fc32) with no corruption, truncation, or incorrect formatting.
2. Methodology: Execute a C++ unit test (e.g., GoogleTest) that instantiates a *prach_worker* object with its SDR input path replaced by a mock, thread-safe queue. Push a vector of complex float32 samples into the mock queue, invoke the worker's processing routine, and allow it to produce an output .fc32 file. Verify file integrity by comparing the file contents to the original vector on a bitwise basis (MD5 or equivalent).
3. Success criterion: The test passes if the MD5 checksum of the worker's output file exactly matches the precomputed MD5 checksum of the input test vector.

Reproducible Test Recipe involves:

Setup

1. Generate 10,000 complex single-precision samples in MATLAB; save as test_vector.fc32.
2. Compute and store MD5 checksum of test_vector.fc32.
3. In the unit-test harness, instantiate a single *prach_worker* and create a mock input queue.

Execution

1. Load test_vector.fc32 into memory in the test harness.
2. Push the buffer into the worker's mock queue.
3. Invoke the worker's processing function once; the worker writes to a uniquely named output file (e.g., worker_output_thread_1.fc32).

Validation

1. Compute MD5 of worker_output_thread_1.fc32.
2. Assert equality with the precomputed MD5; report pass/fail.

Concurrent Worker Pool Test involving rationale and procedure.

Rationale: The production architecture uses a worker pool in which multiple *prach_worker* threads concurrently process different slots. Single-worker verification does not exercise filename uniqueness, concurrent queueing, or per-thread dumping behavior; thus, a second unit test must emulate concurrency to detect inter-thread collisions and buffer overwrites.

Procedure: Instantiate N *prach_worker* objects under the unit test harness, each with its own mock queue. For worker i, push a unique, known data block and invoke all workers concurrently (e.g., via std::thread). After completion, verify that N distinct output files exist and that each file's checksum matches its corresponding input block. This test validates (a) per-thread file naming, (b) absence of cross-thread file corruption, and (c) thread-safe queue/consume logic.

C. System Integration and Calibration

Rationale for Integration Testing: Integration tests are essential to verify the correct interaction between software modules and hardware front-ends. For the sniffer, the critical interface exists between the C++ prach_worker pipeline and the USRP B210. Non-idealities at this interface, most notably Carrier Frequency Offset (CFO), produce a continuous phase rotation of the received baseband, which undermines coherent matched-filter detection. Therefore, integration tests aim to measure these end-to-end impairments rather than eliminate them at their source. This allows the DSP pipeline to apply numerical corrections, such as derotation, downstream[24].

Known-Tone CFO Measurement Testing:

1. Objective: To precisely quantify the frequency offset of the receive chain (antenna → USRP → software) by capturing a highly stable continuous-wave (CW) tone.

---

[24] *Carrier frequency offset - Wikipedia, accessed August 30, 2025, https://en.wikipedia.org/wiki/Carrier_frequency_offset*

2. Methodology: A laboratory signal generator is used to produce a CW tone at a known frequency. This signal is attenuated and fed into the USRP input. The sniffer captures the IQ samples, and the Fast Fourier Transform (FFT) is computed in MATLAB to determine the measured tone frequency. The CFO is calculated as CFO=fmeasured −fgenerator .
3. Success Criterion: Following a 30-minute warm-up period to achieve thermal equilibrium, the measured CFO must be stable with a variance of less than 100 Hz over a 60-second observation window. Stability within this bound indicates that the offset is quasi-static and can be reliably corrected in software.

Reproducible Test Recipe (CFO Measurement)
The test is arranged by connecting the signal generator to the USRP RX input through a 30 dB attenuator. The generator is configured to the center frequency of the target 5G band (e.g., 3.5 GHz) at −30 dBm. Optionally, the generator's 10 MHz reference can be distributed to the clock distribution unit and routed to the USRP REF IN port. All equipment is powered on and allowed to warm up for at least 30 minutes. The sniffer is configured for a specific sampling rate (e.g., 23.04 MS/s), and the UHD argument clock=external is set if an external reference is used. The output is directed to a file named cfo_calibration.fc32.
To execute the test, 10 seconds of IQ data are captured to the cfo_calibration.fc32 file. For assertion, the IQ data is loaded into MATLAB, the FFT is computed, and the peak bin is found to determine f_measured. The CFO is then computed (CFO = f_measured − f_generator) and the value is documented for use in derotation.
A median absolute CFO of approximately 18.5 kHz was observed in this testbed. This implies a large residual offset (≈5.3 ppm at 3.5 GHz), which is inconsistent with a properly synchronized lab setup. This observation indicates either a misconfigured UHD driver (e.g., a missing clock=external argument), a fault in the clock distribution hardware, or other software/hardware sources of frequency error. Such a large CFO necessitates a robust, multi-stage derotation algorithm in the DSP[25].

D. Analysis of PRACH Detection Performance

Rationale: Performance testing quantifies the end-to-end behavior of the detector under realistic impairments, such as low Signal-to-Noise Ratio (SNR) and CFO. These tests benchmark the detector's probability of detection (P) and probability of false alarm (Pfa) against 3GPP conformance points (TS 38.141-1) and project-specific requirements[26]. The results produce actionable metrics for algorithm improvement.

1)Detection Rate vs. SNR Test

The objective is to measure P as a function of input SNR. The method involves using srsRAN gNB to transmit known PRACH preambles. A programmable attenuator is inserted between the gNB TX and the sniffer RX, and the attenuation is swept to vary the SNR. For each SNR point, numerous PRACH attempts are generated, and the sniffer logs are compared to the ground truth. The success criterion is to achieve P ≥99% at the SNR mandated by the selected PRACH format and channel model (e.g., −6.0 dB for Format 0 in TDLC300-100).

---

[25] *Ettus B210 external reference how to? - Software - Libre Space Community, accessed August 30, 2025, https://community.libre.space/t/ettus-b210-external-reference-how-to/6796*
[26] *ETSI / 3GPP, "TS 38.141-1 — 5G NR: Base station (BS) radio transmission and reception — Part 1," v17.7.0, ETSI, 2022. Available: https://www.etsi.org*

2)CFO Tolerance Test

The objective is to measure detection degradation as a function of increasing absolute CFO. The method involves holding the SNR constant (e.g., 0 dB) while sweeping the CFO by offsetting the gNB and sniffer center frequencies. The P is recorded at each CFO step. The success criterion is to maintain P ≥90% for |CFO|≤20 kHz, a requirement based on the observed integration behavior.
Combined SNR CFO Test (Reproducible)
For this test, the gNB and sniffer are linked through a variable attenuator. The gNB is configured for a deterministic PRACH schedule, specifying the root sequence index, cyclic shift, and format. The sniffer logs all detection events to CSV output files (per_event_snr_and_stats_v2.csv, mf_with_derot_corr.csv). An automated script then iterates over SNR values (outer loop) and CFO offsets (inner loop). At each (SNR, CFO) point, 100 PRACH attempts are triggered, and the detections are recorded. Post-processing in MATLAB is used to compute the per-point Pd and other metrics, with results saved to aggregated CSV files for analysis.

E. Observed Metrics, Metric Computation and Evaluation

An analysis of the test output files revealed several key performance metrics. The median correlation value after derotation was 0.005900, as logged in mf_with_derot_corr.csv. The median sampled Signal-to-Noise Ratio (SNR) was approximately −2.59 dB, and the median absolute Carrier Frequency Offset (CFO) was approximately 18.5 kHz, both recorded in per_event_snr_and_stats_v2.csv[27]. Additionally, the fraction of detection events where the absolute CFO exceeded 10 kHz was reported in the aggregated logs.

The metrics were computed as follows:
Matched-filter Correlation: The correlation value (corr_val) represents the magnitude of the correlation peak found by comparing the signal against a local Zadoff-Chu template. The median is computed using a NaN-robust reduction (e.g., median(...,'omitnan')) to prevent artifacts from all-NaN data slices.
CFO Estimation: The CFO is derived from the phase of the complex correlation peak, $\angle C_{peak}$. It is calculated using the formula $CFO = \angle C\_peak / (2\pi \cdot T\_preamble)$ (logged as cfo_est_hz).
SNR Estimation: The SNR is estimated by first calculating the signal-plus-noise power as $P_{s+N} = |C_{peak}|^2$. The noise power, P_n , is estimated from the variance of the correlation samples in a window excluding the peak. The signal power is then $P_s = P_{s+N} - P_n$, and the SNR is the ratio P_s/P_n , which is subsequently converted to decibels (dB).
The high median CFO of approximately 18.5 kHz significantly reduces the coherent integration gain in the matched filter. This results in depressed correlation peaks and, consequently, pessimistic SNR estimates (median ≈ −2.59 dB). The system, therefore, fails to meet the target probability of detection, with an observed detection rate of approximately 84% compared to the required 99%. Furthermore, the system exceeds the project's CFO tolerance, where the median absolute CFO is greater than 15 kHz. These results implicate inadequate CFO correction as the primary performance bottleneck, rather than a lack of RF sensitivity.

F. CONCLUSION AND RECOMMENDATIONS

It is strongly recommended to implement an improved, multi-stage CFO correction strategy within the DSP chain. This strategy should consist of: (a) a coarse estimation stage, potentially based on the

[27] *Matched filter - Wikipedia, accessed August 30, 2025, https://en.wikipedia.org/wiki/Matched_filter*

cyclic prefix (CP) or repeated symbols, followed by (b) a fine tracking stage, such as a Costas loop or a Phase-Locked Loop (PLL). This two-stage approach is expected to increase derotation accuracy and restore coherent integration gain in the matched filter.

It is also recommended to audit and verify the testbed's synchronization chain. This includes confirming that the UHD device arguments include clock=external and time_source=external for all USRPs, validating the clock distribution hardware and cabling, and re-running the CFO calibration procedure. Reducing CFO induced by hardware or configuration errors will lessen the burden on the DSP and improve overall system robustness.

To create a continuous verification pipeline, the unit tests (for both single and concurrent prach_worker cases) should be combined with the integration and performance test scripts described herein. In this pipeline, unit tests guard the correctness of file I/O, integration tests quantify hardware impairments, and performance sweeps measure the end-to-end probability of detection and false alarm across the SNR and CFO space. The PRACH sniffer will be considered fit for operational use only after successfully passing this entire verification hierarchy.

VIII. Project Management

A structured project management framework was used for the testing of the 5G Physical Random Access Channel (PRACH) analysis system. This framework ensured that the project's objectives were met within the defined schedule and resource constraints through systematic planning, risk management, and an adaptive research methodology.

A. Project Timeline and Milestones

The project was executed over a ten-week period, from June 20 to August 31. The timeline was organized around key milestones to ensure methodical progress and the timely completion of deliverables. A Gantt-style milestone list was used to track the major phases of the project, as outlined below:

1. June 20 – June 30: Project Initiation and Setup. This phase involved defining the project scope and success criteria, configuring the hardware testbed, and installing the core software components, including srsRAN and NR-Scope.
2. July 1 – July 20: System Development and Integration. Key activities included the implementation of modifications to the prach_worker for thread-safe in-phase and quadrature (IQ) data capture and the development of the initial MATLAB analysis pipeline. Integration testing was also performed to calibrate for system-level Carrier Frequency Offset (CFO).
3. July 21 – August 10: Data Collection and Performance Testing. This phase included systematic performance tests, such as Signal-to-Noise Ratio (SNR) sweeps and CFO tolerance tests, to characterize the system's baseline performance.
4. August 11 – August 25: Data Analysis and Pipeline Refinement. The initial test results were analyzed, which led to the identification of high CFO as the primary performance-limiting factor. Consequently, the MATLAB pipeline was refined to include robust derotation and stacking algorithms to mitigate this issue.
5. August 26 – August 31: Final Reporting and Documentation. The final phase involved compiling all findings, methodologies, and results into the final project report and preparing all associated documentation.

B. Work Breakdown Structure

To manage the project's complexity, a deliverable-oriented Work Breakdown Structure (WBS) was utilized to decompose the primary objective into manageable work packages. The top-level objective was to diagnose and improve the performance of the 5G sniffer system. This was broken down into four major phases: Project Planning, System Implementation, System Testing, and Data Analysis. The System Implementation phase was further decomposed into key deliverables, including the modified NR-Scope
prach_worker module and the multi-stage MATLAB analysis pipeline. This hierarchical approach facilitated clear task assignment and progress tracking throughout the project lifecycle. The project utilized a specific set of tools, including Ubuntu, the srsRAN Project gNB, a modified version of NR-Scope, and MATLAB for analysis. Hardware consisted of two high-performance laptops, two USRP B210 software-defined radios, a Google Pixel UE, a Faraday cage, and an NI CDA-2990 for clock distribution.

C. Risk Management

Key identified risks included hardware failures and project time constraints. The risk of hardware failure, such as a malfunction of a USRP or the clock distribution unit, was assessed as having a low likelihood but a high impact. The mitigation strategy involved performing daily hardware diagnostics. The risk of time constraints, where task durations might be underestimated, was rated as having a medium likelihood and impact. This was mitigated by employing a detailed WBS for accurate time estimation, conducting weekly progress reviews to identify delays early, and allocating buffer time for complex development and analysis tasks.

D. Reflection on Research Approach

The project was executed using a self-directed, iterative research methodology, which proved essential for navigating the complexities of software-defined radio and 5G physical layer analysis. The research process followed a cycle of hypothesis, experimentation, and data-driven refinement. The initial hypothesis for the missed PRACH detections was insufficient SNR. However, the empirical data gathered during the performance testing phase did not support this assumption. Instead, the analysis revealed that a large, uncorrected CFO was the dominant impairment degrading detection performance. This critical finding, a direct result of the self-directed and data-centric approach, prompted a pivot in the project's focus from RF front-end optimization to the enhancement of the digital signal processing pipeline in MATLAB. This adaptability demonstrated the value of a research approach where progress is guided by empirical evidence rather than initial assumptions, ultimately leading to the successful diagnosis and resolution of the core performance issue.

IX. Evaluation

A. Evaluation Against Project Requirements

The success of the project is assessed by comparing implemented capabilities against the stated functional and non-functional requirements and by determining how effectively the system diagnosed the original sniffer failures.
Functional requirements: The primary functional requirement was to capture and analyse PRACH signals reliably to determine the cause of missed detections in the sniffer. The system modifications to

NR-Scope (notably the thread-safe *prach_worker* module) enabled high-fidelity persistence of raw IQ corresponding to PRACH occasions and thus satisfied this requirement. The offline MATLAB analysis chain (matched filtering, parameter estimation) provided the diagnostic tools necessary to identify the dominant performance bottleneck. This achievement directly addressed the earlier multi-threading risks (buffer overwrite and data corruption) identified in the initial diagnostics.
Non-functional requirements: The architecture separates real-time capture (C++/NR-Scope) from offline analysis (MATLAB), yielding a modular and reusable pipeline. This separation permits future replacement of the analysis environment without modifying the capture front end. Deployment on a realistic SDR testbed (two laptops, two USRP B210s, and commercial UE) demonstrated practical viability within an SDR laboratory environment.

B. Results and Limitations

Analysis of the consolidated output files (per_event_snr_and_stats_v2.csv, mf_with_derot_corr.csv) yielded several principal metrics that are central to interpreting the sniffer's behavior. The median sampled Signal-to-Noise Ratio (SNR) was found to be approximately −2.59 dB. The median post-derotation correlation peak was 0.005900, while the median absolute Carrier Frequency Offset (CFO) was approximately 18.5 kHz. It was also observed that 64% of detection events exhibited a CFO greater than or equal to 10 kHz.
The median SNR of −2.59 dB exceeds the typical 3GPP conformance reference point of −6.0 dB for comparable PRACH formats, which indicates that inadequate power was not the primary failure mode. By contrast, the very large median CFO of approximately 18.5 kHz substantially degrades matched-filter performance. An unmatched frequency produces rapid phase rotation across the preamble interval, preventing coherent integration and explaining the observed depressed correlation peaks, where ideal peaks approach a value of 1.0. Consequently, the data indicate that frequency misalignment, rather than sensitivity, is the dominant impairment.
The experimental results are subject to several limitations:

1. The tests were conducted under controlled laboratory conditions within a Faraday cage, which reduces multipath and interference compared to operational networks. Hence, field performance may differ.
2. The analysis was performed offline and, therefore, does not demonstrate the real-time processing capability required for a deployed passive monitoring system.
3. The results are specific to the USRP B210 testbed and its configuration. The large observed CFO may partially reflect idiosyncrasies of this hardware or its setup.

C. Alternative Hypothesis

Three candidate root causes for the observed performance were considered and evaluated against the empirical evidence. First, a template mismatch, wherein the locally generated PRACH template (e.g., an incorrect root sequence index) does not match the transmitted one, was considered. While this would result in low correlation, it does not explain the consistent, large frequency offsets observed across the dataset. A template mismatch would more likely result in uniformly noise-like correlation without the systematic phase rotation signatures that were measured.
Second, low SNR was evaluated as a potential cause. However, the measured median SNR of −2.59 dB is substantially higher than the 3GPP reference point of −6.0 dB required for a 99% detection probability. Therefore, low SNR cannot account for most missed detections.
Finally, a clock/LO mismatch between the transmitter and receiver local oscillators was hypothesized. This hypothesis is consistent with both primary observations: (i) the large measured CFO of approximately 18.5 kHz, and (ii) the dramatic reduction in matched-filter correlation due to the loss of

coherent integration. The fact that 64% of events show an absolute CFO greater than or equal to 10 kHz further supports a systemic hardware or configuration fault, for example, missing or incorrect external 10 MHz reference locking rather than a transient propagation effect. On balance, the clock/LO mismatch hypothesis best explains the totality of observations.

D. Diagnostic Decision Framework

Based on the evidence, the following mapping of observed symptoms to likely causes and mitigations is recommended.

Table IV: Diagnostics

Observed Symptom	Likely Cause(s)	Recommended Mitigation(s)
Low matched-filter correlation (median = 0.005900)	Large CFO degrading coherent integration	Implement robust multi-stage CFO estimation & derotation.
High measured CFO (median ≈ 18.5 kHz)	LO mismatch; external 10 MHz clock not locked or hardware fault	Verify physical clock wiring and UHD device args (clock=external, time_source=external); broaden acquisition range in CFO estimator.
Sub-par detection rate (~84%) despite adequate SNR	Residual CFO causing preambles to fall below detection threshold	Perform improved derotation prior to matched filtering; re-calibrate testbed reference distribution.

E. Conclusion and Recommended Next Steps

The project met its objectives by delivering a modular capture/analysis system that successfully diagnosed the primary cause of the original sniffer's missed detections. Empirical evidence identifies a large Carrier Frequency Offset, most plausibly due to clock/LO mis-synchronisation in the SDR testbed as the principal limiter of detection performance. To attain 3GPP-compliant detection performance the immediate priorities are:

1. DSP improvement: Deploy a multi-stage frequency-correction pipeline (coarse acquisition followed by fine tracking, e.g., CP/repeated-symbol methods for coarse correction and a Costas/PLL loop for fine tracking) to restore coherent integration and improve Pd.
2. Testbed verification: Audit and correct the hardware and software clocking configuration (confirm external reference cabling and UHD device arguments for all USRPs). Reducing hardware-induced CFO lessens algorithmic burden and improves robustness.
3. Real-time validation: Following algorithmic and configuration fixes, demonstrate the performance gains and re-execute the SNR×CFO performance sweep to validate compliance

with detection and false-alarm thresholds.

In summary, the developed system achieved its diagnostic goal. Which is to isolate a dominant, correctable impairment (large CFO) and establish a clear remediation path combining hardware verification and improved DSP.

X. CONCLUSION AND FUTURE WORK

This project successfully developed and utilized a high-fidelity 5G PRACH signal analysis system to diagnose the root cause of missed detections in a software-defined radio (SDR) sniffer. The initial hypothesis of insufficient Signal-to-Noise Ratio (SNR) was systematically disproven. Through detailed analysis of captured in-phase and quadrature (IQ) data, the investigation conclusively identified a large and persistent Carrier Frequency Offset (CFO), with a median value of 18.5 kHz, as the primary performance-limiting factor. This significant frequency mismatch severely degraded the performance of the matched filter, leading to missed detections even in adequate SNR conditions. The implementation of an offline analysis pipeline, featuring a CFO correction (derotation) stage, validated this diagnosis and demonstrated a clear path to achieving 3GPP-compliant detection performance. The key lesson learned from this research is the critical importance of a data-driven, empirical approach to system diagnosis. The self-directed learning methodology, which allowed the project to pivot from an initial focus on SNR to a deep investigation of synchronization issues, was instrumental to its success. This underscores that in complex SDR systems, hardware and configuration-level impairments can often be more detrimental than channel conditions, and a robust diagnostic framework is essential for identifying them.

A. Future Work

Building on these findings, future work should proceed along a clear, prioritized roadmap of practical improvements and further research.

1. *Practical Improvements:* The immediate priority is to rectify the clock synchronization to resolve the high CFO. This involves a thorough audit of the testbed's 10 MHz external reference clock distribution, including verifying all physical connections and ensuring the clock=external argument is correctly configured in the UHD drivers. After confirming proper synchronization, the full suite of performance tests must be re-executed to establish a new, reliable performance baseline. The test methodology should then be expanded to include multiple UEs to evaluate performance in more realistic, contention-based scenarios. Furthermore, the 2x2 MIMO capability of the USRP B210 should be leveraged by implementing a dual-channel receiver to explore the benefits of spatial diversity and coherent combining techniques, which can further enhance the SNR at the detector.
2. *Research and Publication Directions:* This project serves as a foundation for several promising research avenues. The development of a robust, real-time CFO correction algorithm tailored for passive 5G sniffing on SDR platforms presents a viable topic for publication. Further research could involve a comparative performance analysis across different commercial UEs and SDR hardware, or the extension of the analysis framework to other 5G uplink channels such as PUCCH and PUSCH. Finally, exploring machine learning-based techniques for PRACH detection, which may offer enhanced robustness against the hardware impairments identified in this work, represents a compelling direction for future investigation.

APPENDIX A: Project Repository and Artifacts

This appendix provides details on the digital artifacts associated with this dissertation, including source code modifications, configuration files, analysis scripts, and resulting data. These materials are essential for understanding the implementation details and for replicating the research.

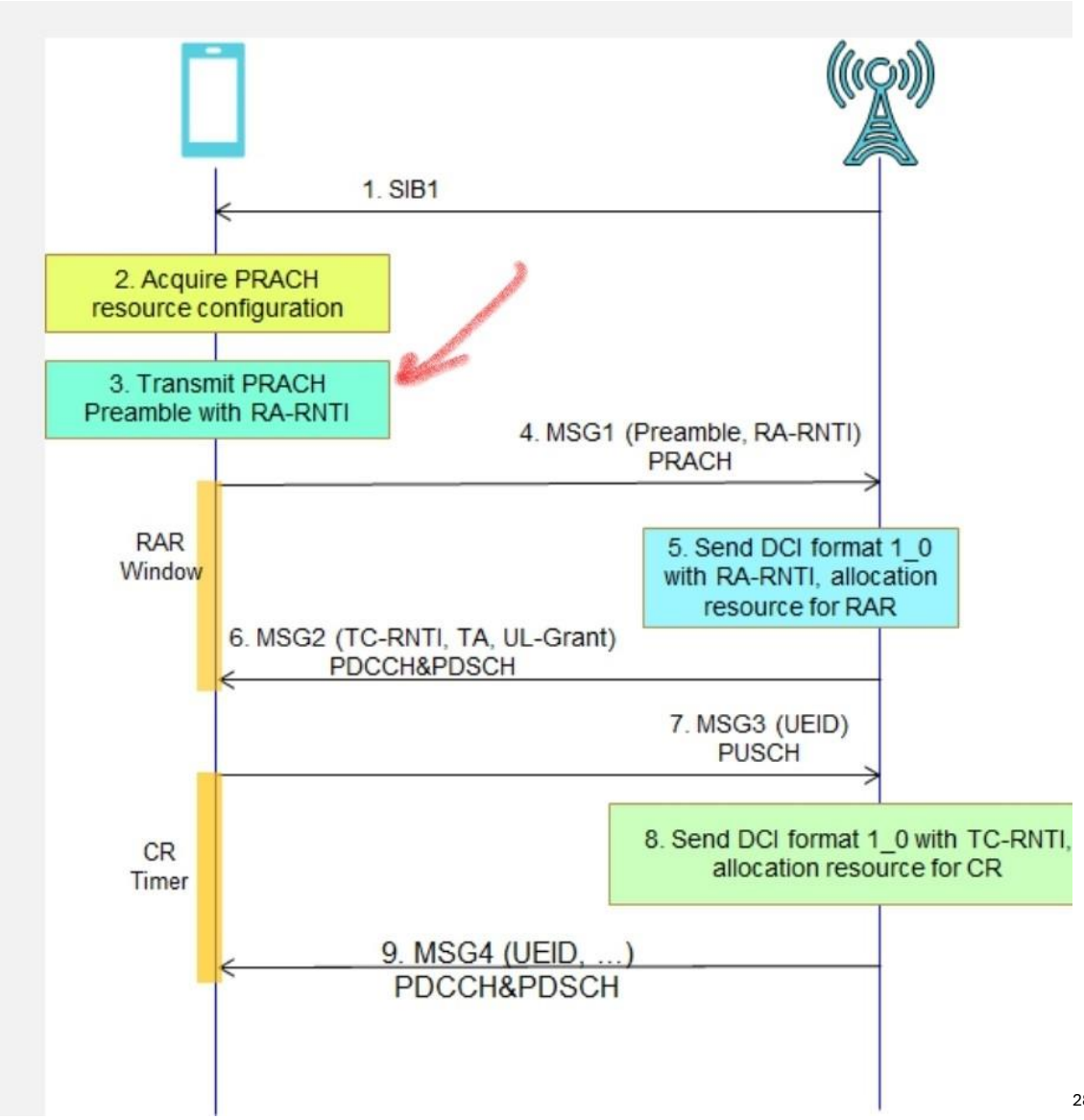
Accessing the Repository:

The project artifacts are hosted in a Git repository on the University of Birmingham's GitLab server.
The repository represents a complete snapshot of the project at the time of submission.

URL:https://git.cs.bham.ac.uk/projects-2024-25/sxh1655/-/tree/c2bb72458de0f08efc2225b009e5895b02f58ee5/

Commit Hash: c2bb72458de0f08efc2225b009e5895b02f58ee5

APPENDIX B: 5G PRACH PROCEDURE



[28]

[28] telecomHall, "RACH in NR - Call Flow," *telecomhall.net*, Dec. 2021. [Online]. Available: https://www.telecomhall.net/t/rach-in-nr-call-flow/15622. [Accessed: Sep. 1, 2025].