UNIVERSITÀ DEGLI STUDI DI PALERMO

PhD course in Information and Communication Technologies
Department of Engineering

From Passive Sniffing to Active Signal Injection: Exploiting Physical Layer Vulnerabilities in 5G New Radio.

AUTHOR
Ing. Alessandra Dino

COORDINATOR
Prof. Marco La Cascia

TUTOR
Prof. Ilenia Tinnirello

CO-TUTOR
Prof. Stefano Mangione

XXXVIII Cycle
ACADEMIC YEAR 2024 - 2025

Abstract

5G New Radio introduces unprecedented performance and flexibility, but its Physical layer still exposes critical information and control signals that remain vulnerable to observation and manipulation. This thesis demonstrates that these exposures can be systematically exploited using low-cost, open-source tools, enabling both high-fidelity passive analysis and precise active attacks.

We develop a custom 5G sniffer capable of decoding control- and user-plane data, and reference signals, providing fine-grained visibility into downlink scheduling, modulation, resource allocation, and channel-state reporting. Using this visibility, we show that passive metadata alone suffices to classify application-level traffic with high accuracy, revealing user behavior despite encrypted payloads.

Leveraging frame-synchronized actuation, we then escalate to active attacks. We demonstrate active attacks that force MIMO rank reduction and a DCI spoofing attack that manipulates scheduling decisions and measurably alters UE energy consumption. Both attacks operate without breaking encryption and remain feasible on commercial 5G networks.

These findings show that the 5G Physical layer constitutes a practical attack surface, where leakage and manipulability persist despite higher-layer security. The thesis also outlines countermeasures that harden control signaling and detection mechanisms without compromising 5G performance.

Contents

List of Figures

List of Tables

Derived Works

This thesis builds upon the following published and ongoing works:

- Mangione, S., Dino, A., Garbo, G., & Croce, D. (2024, June). Crystal oscillator error compensation in software defined radios for 5G network testbeds. In 2024 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit) (pp. 1-5). IEEE.
- Alaimo, R., Corallo, R., Schilleci, S., Dino, A., Mangione, S., Tinnirello, I., & Garlisi, D. (2025). Undercover Disruption: Stealth Jamming Attacks on 5G Synchronization Stages. In CEUR WORKSHOP PROCEEDINGS (Vol. 3962). https://ceur-ws.org/Vol-3962/.
- Dino, A., Giuliano, F., Mangione, S., Garlisi, D., & Tinnirello, I. (2025, November). Silent Drain: From Energy Profiling to Practical Denial-of-Energy Attacks in 5G. In Proceedings of the ACM Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization (pp. 113-120).
- Vicario, A., Pagano, A., Dino, A., Croce, D., & Tinnirello, I. (2025, November). Towards Intelligent Mobility Management: Customized Handover in 5G O-RAN Networks. In Proceedings of the ACM Workshop on Wireless Network Testbeds, Experimental evaluation & Characterization (pp. 65-72).
- Palamà, I., Mangione, S., Dino, A., Focarelli, G., & Bianchi, G. (2025). Golden Sniffer: Low-Complexity 5G New Radio Signal Interception Exploiting Gold Sequences. Accepted at ACM Mobicom 2026.
- Dino, A., Giuliano, F., Mangione, S., Tinnirello, I., & Garlisi, D. (2025). Hijacking 5G MIMO: a Downgrade Attack via Tampered Zero-Power Channel State Information. Accepted at IEEE INFOCOM 2026.
- Dino, A., Palamà, I., Mangione, S., Tinnirello, I., & Bianchi, G. (2025). Hidden but Exposed: Deep Learning-Based Identification of User Activity in 5G New Radio. Submitted to IEEE Transaction on Wireless Communication.

Introduction

5Gth generation of mobile communication systems is an important step in the development of wireless technologies. Unlike previous generations, which were designed to provide mainly consumer-oriented services such as voice, video, and broadband data, 5G has been designed as a multi-service infrastructure for a wide variety of applications, including massive machine-type communications (mMTC), and ultra-reliable low-latency services (uRLLC). This heterogeneity of use cases results in a diversified set of performance requirements: certain use cases demand peak data rates of the order of Gbps, others rely on end-to-end latencies of only a few milliseconds, and some applications require extreme energy efficiency to guarantee device lifetimes of several years. To address these requirements, 5G embraces architectural paradigms such as Network Function Virtualization (NFV), Software-Defined Networking (SDN), and network slicing, enabling elastic and dynamically reconfigurable allocation of network resources. At the radio layer, capacity is boosted through the use of higher frequency bands and advanced multi-antenna techniques (e.g. massive Multiple-input-multiple-output (MIMO)), which mitigate coverage limitations and improve spectral efficiency.

These innovations make 5G significantly more capable than previous generations, but they also open new dimensions of vulnerability. Traditional security analyses have concentrated on cryptographic protections applied to the Control Plane and User Plane, often assuming that encrypted payloads sufficiently preserve confidentiality. However, this assumption no longer holds when considering the Physical (PHY)-layer. Even with strong cryptography, the PHY-layer remains an exposed source of informations. Passive Over-the-Air (OTA) monitoring can reveal the number of active UEs in a cell, infer application-level behavior through traffic classification, or expose beamforming and antenna-usage characteristics. Furthermore, the radio interface itself enables active adversarial actions: attackers can influence a device's link adaptation, force antenna-rank changes, or perform Denial of Energy (DoE) attacks that deliberately drain battery resources.

A key enabler of this work is the increasing availability and maturity of open-source 5G software stacks, decoding tools, and SDR (Software Defined Radio)-based frameworks, which have made it practically possible to observe, analyze, and manipulate 5G signals at low cost. These tools allow researchers to perform fine-grained PHY measurements, decode control-channel metadata, and implement synchronized active interventions without relying on proprietary equipment. In short, open-source ecosystems have transformed what used to be theoretical analyses into reproducible, hands-on experiments using commodity hardware.

The objective of this thesis is therefore to leverage this open-source instrumentation to systematically study what can be inferred from PHY-layer observations, and to evaluate the feasibility and real-world impact of both passive and active attacks against 5G's radio interface. The goal is to expand the security perspective on 5G by explicitly integrating PHY-layer vulnerabilities into comprehensive threat models and by discussing potential mitigation strategies. Specifically, this work centers around three guiding research questions:

- *What information about user activity and network behavior can be extracted from passive observation of PHY-layer signals?*
- *To what extent can active manipulations of the radio interface, implemented with low-cost SDR platforms, alter UE behavior, resource usage, or energy efficiency?*
- *Which countermeasures can effectively mitigate these threats without compromising the flexibility and performance goals of 5G?*

Thesis Organization: The remainder of this thesis is organized as follows.
Chapter 1 reviews the fundamentals of 5G NR, providing an overview of network architecture, the Radio Access Network (RAN) and Core Network concepts that are required to understand the subsequent experimental work. Chapter 2 presents the design and implementation of a purpose-built 5G sniffer and describes the decoding of control and user-plane channels, as well as reference signals including Non-Zero-Power CSI-RS (NZP-CSI-RS) and ZP-CSI-RS, thereby establishing the practical limits of passive PHY observability. Chapter 3 applies these passive decoding capabilities to Traffic Flow Confidentiality (TFC) experiments: we demonstrate how decoded control/user-plane metadata can be used to perform robust traffic classification both within the same application category and across different categories. Chapter 4 leverages ZP-CSI-RS sniffing to design and evaluate PHY-level active attacks, including a jamming scenario that forces spatial-rank reduction (e.g., rank 4 to rank 1), and quantifies the resulting impact on channel-state estimation and throughput. Chapter 5 builds on the experimental platform to explore DoE and related active manipulation techniques, integrating frame-synchronized actuation with PDCCH decoding to demonstrate end-to-end control-plane attacks such as DCI spoofing. Finally, Chapter 5.6 summarizes the main findings, discusses limitations, and outlines open issues and directions for future research.

Thesis Contributions: This thesis makes both methodological and practical contributions to the study of 5G PHY-Layer security, with a strong emphasis on low-cost experimental reproducibility. The main practical outcomes of this work are:

- **Low-cost 5G experimental platform.** We design and validate a modular 5G SA testbed based on commercial SDRs and open-source software, introducing calibration procedures for error oscillator compensation that enable stable PHY-layer experimentation even with low-cost hardware.

- **Custom 5G NR sniffer.** We develop a dedicated sniffer capable of decoding PDCCH, PDSCH, DM-RS, NZP-CSI-RS and ZP-CSI-RS, supporting multi-BWP operation and full control/user-plane metadata extraction. Support for real-time decoding is currently under development, as we work toward achieving continuous, low-latency PHY-layer observability.
- **Jamming and active interference toolkit.** We implement a set of controlled jamming primitives (beam degradation, rank reduction, and ZP/NZP attack) with frame-level synchronization, enabling reproducible PHY-layer attack experiments.
- **Traffic-obfuscation and TFC evaluation framework.** We introduce a methodology to correlate PHY-layer metadata with application-layer behavior under encryption, enabling systematic assessment of Traffic Flow Confidentiality countermeasures.
- **End-to-end Denial-of-Energy attack implementation.** We build the first practical DoE attack tool for 5G SA, integrating sniffer-side DCI decoding with real-time waveform injection and UE energy profiling, demonstrating synchronized uplink-grant spoofing and connection-state forcing.

Overall, the thesis provides not only conceptual insights into 5G PHY-layer vulnerabilities, but also a complete set of reproducible tools (sniffer, jamming modules, calibration routines, and DoE attack scripts) intended to support future research and facilitate security analysis on commodity hardware.

Chapter 1

Fundamentals of 5G

The study of 5G NR requires a solid understanding of how it differs from the design principles of the previous generation (Long Term Evolution (LTE)). While both generations share the same general objective of providing efficient wireless access, their architectures, protocol stacks, and physical layer implementations differ in several fundamental aspects. A direct comparison of these characteristics provides a useful reference framework, allowing the reader to appreciate not only the specific innovations of 5G, but also the motivation behind them when contrasted with the LTE baseline. Table 1.1 summarizes the key technical differences that will be examined in detail throughout this chapter.
Each of the highlighted dimensions will be revisited and analyzed in detail: the architectural evolution from Envolved Packet Core (EPC) to the service-based 5G Core (5GC) (Section 1.1.1), the introduction of the gNB and its functional split (Section 1.1.2), and the new capabilities of 5G UE (Section 1.1.3) are discussed in Section 1.1. Subsequently, the protocol stack and the PHY-layer are examined in Section 1.2, where aspects such as the flexible frame structure, scalable numerology, advanced modulation and coding schemes are explored in direct comparison with their LTE counterparts. This organization ensures that the reader can trace each improvement of 5G NR back to its LTE baseline, emphasizing both continuity and innovation across generations.

1.1 5G Network Architecture

A mobile telecommunication system has three main components: the CN, the RAN and the user's device, known as UE. The CN is responsible for transporting traffic between mobile devices and external networks, such as the Internet. In addition, it manages the control of communications with these external networks and maintains subscriber-related information on behalf of the network operator. The RAN manages the radio communications between the mobile device and the network. It connects to the CN through the backhaul interface, and to the UE via the air interface, also called radio interface. On this interface, the transmission from the network to the device is defined as the DL, whereas the transmission from the device to the network is the uplink (UL). The UEs are every terminals (smartphones, laptop, IoT

Feature	4G LTE	5G NR
Core Architecture	EPC (Evolved Packet Core), hierarchical and node-centric	5GC (Service-Based Architecture), modular and service-oriented with control/user plane separation
Radio Access	eNodeB (eNB)	Next Generation NodeB (gNB) with CU/DU split
Spectrum usage	Sub-6 GHz (mainly below 3.5 GHz)	Sub-6 GHz and mmWave (up to $\sim$100 GHz)
Channel bandwidth	Up to 20 MHz	Up to 400 MHz
Peak data rate	500–1000 Mbps (UL/DL)	10–20 Gbps (UL/DL)
Latency	$\sim$10 ms	$<$1 ms
Multiple access	OFDMA (DL), SC-FDMA (UL)	OFDMA for both DL and UL
Frame structure	Fixed 15 kHz subcarrier spacing	Scalable numerology
MIMO support	Up to 8$\times$8 MIMO	Massive MIMO up to 64 antennas
Physical channels	PDCCH, PDSCH, PUSCH, PUCCH (LTE design)	Similar set but redesigned with flexible mapping, multiple reference signals (CSI-RS, DMRS)
Network slicing	Not supported	Native support, logical networks on shared infrastructure
Energy efficiency	Optimized but limited	Advanced DRX cycles, RRC_INACTIVE, better support for low-power IoT
Use cases	Primarily mobile broadband	eMBB, mMTC, uRLLC (triple service model)

Table 1.1: Comparison between 4G LTE and 5G NR

devices, ...) connected to the network.

1.1.1 5G Core Network (5GC)

The 5G CN builds upon the 4G CN concepts. The EPC, shown in Figure 1.1, is composed by four main modules:

- Mobility Management Entity (MME) that manages both the communication between the mobile and the EPC and sessions with external network. Its database is represented by Home Subscriber Server (HSS), which contains subscribers' information.
- Packet Data Network Gateway (PGW) delivers incoming and outgoing traffic.

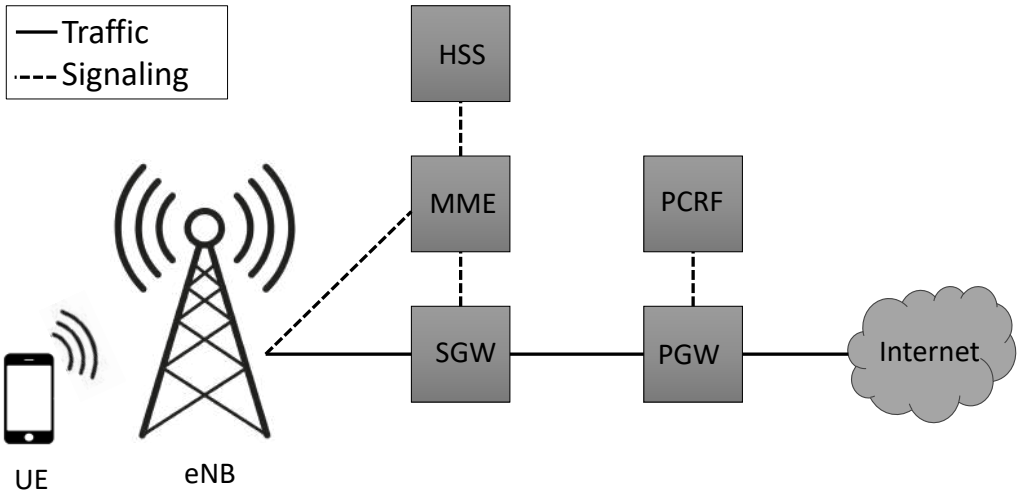


Figure 1.1: Structure of 4G LTE network architecture.

- Serving Gateway (SGW) forwards mobile's traffic between the RAN and the PGW.
- Policy and Charging Rules Function (PCRF) sets the network policies (such as maximum data rate).

From Release 14, there is a separation between Control and User Plane. This architectural choice provides several concrete benefits. First, it allows network operators to scale the control and user planes independently, resulting in greater resource efficiency. Second, user-plane functions can be geographically distributed closer to end users, reducing latency and enabling edge deployments, while control functions remain centralized. Finally, it represents a step towards the architecture of the 5GC, which facilitates interoperability and ensures a smoother transition between EPC- and 5GC-based deployments.

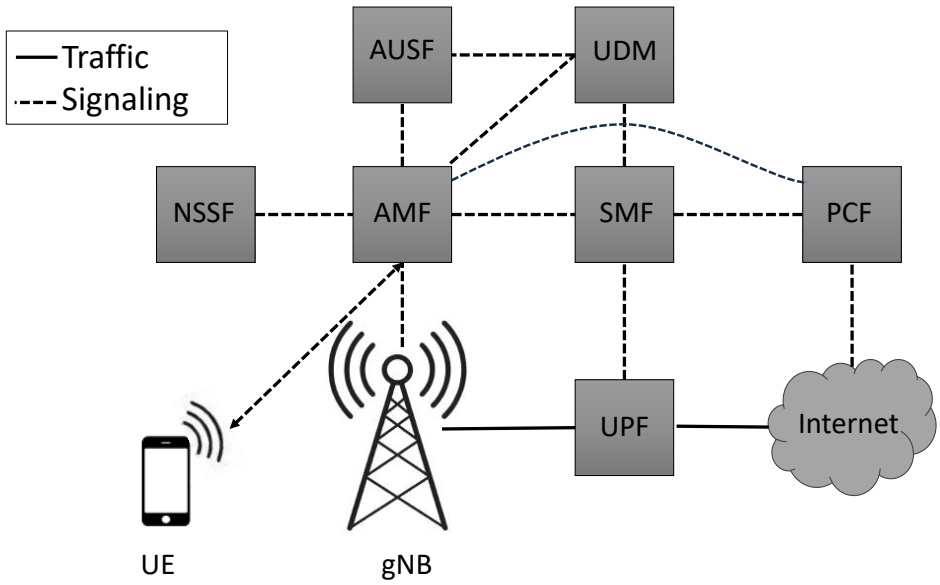


Figure 1.2: Structure of 5G NR network architecture.

The 5G CN departs from the monolithic architecture of the EPC and introduces a service-based, cloud-native design. Instead of rigid nodes, the CN is built around modular Network Functions (NFs), each of which provides specific services accessible through standardized interfaces. The fundamental components, shown in Figure 1.2, are:

- Access and Mobility Management Function (AMF), which manages the interactions between the UE and 5G CN.
- Unified Data Management (UDM) that has the same role of HSS of EPC.
- Session Management Function (SMF) manages the data session (PDU).
- Policy Control Functions (PCFs) has the same role of PCRF of EPC.
- Authentication Server Function (AUSF) manages the authentication procedures.
- Network Slice Selection Function (NSSF) selects, for each user, the logical region of the cell (called slice) where the user will register.
- User Plane Function (UPF) is responsible for packet routing and forwarding, packet inspection, QoS handling in the 5G architecture.

In 5G CN, user connectivity is established through Packet Data Unit (PDU) session, that corresponds to an association between a UE and a data network through a UPF. It can transport IP packets, Ethernet frames, or unstructured data, thus offering greater flexibility compared to LTE bearers, which were restricted to IP. Each PDU session is managed in the Control Plane by the SMF and realized in the User Plane by UPFs.

Moreover, the 5G CN supports non-3GPP technologies access, such as Wi-Fi or LoRa. The Non-3GPP Interworking Function (N3IWF) acts as the gateway, establishing secure IPsec tunnels between the UE and the 5GC. This design allows mobility and session continuity across heterogeneous technologies.

One of the most distinctive features of the 5G CN is network slicing. A slice is a logically instance of the network, tailored to the requirements of a specific service or customer. This mechanism allows the 5GC to flexibly allocate resources for enhanced Mobile Broadband (eMBB), massive Machine-Type Communication (mMTC), or ultra-reliable low-latency communications (URLLC), coexisting within the same infrastructure.

To protect subscriber privacy, 5G CN replaces the plain transmission of the International Mobile Subscriber Identity (IMSI) with the Subscription Concealed Identifier (SUCI). The SUCI is generated by the UE using the public key of the home network and sent to the AMF during registration. Only the home network's Subscription Identifier De-concealing Function (SIDF) can recover the IMSI. This mechanism mitigates IMSI-catching attacks that were possible in LTE, thereby improving user anonymity and resilience against active attackers.

1.1.2 Next-Generation NodeB (gNB)

The Base Station (BS) of a 5G network is called gNB; at the beginning, it was installed in the LTE RAN (E-UTRAN), and communicated with the EPC. Now, the gNB is installed in the next-generation radio access network (NG-RAN). In LTE, the BS was a single network element; in 5G NR, 3GPP splits it in several parts [1], as shown in Figure1.3[1], namely the Central Unit (CU), the Distributed Unit (DU), and the Radio Remote Head (RRH).

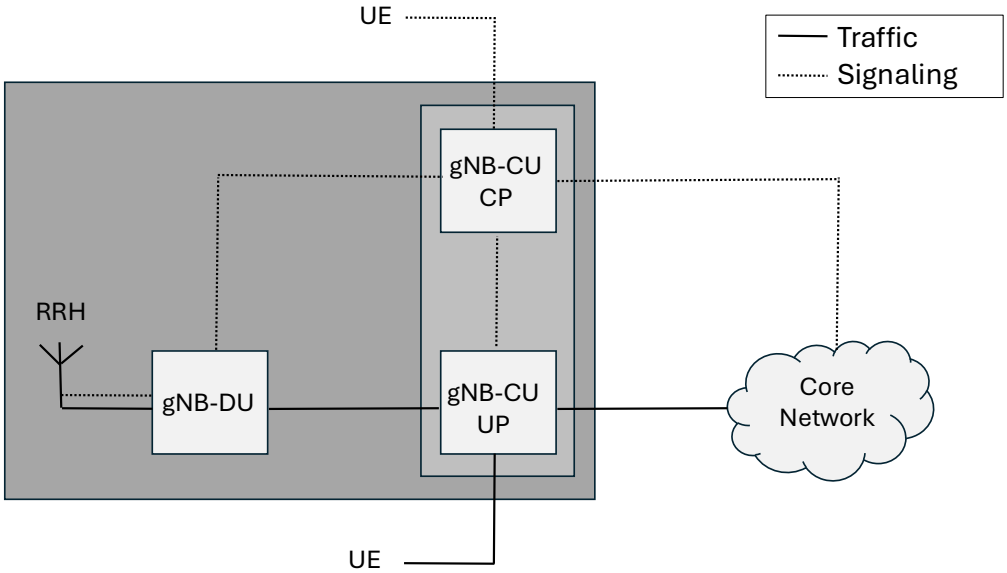


Figure 1.3: Structure of 5G NR Base Station.

Central Unit The CU hosts higher-layer protocols of the radio stack and can be further divided into Control-Plane and User-Plane components:

- **CU-CP:** responsible for Radio Resource Control (RRC) signaling and Service Data Adaptation Protocol (SDAP) *control functions*, meaning it manages the configuration and setup of SDAP mappings between QoS flows (QFIs) and Data Radio Bearers (DRBs). It handles the signaling needed to establish, modify, or release these mappings and overall UE context management. The CU-CP interfaces with the AMF in the 5GC for registration, mobility, and connection procedures.

- **CU-UP:** responsible for the *user-plane handling* of SDAP along with PDCP. It enforces the SDAP mappings configured by the CU-CP, applies QoS rules to actual data packets, and ensures reliable delivery of data. The CU-UP interfaces with the UPF in the 5GC via the N3 interface, delivering data according to the QoS flows defined by SDAP.

The CU is typically deployed in a centralized cloud or data center, allowing flexible resource pooling and coordination among multiple DUs.

[1]Cox, C. (2025). An Introduction to 5G: The New Radio, 5G Network, 5G Advanced and Beyond. John Wiley & Sons.

Distributed Unit The DU hosts the lower layers of the radio protocol stack, including the Radio Link Control (RLC), Medium Access Control (MAC), and parts of the PHY-layer. It is primarily responsible for time-critical functions such as scheduling, Hybrid Automatic Repeat Request (HARQ) processing, and radio resource allocation. By placing the DU closer to the cell sites, operators can minimize latency while still benefiting from centralized CU coordination. The CU and DU are interconnected via the standardized F1 interface.

Radio Remote Head The RRH, referred to as the Radio Unit (RU), contains the Radio-Frequency components and the antenna elements. Its primary role is the transmission and reception of radio signals over the air interface. The RRH connects to the DU via high-capacity, low-latency fronthaul links. This split allows the RRH to be deployed flexibly at cell sites while the DU and CU can be virtualized and hosted in edge or centralized cloud environments.

The functional split of the gNB into CU, DU, and RRH provides multiple advantages. It enables scalable deployments, where the CU can control many DUs, and each DU can be associated with multiple RRHs. Furthermore, it supports cloud-native virtualization, allowing operators to place functions optimally between centralized data centers and edge nodes. This architecture is key to supporting low-latency services, massive connectivity, and flexible spectrum usage in 5G networks.

1.1.3 5G User Equipment (UE)

In 5G systems, the term UE encompasses a wide variety of end devices that access the network through either 3GPP or non-3GPP access technologies. Compared with LTE, the 5G ecosystem significantly expands the range of compatible devices well beyond conventional smartphones. The main categories of UEs in 5G networks can be summarized as follows:

- **Smartphones and Tablets:** the most common consumer devices, providing eMBB services, low latency, and support for advanced multimedia applications.
- **Fixed Wireless Access (FWA) Terminals:** devices designed to provide residential or enterprise broadband connectivity using 5G radio access instead of fixed-line infrastructure. These UEs can act as indoor or outdoor routers that distribute connectivity via Wi-Fi or Ethernet.
- **IoT and Machine-Type Communication Devices:** a large class of UEs that includes sensors, meters, and actuators that typically generate small data volumes but require low-cost, energy-efficient connectivity and long battery lifetimes.
- **Ultra-Reliable Low-Latency Communication Devices:** UEs designed for mission-critical applications, such as industrial robots, or connected healthcare devices.
- **Vehicular UEs (V2X):** on-board units in cars, buses, and other vehicles supporting Vehicle-to-Everything communication. These UEs rely on 5G NR V2X interfaces

to exchange information with infrastructure (V2I), other vehicles (V2V), pedestrians (V2P), and networks (V2N), enhancing traffic safety and autonomous driving functionalities.

- **Industrial and Enterprise Terminals:** specialized UEs deployed in private or campus networks, often designed with extended capabilities such as deterministic latency, high availability, and integration with local edge computing resources.

This broad diversity of UE types highlights the flexibility of 5G as a connectivity platform, supporting consumer broadband services, industrial applications, and massive-scale IoT deployments within a unified framework.

1.2 5G NR basics

This section provides an overview of key aspects of the 5G NR standard relevant to DL communication, focusing on the Reference Signals (RSs), PHY channels, control mechanisms, and Random Access (RA) procedure. For more details, readers are referred to the 3GPP specifications [2, 3, 4, 5, 6].

1.2.1 Physical Layer Overview

The PHY-layer in 5G NR is designed to support a wide range of frequencies, encompassing both sub-6 GHz (FR1) and mmWave (FR2, above 24 GHz) bands, and adapt to various deployment scenarios, thereby enhancing spectral efficiency. Both UL and DL transmissions employ Orthogonal Frequency Division Multiplexing (OFDM), which partitions the available bandwidth (BW) into multiple orthogonal subcarriers, providing robustness against interference and multipath fading.

Numerology and Frame Structure: 5G NR transmissions are organized into continuous sequences called NR frames, each composed by 10 subframes lasting 1 ms. NR enhances the PHY-layer structure by introducing a flexible parameter called numerology μ, which determines the Subcarrier Spacing (SCS) $\Delta_f = 2^\mu \cdot 15\,\text{kHz}$ and number of OFDM symbols per subframe $N_{symb}^{subframe} = 14 \cdot 2^\mu$, with $\mu \in 0, 1, \ldots, 6$. Unlike 4G LTE, in 5G NR, adjusting the numerology μ, the SCS can be customized from the set $\{15, 30, 60, 120, 240, 480, 960\}$ kHz, allowing the network to tailor the frame structure to specific application requirements, ranging from maximizing BW (up to 100 MHz in FR1 and up to 2000 MHz in FR2) to minimizing latency (with slot durations as short as 15.625 μs).

The typical NR frame comprises $N_{\text{slots}} = 10 \cdot 2^\mu$ *slots* of duration $T_{\text{slot}} = 2^{-\mu}\text{ms}$, each containing 14 OFDM symbols. At the most granular level, a Resource Element (RE), the smallest defined unit, is defined as a single OFDM subcarrier during one symbol interval. A group of 12 REs in the frequency domain and 14 REs in the time domain forms a RB, which is the smallest allocable unit of resources in 5G NR.

BWPs: 5G NR introduces the concept of BWPs, contiguous subsets of RBs within the carrier's BW. BWPs allow UEs to operate only over a portion of the total BW, reducing the Radio Frequency front-end and baseband processing requirements, lowering power consumption, and enabling more efficient and flexible use of spectrum resources. Each UE may have up to four DL and four UL BWPs per cell, although only one DL and one UL BWP can be active at any given time. BWPs enables flexible resource allocation by allowing the network to assign different BW sizes and locations to different UEs, accommodating diverse service requirements and device capabilities. Moreover, each BWP can have a specific PHY-layer configuration (i.e., starting RB, BWP size, SCS, cyclic prefix, and frequency location), which can be even dynamically reconfigured through higher-layer RRC signaling, enabling network adaptation to changing traffic conditions and service requirements. Figure 1.4 shows the two types of resource allocation and BWP establishment.

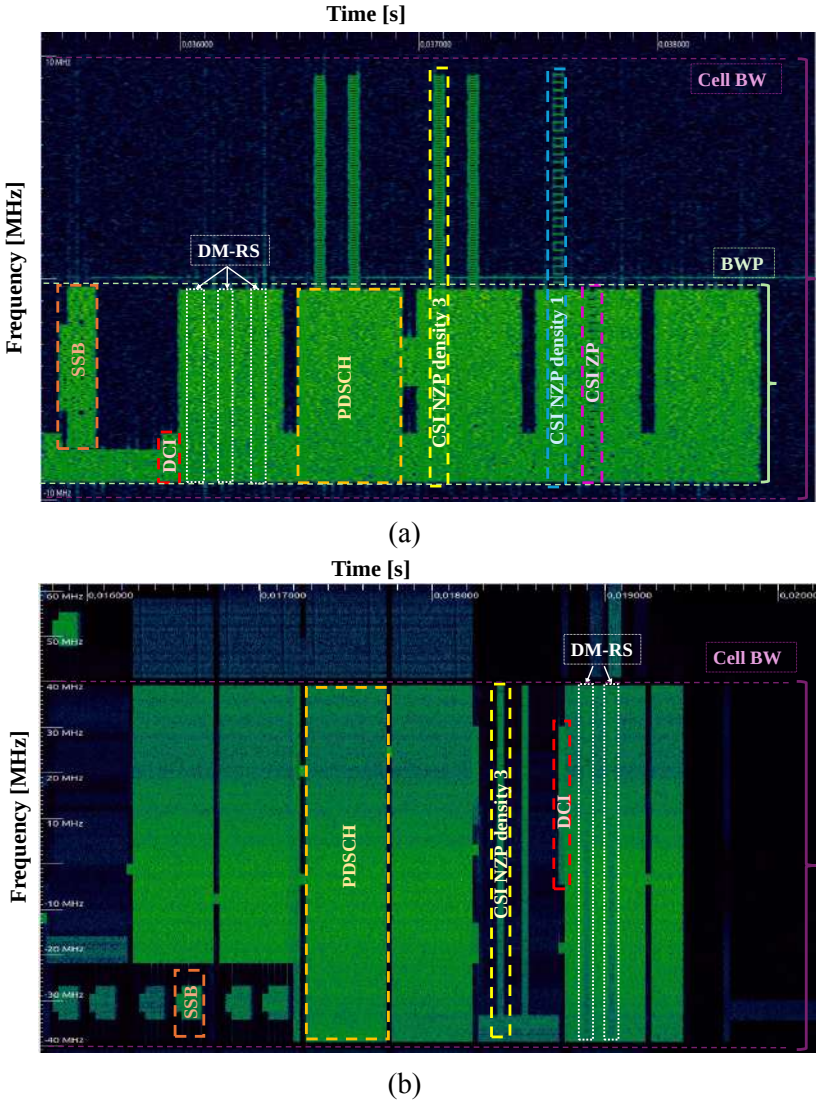


Figure 1.4: Comparison of DL resource allocation in a 5G NR cell: (a) emulated single-UE scenario with straighforward BWP establishment; (b) real-world multi-UE scenario where individual BWPs are more challenging to distinguish.

1.2.2 PHY Channels and Control Mechanism

This subsection provides a brief review of the physical channels, which are crucial for communication in the 5G NR. It also serves to introduce the concept of Downlink Control Information (DCI), carried by the Physical Downlink Control Channel (PDCCH) for scheduling, and reference signals used for channel estimation.

Synchronization Signals (Primary Synchronization Signal (PSS) and Secondary Synchronization Signal (SSS)) enable the UE to achieve both time and frequency synchronization with gNB during the initial cell search phase. These signals are transmitted in a specific time-frequency position within the Synchronization Signal Block (SSB), which also includes the Physical Broadcast Channel (PBCH); this channel contains the Master Information Block (MIB), which provides key system information necessary for initial access, such as the System Frame Number (SFN), SSB SCS and the configuration of the initial downlink BWP.

Physical Random Access Channel (PRACH) is used for the RA procedure, during which the gNB provides UE with Radio Network Temporary Identifier (RNTI), timing advance value, and a scheduling grant for uplink transmission. This procedure starts when a mobile transmits one preamble on the PRACH, that has its own SCS, which can be different from other uplink channels. Since this procedure is unencrypted, it is exposed to eavesdropping and spoofing attacks.

Physical Downlink Control Channel (PDCCH) carries DCI messages that include scheduling grants and control instructions for both downlink and uplink transmissions. This channel is transmitted either in common regions (shared by all UEs) or in UE-specific regions, called *Search Spaces*, which are continuously monitored by each UE to locate its own DCI messages. Each PDCCH transmission is mapped onto a set of *Control Channel Elements* (CCEs) within a specific *Control Resource Set* (CORESET). A CORESET defines a group of physical resources, organized in frequency and time, that are reserved for downlink control signaling. In practice, each CORESET is a region in the time–frequency grid composed of several *Resource Element Groups* (REGs), each REG spanning one RB in frequency (12 subcarriers) and one OFDM symbol in time. Four consecutive REGs form a single CCE, which represents the minimum allocable unit of PDCCH resources. Hence, the number of CCEs determines how much physical space is available to transmit a DCI message.

5G NR specifies multiple CORESETs that can coexist in a cell, each identified by an index. Among them, *CORESET 0* is the default control region used for initial access and broadcast signaling, such as Random Access Response (RAR) or paging, before dedicated search spaces are configured for a specific UE. Other CORESETs (e.g., CORESET 1 or CORESET 2) can then be configured dynamically via higher-layer signaling to support UE-specific control communication.

The number of CCEs aggregated to transmit one DCI defines the Aggregation Level (AL), which governs the trade-off between transmission robustness and spectral efficiency. Formally, AL $\in \{1, 2, 4, 8, 16\}$, meaning that 1, 2, 4, 8, or 16 consecutive CCEs can be assigned

to a single DCI. Higher ALs increase robustness against interference and decoding errors but consume more control resources; lower aggregation levels, instead, are used when the UE experiences favorable channel conditions and can reliably decode with fewer resources.

Downlink Control Information (DCI) messages carry essential information required by the UE to decode downlink data or to prepare for uplink transmission. 5G NR defines two main classes of DCI formats: *Fallback* formats, e.g. DL 1_0 and UL 0_0, used as default scheduling options with fixed fields and reduced size; and *Non-Fallback* formats, e.g. DL 1_1, UL 0_1, 2_0, 2_1, 2_2, and 2_3, which provide greater flexibility and support advanced NR features such as slot format indication, MIMO layer configuration, and dynamic TDD operation.

Physical Uplink Control Channel (PUCCH) is used by UE to ask for resources on Physical Uplink Shared Channel (PUSCH), if it has not. This triggers the transmission of an uplink grant on PDCCH. The message transmitted in the PUCCH is called Uplink Control Information (UCI), which contains or a scheduling requests, but it could be an uplink ACK, or channel quality reports.

Physical Downlink Shared Channel (PDSCH)/Physical Uplink Shared Channel (PUSCH) carry user-plane data, in downlink and uplink direction, respectively, in the resources specified in the PDCCH.

5G NR Reference Signals (RSs) ensure efficient communication between the UE and the network. The main RSs are presented below:

Demodulation Reference Signals (DM-RSs)) are used for channel estimation and demodulation of data on both the PDCCH and PDSCH. DM-RSs are specific to each UE, and their configuration is signaled through encrypted higher-layer RRC messages. Without access to the proper DM-RS configuration, the UE cannot accurately estimate the channel, resulting in demodulation errors on the PDSCH.

Channel State Information Reference Signals (CSI-RSs) are used by the UE in evaluating channel quality for purposes such as link adaptation, beamforming, and mobility management. There are two types of Channel State Information Reference Signals (CSI-RSs): NZP-CSI-RS, transmitted with power for active channel measurements, and ZP-CSI-RS, which allocate zero-power REs to enable interference measurements. The configuration of CSI-RSs, communicated via encrypted higher-layer RRC signaling, includes details such as resource mapping, density, transmission periodicity, and scrambling identities. Proper interpretation of CSI-RS is critical for correctly demapping (i.e., performing rate matching and interleaving) and decoding the PDSCH.

The integrity and confidentiality of both control and data channels are further reinforced through scrambling and modulation.

Scrambling and Modulation: Signals on both control and data channels are scrambled using sequences derived from parameters such as the cell ID and the UE's RNTI, which helps to reduce interference and improves security through signal randomization. Different RNTIs (e.g., C-RNTI, SI-RNTI) are applied depending on the type of data being transmitted. Scram-

bling sequences are initialized based on the UE's specific RNTI, the DM-RS scrambling ID, and the slot number. This ensures that only the intended UE can successfully descramble and decode the transmission. Without the correct RNTI and initialization parameters, the PDCCH or PDSCH cannot be properly descrambled or decoded.

1.2.3 5G NR Random Access Procedure

The Random Access (RA) procedure in 5G NR allows a UE to establish an initial connection with the network. It is essential for synchronization, resource request, and mobility procedures such as handover and beam failure recovery. The RA procedure can be either *Contention-Based* (CBRA) or *Contention-Free* (CFRA).
The RA process consists of the following signalling messages:

Msg1 – Preamble Transmission: The UE selects a preamble and a corresponding Random Access-Radio Network Temporary Identifier (RA-RNTI) and transmits them over the PRACH. In CBRA, the preamble is chosen randomly from a common set, leading to possible contention. In CFRA, the preamble is pre-assigned by the gNB, eliminating contention. In certain scenarios, such as handovers, the network must ensure contention-free access to maintain seamless connectivity.

Msg2 – Random Access Response (RAR): Upon receiving the preamble, the gNB transmits a RAR on the PDCCH. The UE attempts to detect a DCI format 1_0 with CRC scrambled by a corresponding RA-RNTI during a window controlled by higher layers. The RAR contains a Timing Advance (TA) value for synchronization, UL resource allocation for Msg3 and a Temporary C-RNTI. In CFRA the C-RNTI is a unique identifier avoiding collisions and completing the procedure.

Msg3 – RRC Connection Request: For CBRA the UE transmits an RRC request message on the PUSCH, including its identity. In CBRA, multiple UEs may have used the same preamble, leading to potential collisions at this step.

Msg4 – Contention Resolution: The gNB transmits a contention resolution message. If the UE receives its correct identifier, it successfully completes the RA procedure. Otherwise, it assumes a collision and reattempts the process.
The RA procedure occurs entirely in *cleartext* since the UE has not yet established security keys with the network. Encryption and integrity protection are activated only after the successful completion of the RA procedure, typically during the *RRC Setup Complete* or *RRC Resume Complete* message exchange.

1.2.4 MIMO overview and vulnerabilities

MIMO systems represent a key solution to boost channel capacity by creating parallel channels between the transmitter and receiver antennas. Considering a flat-fading MIMO system with n_t transmit and n_r receive antennas, the channel can be represented by the complex matrix $\mathbf{H} \in \mathbb{C}^{n_r \times n_t}$, whose (i, j)-th element h_{ij} denotes the channel gain between the j-th

transmit antenna and the i-th receive antenna. For a transmitted symbol vector $\mathbf{x} \in \mathbb{C}^{n_t \times 1}$, the received signal is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n},$$

where $\mathbf{n} \in \mathbb{C}^{n_r \times 1}$ is an additive white Gaussian noise (AWGN) vector with zero mean and variance σ_n^2 per component.

To decouple the MIMO channel into independent parallel subchannels, we perform the Singular Value Decomposition (SVD) of $\mathbf{H}$ as

$$\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H,$$

where:

- $\mathbf{U} \in \mathbb{C}^{n_r \times n_r}$ and $\mathbf{V} \in \mathbb{C}^{n_t \times n_t}$ are unitary matrices (i.e., $\mathbf{U}^H\mathbf{U} = \mathbf{I}_{n_r}$ and $\mathbf{V}^H\mathbf{V} = \mathbf{I}_{n_t}$);
- $\mathbf{\Sigma} \in \mathbb{R}^{n_r \times n_t}$ is a diagonal matrix containing the singular values $\sigma_1, \sigma_2, \ldots, \sigma_{\min(n_t,n_r)}$ of $\mathbf{H}$, sorted in descending order.

If the transmitted vector is *precoded* as $\mathbf{x} = \mathbf{V}\mathbf{s}$, where $\mathbf{s}$ is the vector of modulation symbols, and the received signal is *post-processed* as $\mathbf{r} = \mathbf{U}^H\mathbf{y}$, then

$$\mathbf{r} = \mathbf{U}^H\mathbf{H}\mathbf{V}\mathbf{s} + \mathbf{U}^H\mathbf{n} = \mathbf{\Sigma}\mathbf{s} + \tilde{\mathbf{n}}.$$

Because $\mathbf{U}$ and $\mathbf{V}$ are unitary, the post-processed noise $\tilde{\mathbf{n}}$ has the same statistical properties as $\mathbf{n}$. The resulting input–output relationship shows that the original MIMO channel has been transformed into $\min(n_t, n_r)$ parallel and independent Single-input-single-output (SISO) subchannels, each with gain equal to the corresponding singular value σ_i.

Let $\rho = E_s/\sigma_n^2$ denote the average signal-to-noise ratio (SNR) per receive antenna, and let P_i be the power allocated to the i-th eigenmode. The total transmitted power is constrained by $\sum_i P_i = P_{\text{tot}}$. The instantaneous channel capacity is then given by

$$C = \sum_{i=1}^{r} \log_2\left(1 + \frac{P_i\,\sigma_i^2}{\sigma_n^2}\right),$$

where $r = \text{rank}(\mathbf{H})$.

To maximize capacity under this power constraint, power should be distributed across eigenmodes according to the *water-filling* principle: stronger subchannels (with larger σ_i) receive more power. In the *low-SNR* regime, all available power is typically allocated to the strongest eigenmode, while in the *high-SNR* regime, power tends to be distributed almost uniformly across all active subchannels. This adaptive allocation ensures optimal use of the available transmit power given the instantaneous channel conditions.

In 5G networks, downlink MIMO decisions at the gNB are driven by channel matrix information gathered at the UE and reported to gNB in compressed form. Practically, the UE: (i) estimates the channel matrix $\mathbf{H}[k]$ for each subcarrier k through broadcast reference

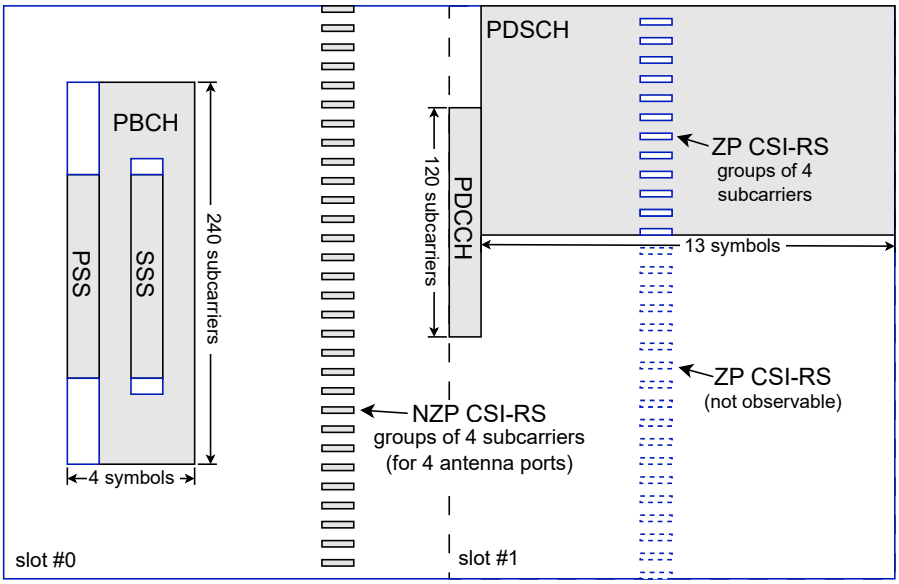


Figure 1.5: Stylized depiction of two slots in a frame: shaded areas represent power-bearing symbols/subcarriers. Note the synchronization block SS/PBCH, the PDCCH, the PDSCH, and the reference signals NZP-CSI-RS and ZP-CSI-RS. The ZP-CSI-RS REs outside of a PDSCH allocation are not observable.

signals; (ii) estimates the interference on a complementary set of reserved elements; (iii) evaluates several Precoding Matrix alternatives [5] and resulting SINR hypotheses to form a compressed feedback message delivered to the gNB.

More into details, in order to allow UEs performing channel estimation, the radio frames sent by gNB, organized in slots and RB, include CSI-RS. These signals are transmitted periodically; their mapping to RBs and subcarriers is predefined in the configuration provided by the gNB.

Figure 1.5 shows an example of CSI-RS configuration: note that NZP-CSI-RS form groups of REs amounting to the total number of transmit antenna ports available on the gNB (four REs in the figure, assuming four antenna ports on the gNB, and density equal to one group for every RB). ZP-CSI-RS, conversely, form groups of four REs that may be arranged in two ways, four adjacent subcarriers in one symbol as shown in the figure, or two subcarriers in two consecutive symbols. For reference, the figure also shows a SS/PBCH block (spanning 240 subcarriers and four symbols), a PDCCH message (spanning 120 subcarriers for AL 1 and one symbol duration), and a full slot PDSCH allocation (spanning 6 RBs and the remainder of the slot, i.e. 13 symbols).

Conversely, CSI reporting by UE can be periodic, semi-persistent or aperiodic. It is given by a triplet of parameters (Rank Indicator (RI), Channel Quality Indicator (CQI), Precoding Matrix Indicator (PMI)) computed at the UE. The RI value is the number of layers (from 1 to 8 maximum layers) chosen for MIMO transmissions. Rather than sending $\mathbf{H}[k]$, the UE finds this number as a function of the eigenvalues of the channel matrix; the standard does not specify the exact selection criteria of RI and leaves them to the vendor design. Given the number of layers, the CQI is an index (from 0 to 15) representing a selected modulation

format, as a function of the estimated SINR values. A higher CQI value indicates that the channel can support a more complex Modulation and Coding Scheme (MCS), thus allowing for a higher data transmission rate with a target Block Error Rate (BLER), typically 10%. Finally, the PMI indicates which precoding matrix to use at the gNB for maximizing the SINR at the UE.

1.2.5 Summary and Outlook

This chapter has outlined the essential components of the 5G NR architecture and PHY-layer, detailing the structure of radio frames and channels, the initial access procedure through the RA mechanism, and the fundamental principles of MIMO transmission and channel modeling. Together, these elements describe how UEs establish, maintain, and utilize the air interface for reliable communication. This foundational understanding of the PHY-layer and its key procedures sets the stage for the next chapter, which investigates how such signals can be passively captured and decoded, enabling practical analysis of 5G control and data channels.

Chapter 2

Breaking the Air Interface: Blind Recovery of Control and User Plane Metadata

Cellular networks, since the first generations and now with 5G, prioritize security and privacy to protect user information via mechanisms like consistent over-the-air encryption and temporary identifiers (e.g., Temporary Mobile Subscriber Identity (TMSI), Globally Unique Temporary Identifier (GUTI) and RNTIs) to prevent identity and location leakage. However, vulnerabilities persist: 4G LTE faces PHY-layer attacks such as jamming, spoofing [7, 8, 9, 10], fake base stations [11, 12, 13, 14, 15], man-in-the-middle [16], and overshadowing [17, 18]. Despite 3GPP's 5G enhancements, these risks remain only partially mitigated, leaving 5G NR vulnerable to similar attack vectors [19, 20, 21, 22, 23, 24].

Sniffing 5G NR transmissions presents a significantly greater challenge compared to LTE, primarily due to the concealment of RNTIs and relevant scrambling parameters. Existing State-of-the-art (SOTA) techniques for intercepting and reconstructing downlink control and data transmissions require exhaustive search methods entailing up to a billion tests per user and rely on scrambling identifier assumptions that may not consistently hold in real deployments. This chapter describes an approach that exploits the linearity and non-cryptographic nature of the Gold sequences used in 5G NR for scrambling, drastically reducing the sniffer parameter space to a handful of test alternatives. [25, 26, 27].

Why 5G sniffing is harder than 4G. It is widely agreed [28, 29, 24] that passive sniffing of control and data traffic is far more challenging in 5G NR than in LTE. In LTE, PDCCH and PBCH control information follows fixed mappings and uses parameters sent unencrypted [30], enabling straightforward decoding. In 5G NR, critical configuration is conveyed through either scrambled (pre-security) or encrypted (post-security) RRC messages, making passive interception significantly harder. More specifically, the increased difficulty in 5G NR sniffing primarily arises from the lack of knowledge by a passive third-party player of two critical temporary parameters assigned to users: (i) a dynamically assigned 16-bit RNTI which uniquely identifies each receiver throughout an access session, and (ii) a further

16-bit identifier used for ”scrambling” (ScramblingID).

Regarding (i), the RNTI can, in principle, be extracted by intercepting the first two messages of the Random Access procedure, where the DL response is scrambled with UL-derived information. Unlike in 4G LTE, failing to reconstruct the RNTI at this stage leaves no later opportunity, as post-security, it is encrypted. In fact, most of existing SOTA 5G sniffers, i.e., Sni5Gect [31], and NR-Scope [32] address RNTI decoding through an opportunistic approach relying on capturing the clear-text RRC registration procedure; 5GSniffer [29], instead, addresses RNTI decoding through a computationally intensive approach. The sniffing method presented in this chapter eliminates brute-force searches, blindly deriving this (and also the other in (ii)) key parameter on-the-fly, drastically improving efficiency. Indeed, a similar challenge arises with parameter (ii), the ScramblingID, a crucial 16-bit value that initializes a pseudo-random sequence applied to encoded data before modulation, ensuring interference reduction and distinct transmissions for each UE. While it defaults to the Physical Cell ID (PCI), most real-world deployments assign it dynamically. As a result, this parameter, rather than just guessed, must in practice also be blindly decoded, and the method proposed in [29], albeit clever, still takes a quite significant amount of time. Finally, each RNTI/ScramblingID pair demands testing of several combinations of additional parameters, i.e., five ALs, three possible DCI durations, and up to 100 possible sizes, adding complexity to passive interception.

Harnessing Gold Sequences. Compared to the pioneering work on blind 5G NR sniffing [29], which makes the blind recovery of key parameters feasible through a set of careful optimizations and essentially brute-force search over a large parameter space, our approach follows a fundamentally different and more algebraic strategy. The proposed *Golden Sniffer* leverages the fact that the scrambling sequences used in 5G NR are Gold sequences: deterministic, pseudo-random sequences with a well-defined linear structure over $\mathrm{GF}(2)$. Because both the Cyclic Redundancy Check (CRC) and the scrambled payload bits are combined by XOR operations, their relationship with the unknown parameters (the DCI payload bits, the RNTI, and the ScramblingID/seed) is linear over $\mathrm{GF}(2)$. By passively capturing transmitted DM-RS sequences and exploiting the known construction and positions of these reference signals, the sniffer obtains observed bit sequences that can be expressed as a system of linear equations in the unknown initialization bits of the Gold generator and the DCI-related fields. Solving this linear system (or a reduced search over a much smaller subspace when the system is underdetermined) yields the initialization parameters directly, removing the need for exhaustive guessing or large-scale brute force required by the SOTA approach in [29].

This algebraic recovery requires a sufficient number of correctly decoded reference symbols and a reasonable SNR, as well as knowledge of the DM-RS mapping used by the cell; under these practical conditions, the method dramatically reduces computational complexity and increases reliability of blind parameter recovery compared to purely brute-force strategies. Moreover, while [29] addresses only the PDCCH, this sniffing method extends to single-layer PDSCH, as it does not rely on higher-layer configurations. This enables *full*

DL decoding making our approach more versatile.

2.1 Related Work

Commercial and academic efforts have developed various monitoring tools over the past decade. Table 2.1 summarizes key features of academic tools developed for LTE and NR.

Commercial solutions. Products like Keysight's WaveJudge [33], Qualcomm's QXDM [34], Actix Analyzer [35], TEMS™ Investigation [36], and thinkRF's Autonomous Spectrum Intelligence Platform [37] offer advanced capabilities for drive testing, performance monitoring, and troubleshooting in LTE and emerging networks. However, their high cost and closed-source nature limit their accessibility for research.

LTE sniffers. LTE sniffing has been addressed by several works. Kumar et al. [38] introduced LTEye, one of the first systems to demonstrate mobile network sniffing using COTS SDRs without operator support. However, its candidate DCIs re-encoding verification is highly sensitive to interference, noise, and synchronization errors, thus requiring near-ideal radio conditions. RMon [39] combines control channel decoding with PHY-layer monitoring to infer network load and cell utilization, but its reliance on ideal conditions limits practicality. To overcome re-encoding limitations, C3ACE [40] uses short-term histograms of decoded RNTIs to filter out false positives, improving robustness at the cost of delayed detection for infrequently scheduled UEs. OWL [41] tracks initial RNTIs from RAR messages and applies coherence checks to exclude invalid DCIs; while it fully decodes the PDCCH and partially the PDSCH, its reliance on re-encoding reduces reliability under poor conditions. FALCON [42] reduces false detections using a recursive, depth-first shortcut-decoding method. LTEProbe [43] enables passive user tracking by correlating UL and DL transmissions and leveraging Timing Advance for precise location estimation. LTESniffer [44] is an open-source tool that fully decodes both DL and UL traffic, infers user-specific configurations and compensates for propagation delays.

5G NR sniffers. With 5G NR, sniffers must rely on complex blind decoding techniques, often resulting in low DCI detection rates, or must capture the very first clear-text registration exchanges to extract configuration hints. 5GSniffer [29], the first open-source tool targeting 5G NR control channels, demonstrates the feasibility of blind extraction of PDCCH information. However, it fails to achieve fully reliable DCI decoding [45, 46], is limited to single-symbol DCIs, and lacks support for flexible BWP allocation. Similarly, Richards et al. [28] have shown promising results in extracting user RNTIs from the PDCCH. Both studies make the simplifying assumption that the Scrambling ID equals the PCI, thereby setting the RNTI used in the scrambling process (i.e. RNTI_{SCR} in Sec. 2.3.3) to zero instead of the actual RNTI used to mask the CRC, as found in commercial deployments (Sec. 2.4.4). This assumption reduces the brute-force sniffer PDCCH DM-RS cardinality, making the proposed approaches feasible. Recently, NRScope [32] demonstrated practical PDCCH decoding, and Sni5Gect [31] extends also to PDSCH data; however, both tools rely on capturing the clear-text reg-

istration procedure to address DCI decoding challenges, a condition not always available in typical network scenarios. Taken together, prior works illuminate only part of the problem: 5GSniffer performs blind decoding but stops at the PDCCH, whereas Sni5Gect reaches the PDSCH only when it can eavesdrop an unencrypted registration exchange. The presented work bridges this gap by achieving truly blind single-layer PDSCH decoding; extending the technique to multi-layer transmissions remains future work.

Table 2.1: Comparative Summary of Related Work on Mobile Network Sniffers.

Work	Mobile Network Technology	BWP Support[1]	Physical channels[2]		Complete DCI Decoding	No Reliance on Unencrypted Reg.
			PDCCH	PDSCH		
Kumar et al. [38]	4G LTE	N/A	●	○	✓[a]	✓
Xie et al. [39]	4G LTE	N/A	●	○	✓	✓
Falkenberg et al. [40]	4G LTE	N/A	●	○	✓[b]	✓
Bui et al. [41]	4G LTE	N/A	●	◖ (RAR)	✓[b]	✓
Falkenberg et al. [42]	4G LTE	N/A	●	○	✓	✓
Kotuliak et al. [43]	4G LTE	N/A	●	●	✓	✓
Hoang et al. [44]	4G LTE	N/A	●	●	✓	✓
Ludant et al. [29]	5G NR	✗	●	○	✗[c]	✓
Richards et al. [28]	5G NR	✗	●	○	✗[c]	✓
Wan et al. [32]	5G NR	✓	●	○	✓	✗
Leo et al. [31]	5G NR	✓	●	●	✓	✗
Our Work	**5G NR**	✓	●	●	✓	✓

[1] Specifies whether it supports network configurations with UE-dedicated BWP, allowing adaptation to multiple BW configurations within a single carrier.
[2] Indicates the physical layer channels investigated in the study: ●: Decoded, ◖: Partially decoded, ○: Not decoded.
[a,b] Requires good signal quality and/or long monitoring time.
[c] Assume PCI as Scrambling ID (default configuration).

2.2 5G NR Sniffer Challenges

This section outlines key challenges in blind monitoring of 5G NR DL control and data traffic, focusing on decoding PDCCH DCI messages, which are critical for PDSCH decoding.

BWP Configuration Inference: BWP configurations, including starting RB, size, SCS, cyclic prefix type, and frequency location, are securely communicated to the UE via encrypted RRC signaling. Consequently, sniffers must blindly infer these configurations to accurately interpret resource assignments in DCIs, increasing both complexity and computational requirements.

Accurately identifying and reconstructing the resources used by each UE is particularly challenging for sniffers in complex multi-UE scenarios. This complexity arises because DCI messages on the PDCCH specify allocations relative to the active BWP, encoding information such as data channel resource allocation (PDSCH and PUSCH), MCS, and essential Hybrid ARQ (HARQ) parameters. Without knowledge of the active BWP, the sniffer cannot reliably map these assignments to the correct resource locations, thus hindering effective control and data channel decoding.

For instance, consider an allocation with on the order of 100 RBs and an unknown active BWP. The BWP can occupy any contiguous block of RBs, with varying starting positions and lengths. The total number of possible BWP configurations can therefore be approximated as $C_B \approx 100 \times 100 \simeq 10^4$ corresponding to 13 bits of information ($log_2 C_B \approx 13$)).

PDCCH Search Space Localization: As discussed in Subsection 1.2.2, the PDCCH transmits scheduling information via DCI messages, using Control Resource Sets (CORESETs) and AL to map resources. The UE monitors PDCCH search spaces within the CORESET to locate its DCIs. A passive sniffer faces substantial challenges because key PDCCH search-space parameters (e.g., CORESET configurations, ALs, monitoring periodicities) are conveyed via higher-layer RRC messages and thus are not directly observable. As a result, the sniffer must effectively search over a large space of possible parameter combinations to detect DCI messages, which is both computationally expensive and prone to missed detections.

To make this quantitative, consider the cost of inferring the per-UE frequency-resource bitmap. In practice, assuming an RBG granularity of 6 RBs, a carrier with N_{RB} RBs (units of $N_{\mathrm{sc}}^{\mathrm{RB}} = 12$ subcarriers) yields approximately $\lfloor N_{\mathrm{RB}}/6 \rfloor$ independent bitmap positions. Hence, the number of possible bitmaps per UE is on the order of $C_C = 2^{\lfloor N_{\mathrm{RB}}/6 \rfloor}$, i.e. the sniffer must potentially test all $2^{\lfloor N_{\mathrm{RB}}/6 \rfloor}$ combinations to determine which RBGs are active for that UE.

In addition, the sniffer must also consider other unknown control parameters. If there are C_A possible aggregation levels and C_D possible DCI durations to try, the total number of parameter combinations per UE multiplies accordingly. For example, assuming: $\lfloor N_{\mathrm{RB}}/6 \rfloor \approx \lfloor 100/6 \rfloor = 16, \quad C_A = 5$ (AL), $\quad C_D = 3$ (DCI durations), the total number of bits to scan per UE is approximately $\log_2 C_C + \log_2 C_A + \log_2 C_D = \lfloor N_{\mathrm{RB}}/6 \rfloor + \log_2(5) + \log_2(3) \approx 16 + 2.32 + 1.59 \approx 19.9 \approx 20$ bits.

Thus, under these realistic assumptions, a sniffer must search a space of roughly 2^{20} candidate parameter combinations per UE.

DCI Format Ambiguity Resolution: DCI messages inform UEs about resource allocations for both DL PDSCH and UL PUSCH, as well as MCS, HARQ feedback, and power control commands. DCIs vary in size and format, depending on factors such as DL/UL assignments, BWP size, number of configured antennas, and UE capabilities.

As already explained in Subsection 1.2.2, DCIs are categorized into *Fallback* formats, – simpler, fixed-size formats such as DL 1_0 and UL 0_0 for basic operations – and more flexible *Non-Fallback* formats, including DL 1_1, UL 0_1, and the 2_x series, which support advanced NR features such as slot format indication, pre-emption notifications, and UL power control.

Registered UEs rely on configuration information conveyed through RRC signaling to determine the expected DCI length and format, enabling correct decoding. Passive sniffers face considerable difficulty because key PDCCH search-space parameters (i.e. CORESET configurations, ALs, and monitoring periodicities) are signalled via higher-layer RRC messages and remain inaccessible to the sniffer. Thus, the sniffer must effectively attempt decoding over all plausible combinations of parameters, greatly increasing computational burden and the chance of missing relevant DCI messages.

As a concrete illustration, let us examine the uncertainty in one of these parameters: the

DCI message length. The 3GPP standard indicates that the actual bit-length of a DCI depends on multiple configuration fields (e.g., frequency-domain resource assignment size, number of RBs, RBG granularities) and varies substantially across formats and setups.

In practice, a sniffer that knows only the cell bandwidth (for example 100 RBs) and none of the per-UE or per-format configuration fields must consider a large number of candidate lengths. For example, if the bit-length can take on around one hundred plausible values under typical deployments, then the number of bits of uncertainty becomes $\log_2 C_L = \log_2 100 \approx 6.64 \approx 7$ bits.

PDCCH/PDSCH DM-RS & Scrambling: Both PDCCH and PDSCH channels utilize DM-RSs for precise channel estimation and reliable demodulation of control and data information. These user-specific signals' configurations include the number of symbols, scrambling initialization, and mapping type (*A* or *B*), which affect the DM-RSs placement within the slot. DM-RS sequences are generated through a pseudo-random scrambling process initialized by encrypted configuration parameters provided via RRC. Specifically, PDCCH DM-RS sequences are generated using a Scrambling ID (e.g., *pdcch-DMRS-ScramblingID*) combined with the slot number, whereas the PDSCH DM-RSs sequences rely on user-specific scrambling IDs (*scramblingID0*, *scramblingID1*) and the UE's RNTI.

Since these parameters are not accessible to the sniffer, accurately reconstructing DM-RSs becomes extremely challenging, leading to degraded channel estimation and failures in decoding both PDCCH and PDSCH channels. Due to the fact that the PDSCH DM-RS symbol-position ambiguity (C_M) is a small bounded set, the 16-bit Scrambling ID ($C_S = 2^{16}$) brute-forcing dominates.

RNTI and Scrambling ID Extraction: Control and data channels in 5G NR are scrambled using sequences derived from higher-layer signaled parameters, i.e., the DM-RS Scrambling ID, the slot number, and the UE's unique RNTI, effectively randomizing the transmitted signals to mitigate interference and enhance security. Different RNTIs (e.g., C-RNTI, SI-RNTI) may be used depending on the data type. The RNTI specifically scrambles the CRC bits appended to the DCI payload, making it essential for decoding control messages. Without knowing the correct RNTI, a sniffer cannot validate decoded DCI, resulting in errors or missed detections.

Since both the RNTI and associated Scrambling IDs are communicated through encrypted higher-layer RRC signaling, sniffers must indirectly infer these values. Even if brute-force search to recover these parameters is theoretically possible [28, 29], it incurs substantial computational complexity and may be impractical in multi-UE real-time environments. Jointly guessing the 16-bit CRC-masking RNTI ($C_R = 2^{16}$) and the 16-bit payload-scrambling seed ($C_S = 2^{16}$) again yields 2^{32} pairs to be tested.

CSI-RS Configuration Inference: CSI-RSs enable UEs to measure channel quality for link adaptation, beam management, and mobility. Optimizing resource utilization, NZP-CSI-RS and ZP-CSI-RS are typically common to all UEs, allowing estimation of channel rank for MIMO and interference measurement. The CSI-RSs configurations – such as

resource mapping, density patterns, transmission periodicity, scrambling IDs, and antenna ports – are conveyed via encrypted RRC messages. A sniffer, lacking access to these configurations, may misinterpret reserved REs as data-carrying, leading to errors in demapping, rate matching, and PDSCH decoding, especially for ZP-CSI-RS where intentionally blank elements must be distinguished from unscheduled data.

To quantify the computational challenge, consider the number of plausible configurations a sniffer must test:

- **NZP-CSI-RS (C_X)**: Several parameters jointly define the configuration space, including density/pattern (3 bits), transmission periodicity (3 bits), scrambling ID (4 bits), antenna port/layer mapping (4 bits), and an additional resource offset or configuration index (2 bits). Combining these factors yields approximately $C_X \simeq 2^{16}$ possible NZP-CSI-RS configurations.
- **ZP-CSI-RS (C_Z)**: Fewer parameters are involved, such as density/pattern (2 bits), periodicity (2 bits), scrambling ID (4 bits), antenna port/layer mapping (2 bits), and a configuration index (2 bits), resulting in $C_Z \simeq 2^{12}$ possible ZP-CSI-RS configurations.

Assuming independence between NZP and ZP configurations, the total number of candidate CSI-RS setups that must be tested is roughly $2^{16} \cdot 2^{12} = 2^{28}$ alternatives.

2.3 Golden Sniffer: Approach

To overcome passive 5G NR monitoring challenges, this approach reduces a naïve exhaustive sniffer parameter space of 2^{58} configurations, from possible CORESET configurations, 16-bit RNTI, 16-bit ScramblingID, 5 ALs, 3 DCI durations, and up to 100 DCI sizes, to at most 50 candidates (active DCI sizes). This reduction is achieved after the initial optimized determination of key parameters, which are then cached and maintained over time for reuse in subsequent decoding, ensuring sustained efficiency without repeated exhaustive searches. Below, each solution is explained in detail.

2.3.1 Iterative Discovery Techniques

The presented challenges involve unknown parameters with finite sets of possible values, conveyed via encrypted RRC messages. Power distribution analysis is fundamental to considerably reducing the sniffer parameter space by identifying energy patterns in RBs, to identify the position of DCI candidates, thus avoiding exhaustive brute-force searches and improving computational efficiency. We test only compatible ALs and DCI durations, proceeding from largest to smallest allocation, reducing the search burden. Once a PDCCH message and associated PDSCH are decoded successfully, the BWP configuration for that RNTI is cached.

In order to identify the time and frequency resources allocated to the data and signaling transmissions, the sniffer first synchronizes with the target 5G cell by detecting the SSB. This standard procedure involves, as explained in Subsection 1.2.2, time-domain correlation for PSS detection, followed by frequency-domain correlation for SSS detection, and then the decoding of the PBCH. A successful PBCH decoding provides essential synchronization information, including the synchronization frame number and the slot index. The noise floor P_{noise} can also be estimated at this point, by averaging the power of subcarriers around the PSS region, which are left intentionally empty in NR. Once synchronization is completed, the receiver can perform FFT processing, obtaining the time-frequency representation of the frame $Y \in \mathbb{C}^{N_{\text{sc}}^{\text{frame}} \times N_{\text{symb}}^{\text{frame}}}$. The number of OFDM symbols in a frame is given by $N_{\text{symb}}^{\text{frame}} = 14 \cdot 10 \cdot 2^{\mu}$, while the number of subcarriers $N_{\text{sc}}^{\text{frame}} = N_{\text{RB}} \cdot N_{\text{sc}}^{\text{RB}}$ [47].

Now, the iterative detection process can start.

Iterative BWP Estimation: To infer the UE's active BWP, *Golden Sniffer* analyzes the information embedded in decoded DCI messages, particularly the Frequency Domain Allocation (FDA) field, and validates each inferred configuration through successful PDSCH decoding.

Each 5G NR carrier can host multiple BWPs with different sizes and frequency offsets, as defined in [47]. Since the FDA field in a DCI is expressed relative to the currently active BWP, the sniffer cannot know in advance which BWP configuration is in use. Therefore, it must test several BWP hypotheses to find the one that matches the transmitted data. The inference procedure proceeds as follows:

1. *FDA extraction*: The sniffer begins by extracting the FDA field from the decoded DCI. This field specifies which RBs within the active BWP are allocated to the PDSCH.

2. *Candidate BWP generation*: Based on carrier configuration parameters obtained from PBCH decoding (N_{RB} and μ), the sniffer enumerates all feasible BWP sizes and offsets. Each hypothesis corresponds to a possible frequency span, where the UE could be operating.

3. *RIV interpretation for each hypothesis*: When the FDA is encoded using a *Type 1* scheme, it contains a Resource Indication Value (RIV), a compact number that encodes both the starting RB and size. Because the meaning of an RIV depends on the BWP's total number of RBs, the sniffer must reinterpret this parameter under each BWP hypothesis. Each reinterpretation yields a different candidate pair $(\text{starting RB}, \text{size})$.

4. *Validation through PDSCH decoding*: For every BWP hypothesis, the sniffer attempts to decode the PDSCH using the corresponding RB mapping. If decoding succeeds and the CRC check passes, the hypothesis is confirmed as the UE's active BWP.

5. *Type 0 FDA special case*: If the DCI uses a *Type 0* FDA scheme, the allocation is represented as a direct bitmap of RBGs. In this case, the mapping to RBs is absolute

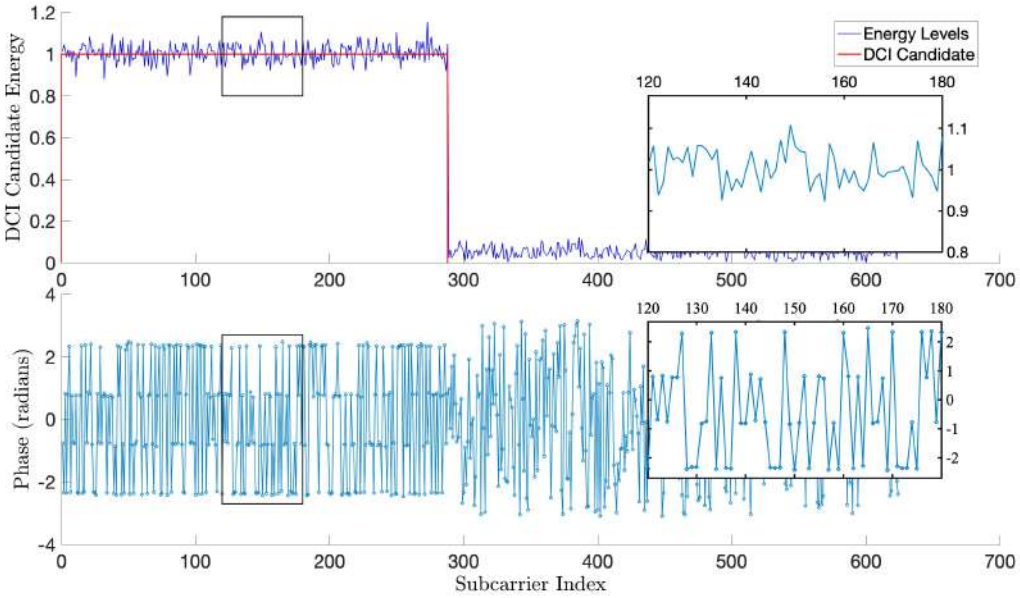


Figure 2.1: DCI candidates: (top) power levels with red-highlighted candidates (288 carriers); (bottom) IQ symbol phases revealing four distinct phases.

and does not depend on the BWP size, making the decoding process straightforward [3].

Once a valid BWP configuration is identified and confirmed through successful PDSCH decoding, its parameters (starting RB and size) are stored and reused for subsequent DCI messages belonging to the same UE. This caching avoids redundant hypothesis testing and allows the sniffer to track user activity efficiently across transmissions.

This solution determines $C_B \simeq 10^4$ (the correct BWP configuration) by trial and error based on PDSCH decoding. To reduce the number of trials, the FDA field from decoded DCIs is exploited, and BWP hypotheses that are not compatible with the DCI allocation are not considered. After convergence, the complexity is reduced to $O(1)$.

Exhaustive PDCCH Search Discovery: To solve the UE PDCCH search spaces challenge, we directly try to locate the DCI candidates in the first slot symbol, bypassing per-UE CORESET configurations enumeration. To reduce the complexity, we analyze the DCI candidate's power levels, illustrated in Figure 2.1, which shows how a candidate will correspond to higher power intensity. We assume non-interleaved PDCCH with contiguous frequency resources, enabling us to restrict the positions of candidate DCIs only to sharp power clusters, without exhaustive search. Interleaved mapping and non-contiguous frequency masks are left to future work. Figure 2.1 also reveals that Quadrature Phase Shift Keying (QPSK) DCI candidate phases assume only four values, hinting at the key intuition behind blind decoding.

This solution greedily tackles this challenge by looking at RB-power clusters and trying to detect valid DCIs starting from largest AL ($C_A = 5$) and ($C_D = 3$) durations. This results in a stochastic number of decoding attempts that depends on the size of identified clusters, pruning C_C, C_A and C_D to just a few candidates, achieving $O(1)$ complexity.

Adaptive DCI Format Inference: Inferring the DCI format and size without prior

knowledge is challenging due to the high variability in DCI lengths. Our approach addresses this uncertainty through iterative interpretation and caching of successfully decoded formats. While both *Fallback* (1_0 and 0_0) and *Non-Fallback* (1_1 and 0_1) DCIs can be detected, we currently perform full interpretation only for *Fallback* formats relevant to PDSCH decoding, leaving *Non-Fallback* support to future work.

Initially, the sniffer faces about $C_L \simeq 100$ possible DCI lengths ($\simeq 7$ bits of uncertainty), as each length depends on the unknown BWP and aggregation parameters. By iteratively testing and caching the valid formats associated with the active BWP, the sniffer restricts the search to only the currently used lengths. This adaptive filtering effectively removes 6 bits of uncertainty from C_L, resulting in a substantial reduction in computational effort during DCI interpretation.

2.3.2 DM-RS Reconstruction

Reconstructing PDCCH and PDSCH DM-RSs and scrambling sequences is addressed by exploiting the non-cryptographic nature of 5G NR Gold sequences. By capturing transmitted DM-RS sequences, and applying our proposed technique, we can recover the sequences' initialization parameters. While detailed for PDCCH DM-RS, the approach can be also applied to PDSCH , differing mainly in DM-RS symbol mapping and sequence parameters. For PDSCH, we perform a blind search across all OFDM symbols in the slot to identify the DM-RS positions and accurately interpret the slot format. However, assuming high-SNR regime, QPSK-modulated DM-RS symbols are usually easy to distinguish from the higher-order modulated PDSCH symbols. Thus, by applying our technique, we can reconstruct the DM-RS sequences for both channels, enabling accurate channel estimation and decoding without prior knowledge of higher-layer configurations.

Gold Sequence Generation and Analysis: The DM-RS sequences are generated using a pseudo-random length-31 Gold sequence $c(n)$ [2]. Our method involves capturing the transmitted DM-RS symbols and estimating the sequence $c(n)$, to obtain initialization parameters, such as scrambling IDs and slot numbers. The sequence $c(n)$ is defined as:

$$c(n) = x_1(n + N_C) \oplus x_2(n + N_C) \tag{2.1}$$

with $N_C = 1600$. The independent sequences $x_1(n)$ and $x_2(n)$ are defined by the following recursions:

$$\begin{aligned} x_1(n + 31) &= x_1(n + 3) \oplus x_1(n), \\ x_2(n + 31) &= x_2(n + 3) \oplus x_2(n + 2) \oplus x_2(n + 1) \oplus x_2(n) \end{aligned} \tag{2.2}$$

where $x_1(n)$ is initialized with $x_1(0) = 1$ and $x_1(n) = 0$ for $n = 1, \ldots, 30$, while $x_2(n)$ initialization is given by $c_{\text{init}} = \sum_{i=0}^{30} x_2(i) \cdot 2^i$, and encapsulates several higher-layer parameters of interest depending on sequence's application. When used for the PDCCH DM-RS, c_{init} in-

cludes the slot number, the symbol index, and the ScramblingID, which is securely delivered only to the intended user. Since $x_1(n)$ initialization is known, the sequence $x_1(n + N_C)$ can be precomputed. Our goal is to estimate c_{init} from a sequence of observations $\hat{c}(n)$. Given that $x_1(n)$ is known, the estimated sequence $\hat{c}(n)$ can be associated with an estimate of $x_2(n)$ as:

$$\hat{x_2}(n + N_C) = \hat{c}(n) \oplus x_1(n + N_C) \tag{2.3}$$

where $x_1(n+N_C)$ is precomputed. The recursion for $x_2(n)$ can then be reformulated in terms of future values:

$$x_2(n) = x_2(n + 1) \oplus x_2(n + 2) \oplus x_2(n + 3) \oplus x_2(n + 31) \tag{2.4}$$

allowing for the estimation of the initialization vector c_{init} from any set of 31 consecutive values of the sequence $c(n)$.

The sequence $c(n)$ is used to generate the received DM-RS sequences $r_l(m)$. Indeed, for the l-th OFDM symbol, $r_l(m)$ is defined by the QPSK mapping:

$$r_l(m) = \frac{1}{\sqrt{2}}(1 - 2 \cdot c(2m)) + j\frac{1}{\sqrt{2}}(1 - 2 \cdot c(2m + 1)) \tag{2.5}$$

Therefore, to obtain 31 consecutive values of $c(n)$, we need only 16 consecutive DM-RS symbols $r_l(m)$.

Mapping of DM-RS in PDCCH: The DM-RS sequence $r_l(m)$ is mapped to physical resources as:

$$a_{k,l}^{(p,\mu)} = \beta_{\text{DMRS}}^{\text{PDCCH}} \cdot r_l(3n + k') \tag{2.6}$$

where $a_{k,l}^{(p,\mu)}$ is the transmitted symbol at subcarrier k and OFDM symbol l, for antenna port p and numerology μ, $\beta_{\text{DMRS}}^{\text{PDCCH}}$ is the power scaling factor for the PDCCH DM-RS, $k = nN_{\text{sc}}^{\text{RB}} + 4k' + 1$ with $k' = 0, 1, 2$, and $n \geq 0$.

Since the number of subcarriers in a RB $N_{\text{sc}}^{\text{RB}}$ is equal to 12, we obtain $k = 4(3n+k')+1$, where $n' = 3n + k'$ can take every possible integer value. Consequently, Equation (2.6) becomes:

$$a_{4n'+1,l}^{(p,\mu)} = \beta_{\text{DMRS}}^{\text{PDCCH}} \cdot r_l(n'). \tag{2.7}$$

Hence, the DM-RS symbols are transmitted every fourth subcarrier, starting at index 1 within the PDCCH RBs.

Phase Analysis for Symbol Estimation: Our strategy exploits the limited alphabet of the QPSK mapping of $c(n)$ to infer the full PDCCH message and its embedded DM-RS sequence by analyzing the estimated phase values at the receiver. This is illustrated in Figure 2.1 (bottom), which displays the four distinct phase states of DCI candidate regions.

Let us define $\Delta\Phi_{k,l}$, as the $\pi/2$-spaced phase increment estimate with minimum angular

distance from the observed phase increment:

$$\Delta\Phi_{k,l} = \arg \min_{\Phi \in \{0, \pm\pi/2, \pi\}} \left| e^{j(\angle y_{k+1,l} - \angle y_{k,l})} - e^{j\Phi} \right| \tag{2.8}$$

which, under Line-of-sight (LOS)-dominant path and high-SNR regime assumptions, equals $\Delta\Phi_{k,l} = \angle a_{k+1,l} - \angle a_{k,l}$, enabling candidate symbol sequences reconstruction:

$$\hat{a}_{k_0+\Delta k,l} = \hat{a}_{k_0,l} e^{j \sum_{k=k_0}^{k_0+\Delta k-1} \Delta\Phi_{k,l}}, \tag{2.9}$$

where the four possibilities stem from the initial choice of $\hat{a}_{k_0,l}$. Each reconstructed sequence embeds a candidate DM-RS, obtained from (2.7) as $\hat{r}_l(n') = \hat{a}_{4n'+1,l}$, which, in turn, yields the following tentative Gold sequence:

$$\hat{c}(n) = \begin{cases} \mathrm{Re}\{\hat{r}_l(n)\} < 0, & n \text{ even} \\ \mathrm{Im}\{\hat{r}_l(n)\} < 0, & n \text{ odd} \end{cases}$$

Since the sequence $x_1(n)$ is known, the candidate sequence $\hat{c}(n)$ corresponds to the one for which the estimated sequences $\hat{x_2}(n)$, as in (2.3), satisfies the recursive formula in (2.2). Given a suitable candidate, we can compute N_C more terms in order to recover the initial state c_{init}.

The QPSK mapping of $c(n)$ for 5G NR reference signals is defined in [2].

Gold Sequence-Based DM-RS Recovery & Scrambling Estimation: After estimating the initialization value c_{init}, we solve for the Gold sequence generation parameters, such as the ScramblingID N_{ID}, the slot number $n^{\mu}_{\text{s,f}}$, and the OFDM symbol index l. For the PDCCH DM-RS, c_{init} is defined as [2]:

$$c_{\text{init}} = [2^{17} \cdot x \cdot (2N_{\text{ID}} + 1) + 2N_{\text{ID}}] \bmod 2^{31} \tag{2.10}$$

where $x = N^{\text{slot}}_{\text{symb}} \cdot n^{\mu}_{\text{s,f}} + l + 1$. Here, $N^{\text{slot}}_{\text{symb}}$ is the number of OFDM symbols per slot (typically 14), $n^{\mu}_{\text{s,f}}$ is the slot number within a frame (with $0 \leq n^{\mu}_{\text{s,f}} < 10 \cdot 2^{\mu}$), and l is the OFDM symbol index within the slot ($0 \leq l < N^{\text{slot}}_{\text{symb}}$). The parameter N_{ID} is either provided as the higher-layer parameter *pdcch-DMRS-ScramblingID* or, if not available, the PCI $N^{\text{cell}}_{\text{ID}}$. Given c_{init}, we extract 16-bit N_{ID} as $(c_{\text{init}}/2) \bmod 2^{16}$, and we obtain x from (2.10) checking that $x \leq 140 \cdot 2^{\mu}$ matches the slot and symbol index.

This approach jointly resolves both the ScramblingID ambiguity (C_S) and the DM-RS position uncertainty (C_M), yielding an $O(1)$ solution instead of an exponential 2^{16} brute-force. The validated (N_{ID}, x) pair is cached to eliminate residual slot/symbol ambiguity in subsequent frames.

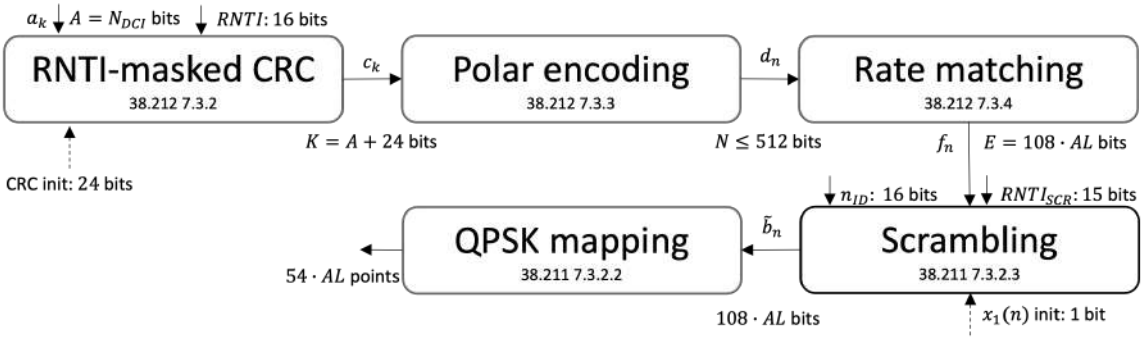


Figure 2.2: PDCCH DCI message generation steps.

2.3.3 Linear Inversion Technique for RNTI and Scrambling ID Recovery

Once the DM-RS initialization and its ScramblingID N_{ID} are determined, *Golden Sniffer* proceeds to infer the user-specific parameters used for data scrambling on the PDCCH, namely the RNTI and the data ScramblingID n_{ID}. This is achieved by exploiting the linear dependence between the encoded and scrambled bits in the PDCCH transmission process. The key idea of our approach is that the scrambling process applied to the PDCCH bits is linear (mod 2), meaning that each transmitted bit is the XOR of the encoded DCI bit and a pseudo-random Gold sequence determined by a few parameters. These parameters include unknown quantities, such as the DCI bits, the RNTI, and the cell or user-specific n_{ID}, and known ones, such as the CRC and the initialization sequence $x_1(n)$. By exploiting this linear relationship, we can express the received scrambled bits as a system of linear equations in the unknown parameters. Solving this system allows direct recovery of n_{ID} and RNTI without resorting to brute-force search over 2^{31} candidates. Once possible DCI sizes are known, this inversion operates in constant time, $O(1)$, drastically reducing computational load compared to traditional exhaustive methods.

Although scrambling with the sequence $c(n)$ might seem to make recovering the initialization c_{init} from the scrambled bits $\tilde{b}(i)$ challenging, since the original bit sequence $b(i)$ appears to act like a one-time pad, rendering c_{init} unrecoverable if $b(i)$ were truly random, closer examination reveals that $b(i)$ is a structured sequence constrained by the encoding process, including linear block codes and CRCs. This structured nature of $b(i)$ enables exploitation of inherent linearity to recover the RNTI and other unknowns. As reported in Figure 2.2, the details of DCI generation steps are:

Step 1: CRC Attachment and RNTI Masking. The DCI payload $\mathbf{a}$ consists of $A = N_{\text{DCI}}$ bits. A 24-bit CRC is appended to $\mathbf{a}$ to form the sequence $\mathbf{c}$ of length $K = A + 24$ bits. This combination of *CRC Computation and Attachment* with *RNTI Masking* can be expressed

as:

$$\mathbf{c} = \underbrace{\left[\mathbf{a}_{\text{init}}^{\text{CRC}}\ \mathbf{a}\right] \begin{bmatrix} \mathbf{0}_{24\times A} & \mathbf{G}_{K\times 24}^{\text{CRC}} \\ \mathbf{I}_A \end{bmatrix}}_{\text{CRC Computation and Attachment}} + \underbrace{\mathbf{a}_{\text{RNTI}} \left[\mathbf{0}_{16\times K-16}\ \mathbf{I}_{16}\right]}_{\text{RNTI Masking}} \tag{2.11}$$

In *CRC Computation and Attachment*, $\mathbf{a}_{\text{init}}^{\text{CRC}}$ is the 24-bit initial CRC register state, and $\mathbf{G}_{K\times 24}^{\text{CRC}}$ is the CRC generator matrix that transforms the payload $\mathbf{a}$ into the appended CRC bits. The $A \times A$ identity matrix $\mathbf{I}_A$ preserves the original unaltered payload $\mathbf{a}$ within $\mathbf{c}$, while the zero matrix $\mathbf{0}_{24\times A}$ hides the initialization from the output.

The CRC computation fundamentally calculates the remainder of a division by a generator polynomial, a linear operation over the binary field $\mathbb{F}_2$. Note that the CRC register is initialized to a non-zero value so that the operation results non-linear, i.e., affine linear. Embedding this initialization in the unknowns renders the overall operation linear by decomposing the CRC output into two linear components: Free Evolution (i.e., the output from initial CRC state with no input, $\mathbf{a} = \mathbf{0}$) and Forced Evolution (i.e., the output from a zero initial CRC state with actual input payload $\mathbf{a}$). Thus, the CRC output can be expressed as the sum in $\mathbb{F}_2$ of these two linear contributions.

In *RNTI masking*, the 16-bit RNTI $\mathbf{a}_{\text{RNTI}}$ is XORed with the CRC, uniquely binding the control message to its user. The zero matrix $\mathbf{0}_{16\times K-16}$ prevents RNTI influence on payload bits, while the identity matrix $\mathbf{I}_{16}$ aligns RNTI bits with the CRC portion of $\mathbf{c}$. The matrix formulation in Equation (2.11) demonstrates that $\mathbf{c}$ is a linear combination of $\mathbf{a}$ and the RNTI, ensuring that both CRC attachment and RNTI masking are linear operations over $\mathbb{F}_2$.

Step 2: Polar Encoding and Rate Matching. The sequence $\mathbf{c}$ is polar encoded using a generator matrix $\mathbf{G}_{\text{polar}} \in \mathbb{B}^{K\times N}$ to produce a N-bit codeword $\mathbf{d} = \mathbf{c} \cdot \mathbf{G}_{\text{polar}} \in \mathbb{B}^N$. Subsequently, $\mathbf{d}$ undergoes rate matching via matrix $\mathbf{G}_{\text{RM}}$ to fit the allocated PDCCH resources, yielding a sequence $\mathbf{f} = \mathbf{d} \cdot \mathbf{G}_{\text{RM}}$. with length $E = 108 \times \text{AL}$, i.e., the number of Control Channel Elements (CCEs) assigned. This linear transformation selects, punctures (if $E < N$), or repeats bits (if $E > N$) from $\mathbf{d}$ to achieve the required code rate. Since each CCE spans $N_{\text{sc}}^{\text{CCE}} = 6 \times N_{\text{sc}}^{\text{RB}} = 72$ subcarriers, and one out of every four allocated subcarriers is reserved for DM-RS, only the remaining three-fourths, i.e. $54 \times \text{AL}$ subcarriers are available for DCI data. Given that QPSK modulation maps each subcarrier to 2 bits, the total bits that can be transmitted for the DCI is $E = 108 \times \text{AL}$ bits, ensuring that the rate-matched sequence $\mathbf{f}$ fits within the allocated PDCCH resources, according to the AL and modulation scheme.

Step 3: Scrambling. The rate-matched sequence $\mathbf{f}$ is scrambled by XORing it with a scrambling sequence $\mathbf{c}_{\text{scr}}$ generated from $c(n)$ as $\tilde{\mathbf{b}} = \mathbf{f} \oplus \mathbf{c}_{\text{scr}}$, which depends on the RNTI RNTI_{SCR} and the Scrambling ID n_{ID}. The scrambling operation is linear, and $\mathbf{c}_{\text{scr}}$ can be

represented as:

$$\mathbf{c}_{\text{scr}} = [\mathbf{f}, \mathbf{a}_{\text{RNTI}_{\text{SCR}}}, \mathbf{a}_{n_{\text{ID}}}, \mathbf{a}_{\text{init}}^{x_1}] \cdot \begin{bmatrix} \mathbf{I}_E \\ \mathbf{G}_{E\times 31}^{\text{SCR}} \\ \mathbf{G}_{E\times 1}^{x_1} \end{bmatrix} \tag{2.12}$$

where $\mathbf{G}_{E\times 31}^{\text{SCR}}$ and $\mathbf{G}_{E\times 1}^{x_1}$ represent the relationship between the scrambling sequence and its initializations.

Step 4: Modulation and Mapping. The scrambled E bits $\tilde{\mathbf{b}}$ are mapped to QPSK symbols and transmitted over the allocated PDCCH physical resources. The PDCCH DM-RS subcarriers can be exploited to extract the *pdcch-DMRS-ScramblingID* using the phase-based method in Subsection 2.3.2.

By combining and rearranging the described operations, we can represent the entire process as a single linear equation:

$$\tilde{\mathbf{b}} = [\mathbf{a}, \mathbf{a}_{\text{RNTI}}, \mathbf{a}_{\text{RNTI}_{\text{SCR}}}, \mathbf{a}_{n_{\text{ID}}}, \mathbf{a}_{\text{init}}^{\text{CRC}}, \mathbf{a}_{\text{init}}^{x_1}] \cdot \mathbf{G}_{\text{DCI}} = \mathbf{a}_{\text{tot}} \cdot \mathbf{G}_{\text{DCI}} \tag{2.13}$$

where $\mathbf{a}_{\text{tot}}$ is the vector of unknowns and known initialization, and $\mathbf{G}_{\text{DCI}}$ is the combined generator matrix representing all encoding and scrambling operations. Given that the initializations $\mathbf{a}_{\text{init}}^{\text{CRC}}$ and $\mathbf{a}_{\text{init}}^{x_1}$ are known, the unknown components can be isolated as follows:

$$\tilde{\mathbf{b}} - [\mathbf{a}_{\text{init}}^{\text{CRC}}, \mathbf{a}_{\text{init}}^{x_1}] \cdot \mathbf{G}_{\text{DCI}}^{\text{known}} = [\mathbf{a}, \mathbf{a}_{\text{RNTI}}, \mathbf{a}_{\text{RNTI}_{\text{SCR}}}, \mathbf{a}_{n_{\text{ID}}}] \cdot \mathbf{G}_{\text{DCI}}^{\text{unk}} \tag{2.14}$$

A solution is considered valid, according to [2] Clause 7.3.2.3, if either the CRC masking RNTI matches the RNTI estimated from DM-RS and the Scrambling ID matches the DM-RS Scrambling ID, i.e., $\mathbf{a}_{n_{\text{ID}}} = N_{\text{ID}} \wedge \mathbf{a}_{\text{RNTI}} = \mathbf{a}_{\text{RNTI}_{\text{SCR}}}$ with N_{ID} from (2.10); or if the Scrambling ID coincides with PCI and the RNTI estimated from the PDCCH scrambling is zero, i.e., $\mathbf{a}_{n_{\text{ID}}} = N_{\text{ID}}^{\text{cell}} \wedge \mathbf{a}_{\text{RNTI}_{\text{SCR}}} = 0$.

This system comprises $A + 47$ unknowns, including the A-bit DCI payload, the 16-bit RNTI, the possible 16-bit RNTI_{SCR}, and the Scrambling ID n_{ID}, and $108 \times \text{AL}$ equations (the length of $\tilde{\mathbf{b}}$). When $108 \times \text{AL} > A + 47$, the system is overdetermined and yields a unique solution. However, as A increases, the risk of false detections increases; with this approach, for $A > 50$, the sniffer requires $\text{AL} > 1$. Although $A = N_{\text{DCI}}$ is initially unknown, once a valid solution is found, it can be cached to streamline decoding of subsequent RNTI-related DCIs, reducing computational complexity.

This solution models the DCI chain as a linear system over $\mathbb{F}_2$ and solves it in $O(1)$ per candidate size. Given c_{init} and pruned C_L, it replaces the 16-bit Scrambling-ID and the 16-bit RNTI brute-force ($C_R C_S = 2^{32}$) with constant-time inversion and validation. Caching locks parameters per RNTI for subsequent DCIs.

2.3.4 Detection of CSI-RS

2.3.4.1 NZP-CSI-RS Detection

Detection of single-layer NZP-CSI-RS can be done using the phase-based method detailed in Subsection 2.3.2. The DM-RS sequence is scrambled using the *scr_ID* or *sequenceGenerationConfig*, both specified in the RRC Setup *NZP-CSI-RS-Resource IE*. The number of dedicated subcarriers depends on the configured density, ranging from one subcarrier every 24 (for density 0.5) to four subcarriers every 24 (for density 3). Assuming the worst condition, i.e., density 0.5, the method applies to resource grids with $N_{RB} \geq 32$.

This solution gets rid of a brute force search over $C_X = 2^{16}$ possibilities, eliminating 16 bits.

2.3.4.2 Adaptive ZP-CSI-RS Detection and Exclusion:

In order to detect the ZP-CSI-RS-bearing symbols, for every $0 \leq l < N_{\text{symb}}^{\text{frame}} - 1$, we mark as busy the RBs which contain at least one subcarrier k such that

$$|Y[k,l]|^2 \geq \beta_{\text{busy}}^{\text{RB}} P_{\text{noise}} \tag{2.15}$$

where $\beta_{\text{busy}}^{\text{RB}}$ is a threshold that controls the trade-off between false positives and false negatives. Under the null hypothesis of *noise only*, each complex sample $Y[k,l]$ can be modeled as Gaussian with zero mean and variance P_{noise}, i.e., $Y[k,l] \sim \mathcal{CN}(0, P_{\text{noise}})$. The corresponding power $|Y[k,l]|^2$ follows an exponential distribution with mean P_{noise}. Normalizing by the noise power, we define $X = \frac{|Y[k,l]|^2}{P_{\text{noise}}}$. The probability that noise alone exceeds the threshold $\beta_{\text{busy}}^{\text{RB}}$ (i.e., a false positive) is $P_{\text{FP}} = \Pr(X \geq \beta_{\text{busy}}^{\text{RB}}) = \int_{\beta_{\text{busy}}^{\text{RB}}}^{\infty} e^{-x} dx = e^{-\beta_{\text{busy}}^{\text{RB}}}$. By arbitrarily setting $\beta_{\text{busy}}^{\text{RB}} = 10$, we obtain $P_{\text{FP}} = e^{-10} \simeq 4.54 \cdot 10^{-5}$.

This means that, in the absence of any signal, a given subcarrier has only a 0.0045% chance of being falsely marked as busy, which provides a low false positive rate while still allowing reliable detection of active RBs.

With reference to busy RBs within symbol l, we choose the RB with highest mean squared value, having the first subcarrier index equal to $\bar{k}$. Then we search for an offset k' within the RB for which the following inequalities hold:

$$\begin{cases} \left|Y[\bar{k}+i,l]\right|^2 < \beta_{\text{busy}}^{\text{RB}} P_{\text{noise}} & k' \leq i < k' + N_{\text{CSI}}^{\text{RB}} \\ \left|Y[\bar{k}+i,l]\right|^2 > \beta_{\text{busy}}^{\text{RB}} P_{\text{noise}} & \text{otherwise} \end{cases} \tag{2.16}$$

where $N_{\text{CSI}}^{\text{RB}}$ is the number of ZP-CSI-RS subcarriers within the RB, which may equal 4 for single-symbol ZP-CSI-RS, or 2 for double-symbol ZP-CSI-RS. In this latter case, the inequalities (2.16) must hold with the same offset k' for two consecutive symbols [5].

Figure 2.3 shows an example of subcarrier power levels across the busy RBs, in a symbol l containing a ZP-CSI-RS with $N_{\text{CSI}}^{\text{RB}} = 4$. The red circles represent the power of subcarriers within the RB with highest mean squared value; the subcarriers starting from index $k' = 8$ are

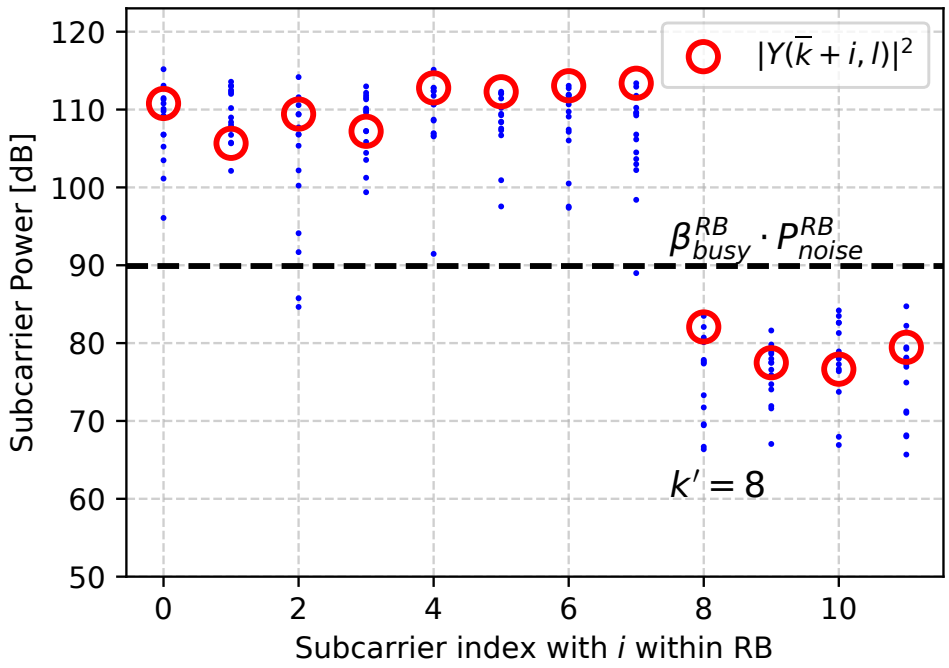


Figure 2.3: Example of subcarriers power levels across 15 busy RBs in a symbol containing a ZP-CSI-RS. The red circles represent the subcarriers power of the RB with highest mean squared value.

clearly below $\beta^{\text{RB}}_{\text{busy}} P_{\text{noise}}$, while all other subcarriers exceed the threshold, satisfying (2.16).

Symbol indexes for which (2.16) is satisfied for some k' will be denoted with $\bar{l}$. Since the periodicity and offset of the ZP-CSI-RS signal is measured in slots, the symbol index $\bar{l}$ is mapped to a tuple (s, l'), where $s = \lfloor \bar{l} / N^{\text{slot}}_{\text{symb}} \rfloor$ is the slot index and $l' = \bar{l} \bmod N^{\text{slot}}_{\text{symb}}$ is the symbol index within slot s, with $N^{\text{slot}}_{\text{symb}} = 14$. The detection algorithm then analyzes the distribution of symbol indices l' to determine the ZP-CSI-RS position with a slot, and the indexes s to infer the ZP-CSI-RS periodicity.

Estimation of ZP-CSI-RS period Once the first ZP-CSI-RS is detected, the ZP-CSI-RS period can be estimated by looking at the next occurrence of ZP-CSI-RS and measuring the time interval elapsed between the two consecutive observations. However, since the ZP-CSI-RS observation depends on the occupancy state of the relevant RB, the measured interval may span multiple actual periods. We propose to repeat the measurement of the time intervals between consecutive observations of ZP-CSI-RS for detecting the real periodicity as the Greatest Common Divisor (GCD) of the observed intervals.

The estimation process can be modeled as a transient discrete Markov chain, in which each transient state represents the current estimate of the ZP-CSI-RS period and the absorbing state represents the smallest possible estimate equal to the real period.

Figure 2.4 shows an example of intervals measured between consecutive ZP-CSI-RS detections and relevant model variables over time. The discrete events triggering the chain evolution are given by the time instants at which a new ZP-CSI-RS is observed. Let I_n be the measurement of the time intervals (in slots) elapsed between the consecutive $(n-1)$-th and n-th observations of ZP-CSI-RSs and x_n be the current estimate of the period at time n.

Let p be the detection probability of a real ZP-CSI-RS in one slot. We assume that false

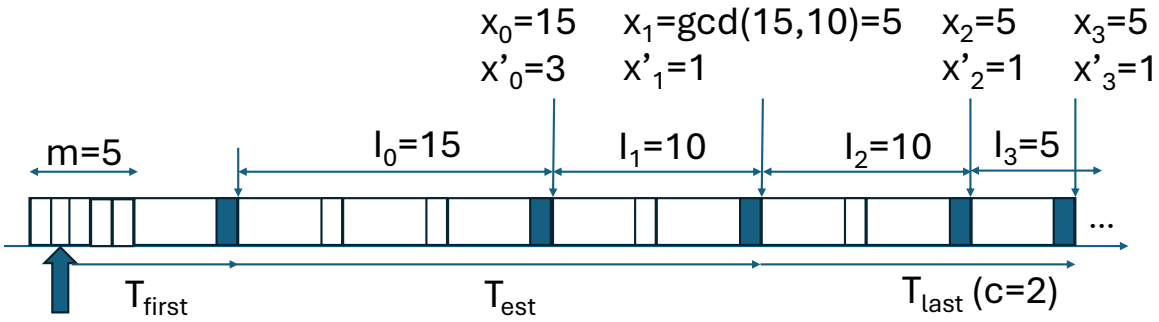


Figure 2.4: Detection of the ZP-CSI-RS period: outlined slots denote slots in which ZP-CSI-RS is transmitted; whereas filled slots denote busy slots (traffic present) where detection is possible.

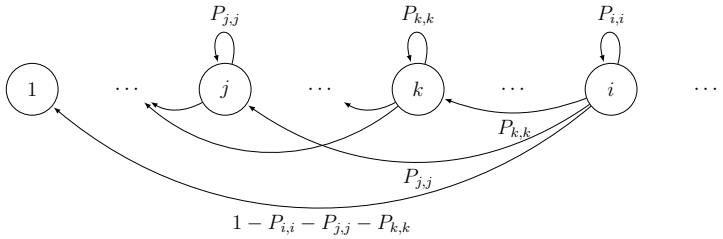


Figure 2.5: Markov model for estimating the ZP-CSI-RS period. Focus on state i transition probabilities assuming $i = j \cdot k$, with j and k nontrivial prime factors.

detections never occur, while missed detections are due to empty RBs or to a failure in the detection process based on power measurements described above. If consecutive RB allocations are independent, we can assume that p is equal to the product of the probability to find a given RB as busy (i.e. by the normalized cell load) and the probability that the power measurements fall in the expected ranges. Note that the first estimate is equal to the first measurement, $x_0 = I_0$, while in subsequent transition times (when other measurements are available) $x_n = \gcd\{x_{n-1}, I_n\}$. Since all estimates are multiples of the unknown period of m slots, with $m \in \{4, 8, 12, 16, 32, 64, 5, 10, 20, 40, 160, 320, 640\}$, we can also consider the sequence of integer values x'_n that quantify these multiples (in Figure 2.4, with $m = 5$, $x_0 = 15$ and $x'_0 = 3$).

Figure 2.5 represents the general Markov chain that applies regardless of the m value configured in the cell. Suppose that at time n the system is in state $x'_n = i$ (i.e. the current estimate of the period is $i \cdot m$ slots, with m unknown). When a new interval measurement $I_{n+1} = y \cdot m$ slots is available, we can refine the period estimate as $x_{n+1} = \gcd\{x_n, I_{n+1}\} = \gcd\{i, y\} \cdot m$, which can be modeled with a transition to a new state $x'_{n+1} = j \leq i$, being $j = \gcd\{i, y\}$. Since the probability that $I_{n+1} = y \cdot m$ is equal to the probability that $y - 1$ schedules of ZP-CSI-RS transmissions are missed, we can express the transition probability $P_{i,j}$ from a generic state i towards a generic state j as:

$$P_{i,j} = \sum_{\substack{y \geq 1: \\ \gcd(i,y) = j}} (1-p)^{y-1} \cdot p \tag{2.17}$$

In particular, $P_{i,j} = 0$ if $j > i$, and the chain is *monotone* in the sense that i can only move to one of its divisors (including itself when y is a multiple of i). The probability to remain in the same state $P_{i,i}$ is given by the probability to observe a measurement which coincides or is an integer multiple of the current one:

$$P_{i,i} = \sum_{m} (1-p)^{m \cdot i - 1} \cdot p = \frac{p}{1-p} \cdot \frac{(1-p)^i}{1-(1-p)^i} \tag{2.18}$$

In principle, arbitrarily many periods can be missed, so the state space is unbounded. However, each realized trajectory is confined to a *finite* divisor lattice of the first observed interval I_0.

Let $\pi(n)$ be the state probability vector at the discrete time n. The components $\pi_i(0) = Pr\{x'_0 = i\}$ follow a geometrical distribution:

$$\pi_i(0) = Pr\{I_0 = i \cdot m\} = (1-p)^{i-1} p, \qquad i = 1, 2, \ldots \tag{2.19}$$

Note that $\pi_1(0) = p$ corresponds to immediate absorption (the first interval already equals the true period).

Average estimation time In order to compute the total time required for estimating the ZP-CSI-RS periods, we can then identify three components: i) the time T_{first} required for the first ZP-CSI-RS detection; ii) the transient time T_{est} of the Markov chain, from the second ZP-CSI-RS detection (which affects the initial state of the chain) to the absorption state; iii) the time T_{last} required to confirm that the absorption state has been reached.

Obviously, ZP-CSI-RS detection can be performed only at scheduled ZP-CSI-RS instants. Once the attacker is activated, we can assume that the waiting time to the next scheduled interval is uniformly distributed in the interval $[0, m]$; from this initial time, we need additional k periods with $k \in 0, 1, 2, \ldots$ with probability $(1-p)^k \cdot p$. It follows that on average we need

$$T_{\text{first}} = \frac{m}{2} + \frac{1-p}{p} \cdot m \tag{2.20}$$

slots for the first successful ZP-CSI-RS detection.

For finding the transient time to the chain absorption state, let consider the usual decomposition of the transition matrix, with the absorbing state $(1, 1)$, $\mathbf{P}$ as

$$\mathbf{P} = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{R} & \mathbf{Q} \end{bmatrix} \tag{2.21}$$

The fundamental matrix is

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} = \mathbf{I} + \mathbf{Q} + \mathbf{Q}^2 + \cdots . \tag{2.22}$$

Let $\boldsymbol{\tau} = \mathbf{N} \cdot \mathbf{1}$ be the vector whose each component $\tau(r)$ gives the expected numbers of chain transitions (interval updates) to reach the absorbing state from each transient state

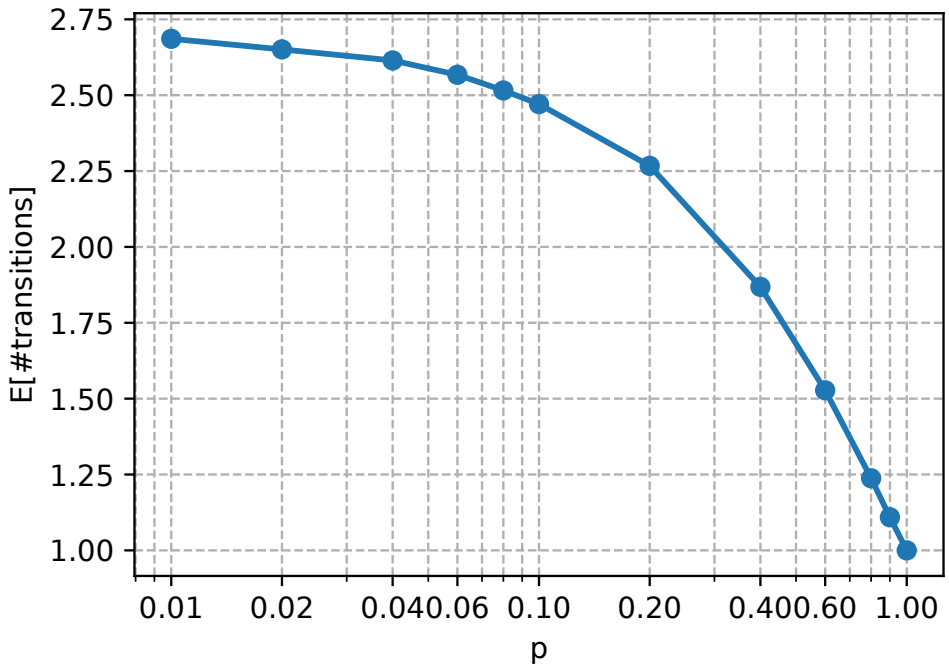


Figure 2.6: Expected number of transitions to the absorbing state, $\mathbb{E}[\#\text{transitions}]$, versus the probability p.

$r+1$ [48]. Taking into account the geometrical distribution of $\pi(0)$, we can then derive the average number of transitions as:

$$\mathbb{E}[\#\text{transitions}] = \sum_{r=2}^{\infty}(1-p)^{r-1}p \cdot \tau(r) \tag{2.23}$$

The absorption time in slots can be expressed by considering the Wald's lemma (according to which the average number of transitions can be multiplied by the average transition time when the transition intervals are i.i.d and the increments are integrable):

$$T_{\text{est}} = \mathbb{E}[\#\text{transitions}] \cdot \frac{m}{p} \tag{2.24}$$

It is interesting to note that, although the average absorption time grows as $1/p$, the average number of transitions is limited even for very small p values, as shown in Figure 2.6.

Finally, if we wait for an additional c observations to confirm that subsequent measurements do not fall below the current estimate, the expected additional time is

$$T_{\text{last}} = \frac{m}{p} \cdot c \tag{2.25}$$

The total expected estimation time is then $T_{\text{first}} + T_{\text{est}} + T_{\text{last}}$.

If decoding succeeds after adjusting for inferred ZP-CSI-RS locations, we cache the ZP-CSI-RS configuration for future attempts. Algorithm 1 summarizes main steps.

This solution eliminates a brute force search over $C_Z \simeq 2^{12}$ options by caching the found ZP-CSI-RS configuration. This eliminates the ambiguity, avoiding a brute force search over $C_Z \simeq 2^{12}$ options and eliminating $\simeq$12 bits.

Algorithm 1 Blind Detection of ZP-CSI-RS Symbols

acquire IQ samples, perform cell synchronization and FFT processing to obtain $Y[k, l]$
estimate the noise floor P_{noise} from the unused subcarriers around the PSS
initialize the set of candidate symbol indexes C_2, C_4
for all symbols l in the frame **do**
    search the RB with the highest mean squared value and at least one subcarrier with power greater than $\beta_{\text{busy}}^{\text{RB}} P_{\text{noise}}$, save the index of its first subcarrier as $\bar{k}$
    search the offset k' within the RB starting at $\bar{k}$ for $N_{\text{CSI}}^{\text{RB}} = \{2, 4\}$ consecutive subcarriers satisfying (2.16)
    if k' is found **then**
        map l to (s, l')
        if $N_{\text{CSI}}^{\text{RB}} = 2$ and the next symbol satisfies (2.16) **then**
            add (s, l', k') to the set C_2
        else if $N_{\text{CSI}}^{\text{RB}} = 4$ **then**
            add (s, l', k') to the set C_4
        end if
    end if
end for
return the set with higher cardinality between C_2 or C_4

2.4 PDCCH & PDSCH Decoding Performance Evaluation

In this section, we describe *Golden Sniffer* operational flow and evaluate its performance in advanced 5G NR scenarios:

- **Controlled Testbed:** A private SDR-based 5G SA deployment based on srsRAN [49] and Open5GS [50] with COTS UEs, and configurable BW, SCS, and transmit power. This setup allows us to validate the sniffer's accuracy under varying radio conditions and system parameters, as well as to stress-test the iterative search and decoding algorithms.
- **Commercial Networks:** 5G operators' cells in n78 band (FSVA3044 VSA [51] at 3680.01 MHz, with 217 RBs for 80 MHz BW, $\mu = 1$, and PCI 710) and n28 band (USRP N310 [52] at 763 MHz, with 52 RBs for 10 MHz BW, $\mu = 0$, and PCI 175). These captures span FDD and TDD modes, confirming that *Golden Sniffer* decodes real-world DL traffic under complex configurations, i.e., user-specific BWPs, multi-symbol DCIs, and user-specific scrambling IDs.

Ethical Considerations. *Golden Sniffer* operates as a fully passive tool, avoiding any transmission/interference. Experiments on commercial networks were limited to control-plane metadata and encrypted traffic patterns, with data minimized, storage limited, and no payload decoding or retention.

2.4.1 Sniffer Operational Flow

Figure 2.7 outlines the end-to-end *Golden Sniffer*'s fully passive pipeline, from raw IQ capture to extracting user-plane traffic. The workflow shows how the Section 2.3 techniques, enable robust decoding of 5G NR DL control and data channels, despite the parameters being

concealed in encrypted RRC messages.

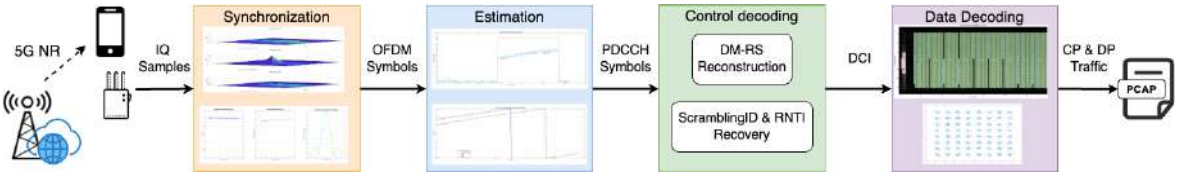


Figure 2.7: Operational workflow of *Golden Sniffer*, showing the four sniffing main stages.

Hardware & Processing. Raw IQ data was captured with an USRP N310 SDR [52] and processed by a *MATLAB*-based prototype at ∼30 s per 1 s on an Apple M3 Pro CPU, indicating feasibility of 5G NR sniffing on commodity hardware.

Workflow. The main stages in Figure 2.7 are described below:

1. **Synchronization & System Information & NZP-CSI-RS:** *Golden Sniffer* (i) processes the SDR-acquired raw IQ data to lock onto the cell by detecting the SSB and decoding the PBCH for basic cell parameters. (ii) iterates over every possible NZP-CSI-RS configuration, using the phase-based method to detect their position and periodicity.
2. **Estimation (PDCCH Search Space Discovery)**: *Golden Sniffer* uses the energy thresholding method to detect possible PDCCH allocations, iterating over possible CORESETs and ALs, applying the phase-based method to detect DM-RS sequences and their initialization parameters.
3. **Control Decoding (PDCCH)**: By iterating over possible DCI sizes and applying the linear inversion technique, *Golden Sniffer* recovers the user-specific RNTI and the Scrambling ID without brute-forcing.
4. **Data Decoding (PDSCH)**: To infer the user-specific BWP configuration, *Golden Sniffer*: (i) iterates over the possible interpretations of the FDA field, (ii) searches for PDSCH DM-RS positions and initialization parameters, (iii) regenerates the corresponding Gold sequences to estimate the channel, then attempts decoding.
   If the decoding is successful, the DCI size, along with the RNTI, ScramblingID, and BWP configuration, are cached, and the decoded message is optionally saved.

2.4.2 Controlled Testbed: UE Traffic Profiling

To evaluate *Golden Sniffer* for network analytics, we setup a 5G testbed with UE-specific BWPs and multi-symbol DCIs, that is a setting in which competing approaches [28, 29] are not effective.

Validation. We disabled encryption in our testbed and confirmed user-plane decoding by intercepting PDSCH transmissions and recovering ICMP Echo Replies; the Wireshark evidence is shown in Figure 2.8, demonstrating accurate demodulation/decoding when security is disabled and supporting comprehensive control- and user-plane capture. This highlights *Golden Sniffer*'s robustness as a platform for 5G network analytics and troubleshooting.

Application-Traffic Profiling. When encryption is enabled, user-plane payloads remain undecipherable. However, *Golden Sniffer* can still extract control-plane information from

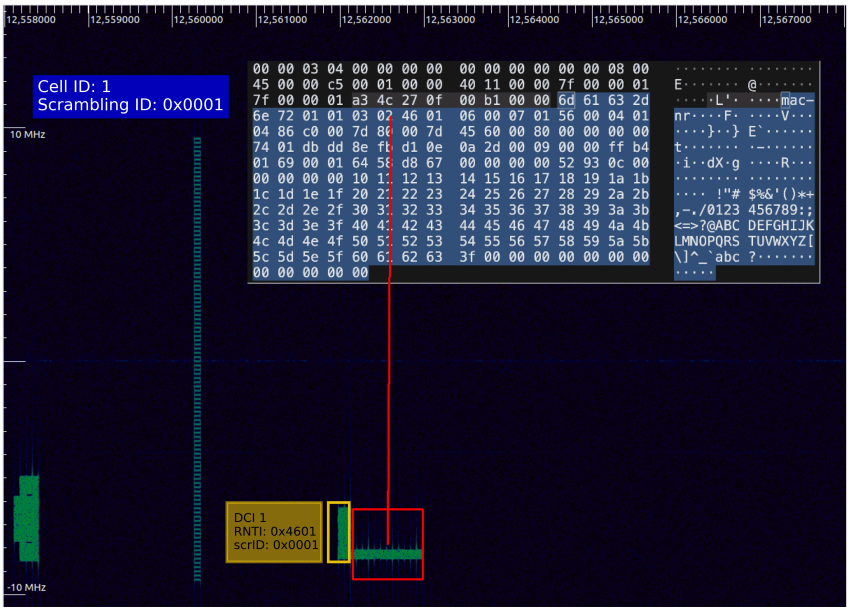


Figure 2.8: Controlled testbed: ICMP Echo Reply packet (highlighted in Wireshark) recovered by *Golden Sniffer* from PDSCH transmissions with encryption disabled.

control messages and PDSCH metadata (e.g., MCS, Transport Block Size (TBS), scheduling times). This sniffing methodology reliably decodes DCIs in the PDCCH and extracts features associated to every encrypted PDSCH transmission, including MCS, allocated RB (Located Resource Block (LRB)), and BWP, without missing any instance. This sniffer enables the extraction of extremely detailed traffic patterns, facilitating precise temporal analysis and permitting the potential application of advanced deep learning models to accurately infer user-specific activity. This aspect will be object of the following chapter of this thesis.

2.4.3 Decoder Performance

By leveraging the controlled testbed, we quantify how reliably *Golden Sniffer* decodes PDCCH/DCI messages.

AWGN Sensitivity.

To quantitatively assess *Golden Sniffer*'s performance, we evaluated its decoding success rate against decreasing PDCCH SNR by injecting Additive White Gaussian Noise (AWGN) into a high-SNR (23.7 dB) IQ capture. Results in Figure 2.9, reveal a waterfall curve for both UL and DL DCI grants: a high-SNR plateau with over 90% success above 16.5 dB, a sharp performance cliff where success drops from 88.7% at 15.4 dB to 12.9% at 12.8 dB, and a failure floor with a near-zero success rate below 12.5 dB.

We validated this trend against a suitable theoretical benchmark. Since our method's success is ultimately determined by the integrity of the 63-bit information block (39-bit DCI

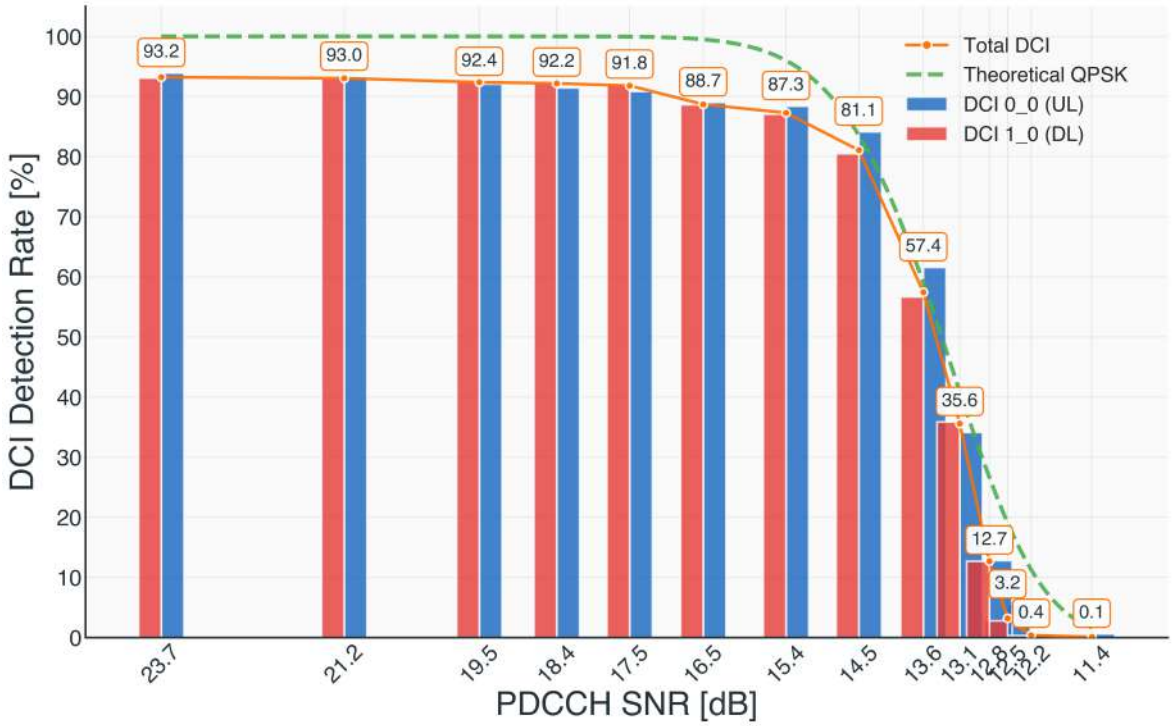


Figure 2.9: DCI detection rate vs. SNR. Measured data (orange line and bars) is benchmarked against the theoretical block success probability (dashed green line).

payload + 24-bit CRC), extended to a transmitted sequence $E = 108 \times 4 = 432$ bits (assuming $AL = 4$), a fitting benchmark is the block success probability over these 432 bits. Derived from the Bit Error Rate (BER) for Gray-coded QPSK $P_b(\gamma_b) = Q\left(\sqrt{2\gamma_b}\right)$, with γ_b the SNR per bit, the block success probability is $P_{\text{succ}}^{\text{DCI}}(\gamma_b) = \left[1 - P_b(\gamma_b)\right]^{432} = \left[1 - Q\left(\sqrt{2\gamma_b}\right)\right]^{432}$. The shape of our measured data aligns remarkably well with this theoretical curve after accounting for an estimated 7 dB implementation loss from our SDR-based receiver. This alignment supports the robustness of *Golden Sniffer* approach.

Baseline comparison. Compared to 5GSniffer [29], *Golden Sniffer* demonstrates substantial improvements in both detection accuracy and runtime efficiency. When evaluated on identical high-SNR IQ capture, as reported in Table 2.2, our tool achieves a DCI detection rate of 94.7%, outperforming the open-source baseline, which attains only 6.8%. Furthermore, *Golden Sniffer* completes the full decoding pipeline in $\sim$ 30 seconds, with a more than 5000 times runtime improvement with respect to 5GSniffer[1]. This performance gap underscores the effectiveness of our targeted sniffer search-space reduction and algorithmic optimizations.

NLOS robustness. To evaluate *Golden Sniffer*'s robustness, we assessed its decoding success rate under Non-line-of-sight (NLOS) conditions using 3GPP Clustered Delay Line (CDL) channel models [53]. CDL models simulate realistic multipath propagation by defining clustered taps with configurable delay spreads and angular dispersion. We conducted 10^4 decoding attempts per scenario, fixing the Doppler shift to 1 Hz (quasi-static) while varying

[1]Runtimes for 5GSniffer were measured under an experimental cap of 30 tested scrambling IDs and renormalized to 2^{16}. Beyond this cap, the host encountered an out-of-memory error.

Table 2.2: Baseline Performance Comparison: DCI Detection Rate and Total Decoding Time.

	Golden Sniffer	5GSniffer
DCI Detection Rate [%]	94.7	6.8
Total Decoding Time [s]	31.54	$168.2 \cdot 10^3$

the channel model, propagation environment, and delay spread. In a moderately dispersive outdoor environment (CDL-A, open area, 300 ns delay spread), the sniffer achieved a DCI detection rate of 70%, indicating good resilience to limited multipath. In an urban macro-cell setting (CDL-C, 500 ns), characterized by denser scattering and more reflective surfaces, performance dropped to 26%. Under a severely dispersive urban NLOS condition (CDL-E, 1 µs), decoding success fell to 12%, highlighting the challenges posed by rich multipath and long delay spreads.

Timing micro-benchmarks. To profile *Golden Sniffer*, we timed its four stages described in Subsection 2.4.1 in the three identified operational SNR zones. In the high-SNR plateau ($\geq$ 19.6 dB), Stage 3 consumed 97.4% of the runtime, with Stages 1, 2, and 4 taking 0.2%, 1.5%, and 1.0%, respectively. This pattern held on the performance cliff (15.4–12.8 dB), where Stage 3's share increased slightly to 98.0%, while the other steps accounted for 0.2%, 1.1%, and 0.7%. In the failure floor region below 12.5 dB, the sniffer could not complete decoding. These results confirm that the exhaustive blind DCI search in Stage 3 is the dominant bottleneck.

2.4.4 Commercial Networks: Real-World Control Data Decoding

To validate *Golden Sniffer* in real-world deployments, we conducted in-the-wild tests on commercial national operator networks. In these scenarios, despite the variability in network configurations and environmental challenges, *Golden Sniffer* consistently extracted unique Scrambling IDs and RNTIs from the DL control channels.

Validation. Our analysis reveals that the common assumption used in competing approaches [28, 29], that the ScramblingID defaults to the PCI, does not hold in operational networks. Indeed, we observed that commercial operators often configure user-specific scrambling parameters to enhance security, rendering brute-force methods ineffective. Figure 2.10 illustrates a commercial capture where multiple *Non-Fallback* DCI formats are observed. User-specific RNTIs and ScramblingIDs are color-coded to clearly demonstrate that *Golden Sniffer* reliably differentiates users and decodes control information, even in complex commercial network. Furthermore, the approach is robust against MIMO attackers.

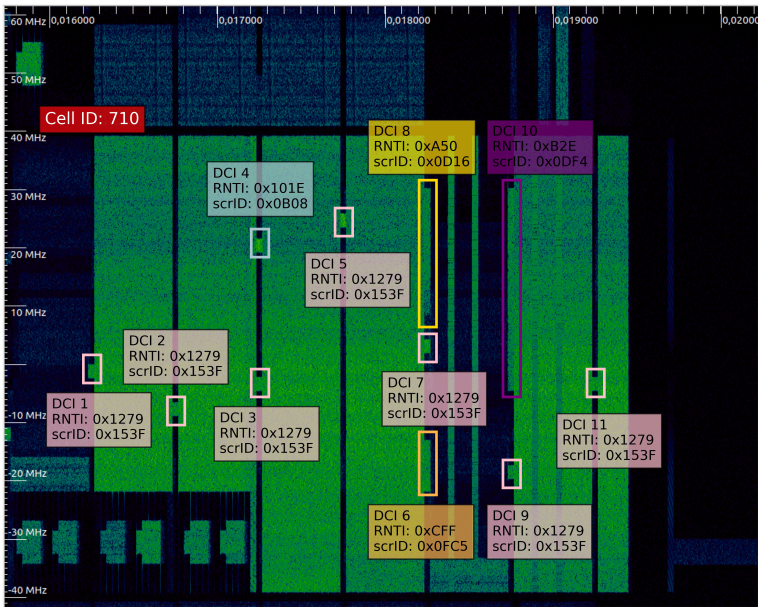


Figure 2.10: 5G N78 commercial network: multiple *Non-Fallback* DCI formats with highlighted user-specific RNTIs and ScramblingIDs successfully extracted.

2.5 Conclusion

Sniffing user-specific control and data traffic over 5G NR has long been considered challenging due to the concealment of crucial per-user radio interface parameters (DCI, RNTI, ScramblingID). The method presented in this chapter faces this notion. Rather than optimizing brute-force methods or relying on the clear-text RRC registration procedure as in prior works, we introduce a novel approach that exploits the linearity and non-cryptographic nature of Gold sequences to blindly recover these parameters, substantially reducing the sniffer parameter space to just a few alternatives. Practicality is demonstrated through a concrete 5G sniffing tool capable of decoding both control and data channels, even under complex user-specific BWP configurations and real-world conditions (tested on captures from national operators). While currently limited to DL interception and not yet real-time due to interpreted language implementation, its algorithmic foundation, avoiding brute-force testing, suggests potential for re-engineering into operational use.

In addition, *Golden Sniffer* will be directly employed in the next chapter to investigate Traffic Flow Confidentiality (TFC). By decoding control and data channels and extracting per-UE parameters such as DCI fields, RNTI, and ScramblingIDs, the sniffer enables the collection of detailed traffic-related metadata, including SFN, slot index, MCS, and TBS. These features form the basis for per-flow analysis in the TFC study, allowing the assessment of how much protocol-level scheduling information and resource allocation patterns can reveal about user traffic. Overall, the findings of this chapter both justify and enable the systematic

TFC study that follows, closing the loop from passive observability to actionable privacy assessment.

Moreover, motivated by the practical feasibility of the attacks demonstrated above, we also propose and evaluate countermeasures specifically aimed at preventing the sniffer from operating successfully.

Chapter 3

Inferring Behavior: From Raw Signals to Application Fingerprinting

Decades of research on statistical traffic analysis have shown that encryption alone cannot fully conceal user activity—observable patterns in traffic timing and volume often reveal sensitive information. Despite this, user traffic over 5G NR is frequently assumed to resist such an analysis due to the perception that intercepting and decoding user-specific 5G traffic is technically prohibitive. However, advanced methods for eavesdropping on 5G NR traffic are currently emerging, challenging the above assumption and potentially exposing user-specific traffic patterns to adversarial analysis.

TFC is a critical set of techniques designed to obscure patterns in network traffic, thereby preventing adversaries from conducting traffic analysis to infer sensitive information. Even when data payloads are encrypted, the observable characteristics of traffic—such as packet sizes, timing, and flow patterns—can be exploited by advanced techniques, including neural networks, to infer significant details about the communication [54, 55, 56].

Recent advancements in neural networks have significantly enhanced traffic analysis capabilities, enabling attackers to accurately classify encrypted traffic by identifying patterns in packet sequences, inter-arrival times, or flow behavior. This highlights the urgency and importance of TFC techniques as a countermeasure, protecting not only the content of communication but also safeguarding metadata. Indeed, without effective TFC mechanisms, even encrypted communications can be profiled, revealing information such as the type of application, user behavior, or specific accessed resources [57, 58, 59, 60, 61]. Key Techniques for Traffic Flow Confidentiality include:

- **Traffic Padding:** This involves adding dummy data to ensure uniformity in traffic patterns. Neural networks are particularly adept at identifying statistical anomalies, and padding mitigates this by making all traffic appear consistent, reducing the information available for classification.

- **Packet Size Obfuscation:** By fragmenting or padding packets to create a uniform size, TFC can prevent neural networks from leveraging packet length as a classifica-

tion feature. Uniform packet sizes effectively neutralize this critical indicator used in encrypted traffic analysis.

- **Timing Variability:** Neural networks often exploit precise timing relationships between packets. By introducing random delays or jitter, TFC disrupts these patterns, making it significantly harder for models to infer meaningful relationships from timing data.
- **Dummy Traffic Generation:** Generating synthetic, meaningless traffic alongside legitimate data can confuse neural network classifiers. This additional noise in the dataset makes it difficult for the model to isolate genuine traffic patterns, reducing classification accuracy.
- **Route Randomization:** Dynamically altering the paths taken by packets across the network prevents attackers from constructing consistent flow patterns. This spatial variation complements other TFC techniques by further obscuring the end-to-end behavior of the communication.

3.1 Introduction

Since the earliest generations, cellular networks have been designed with a strong focus on preventing the leakage of user-related information, especially with the consistent use of OTA encryption and on the usage of temporary identifiers like the GUTI and the RNTI. The scientific research community has thoroughly investigated—and, at times, questioned—the design and implementation of these approaches, identifying vulnerabilities in both 4G [62, 63] and 5G networks [64], as well as configuration issues in real world deployments [65, 66]. These studies, while highlighting risks like user identification [18, 67], activity profiling [68], and location tracking [14, 67], are continuously providing pragmatic feedback that are contributing to refining both standards and implementations.

Conversely, the issue of TFC, widely documented in the literature [69, 70, 71], has been largely overlooked in the context of 5G NR. This oversight likely stems from a twofold (often implicit) assumption.

First, 5G's reliance on systematic encryption at a low layer of the communication stack—specifically within the layer 2 Packet Data Convergence Protocol (PDCP)—conceals higher-layer protocol data (IP addresses, TCP/UDP port numbers, etc.) that could otherwise leak user-related behaviors, such as applications in use, communication endpoints, and other sensitive details [69, 72, 73]. However, decades of research on statistical traffic analysis have shown that encryption alone is insufficient to fully conceal user activity [74, 75, 76, 58, 54, 77, 78, 79, 80, 55, 59, 56]. Indeed, despite encryption, patterns such as variations in traffic timing, packet sizes, and data volume, remain visible, thus offering actionable insights to an observer which can infer sensitive information, including the category of applications being used and user behaviors.

Table 3.1: Comparative Analysis of Technical Specifications between 4G LTE and 5G NR.

Technical Aspect	4G LTE	5G NR
SSB	Single SSB with fixed positioning	Multiple SSBs utilizing beam sweeping
SCS	Fixed subcarrier spacing	Configurable and variable subcarrier spacing
PDCCH	Basic scheduling with static BW	Adaptive scheduling influenced by dynamic BWP
Coding Schemes for PDCCH and PDSCH	Convolutional Codes and Turbo Codes	Polar Codes and LDPC Codes

Second, statistical traffic analysis in 5G has often been considered impractical due to significant technological advancements over 4G/LTE radio specifications. As summarized in Table 3.1, these advancements include improvements in synchronization signal blocks, subcarrier spacing, control channel scheduling and coding schemes. These notable technical steps forward have come along with the perception that *sniffing* traffic generated by a *third party user* over the 5G NR air interface might be exceedingly challenging. This is primarily caused by the dependence of the decoding process on the the RNTI assigned to the receiver, which is sent to the user device over encrypted channels. As a result, a third party eavesdropper can no longer easily access it. This contrasts with 4G/LTE, where the explicit availability of the RNTI simplifies the sniffing process, as clearly demonstrated by a substantial body of research [44, 81, 41].

However, as shown in Chapter 2, sniffing 5G traffic is no longer prohibitive. In light of technical advances in sniffing capabilities, the objective of this chapter is to investigate the potential risks associated with traffic classification by an OTA eavesdropper in 5G NR cells.

3.2 Related Work

This section presents the state of the art in Network Traffic Classification (NTC), organized by wireless network technology. Table 3.2 summarizes the key works discussed.

Wi-Fi Networks: Wi-Fi networks, offering easy traffic capture at Access Points (APs), have attracted various NTC approaches. Rao et al. [83] introduced *Meddle*, using Virtual Private Network (VPN) tunnels, user actions simulated with Monkeyrunner [100], and traffic recorded by *tcpdump* [101]. They applied Secure Sockets Layer (SSL) interception [102] to decrypt flows, classifying over 92% of iOS and Android traffic, though without learning models. Exploiting Transmission Control Protocol (TCP) flow features (i.e., packet length, cumulative bytes, timing), Al-Naymat et al. [84] distinguished Voice over IP (VoIP) from non-VoIP apps using a Random Forest (RF) classifier, achieving more than 96% of accuracy. Similarly, Taylor et al. [75] achieved more than 99% of accuracy employing Support Vector Machine (SVM)/RF on Android TCP flows, and 65.5% accuracy using side-channel data. Alan et al. [85] applied multinomial and Gaussian naïve Bayes to launch-time traffic from $1,595$ Android apps, reaching 88%. Lopez-Martin et al. [86] combined CNNs and RNNs on TCP/User Data Protocol (UDP) flows from *RedIRIS* [103], surpassing 96% accuracy. Qazi et

Table 3.2: Comparative summary of related work on mobile network traffic classification.

Work	Technology	Attacker's Position[1]	Data Source[2]	PDCCH	PDSCH	Analysis Method
Other Networks and Public Datasets						
Yao H. et al. [82]	N/A	Internal	HTTP flow	N/A	N/A	Statistical analysis
Wang W. et al. [78]	N/A	External	Public dataset	N/A	N/A	CNN
Lotfollahi M. et al. [54]	N/A	External	Public dataset	N/A	N/A	CNN
Iliyasu A. S. et al. [77]	N/A	External	Public dataset	N/A	N/A	DCGAN
Aceto G. et al. [79]	N/A	Internal/External	TCP, UDP flow	N/A	N/A	CNN
Miller S. et al. [80]	Ethernet	External	TCP flow	N/A	N/A	MLP
Wi-Fi Networks						
Rao A. et al. [83]	Wi-Fi	Internal	HTTP flow	N/A	N/A	Statistical analysis
Al-Naymat G. et al. [84]	Wi-Fi	External	TCP flow	N/A	N/A	RF
Taylor V. et al. [75]	Wi-Fi	External	TCP flow	N/A	N/A	SVM, RF
Alan H. et al. [85]	Wi-Fi	Internal	TCP flow	N/A	N/A	Multinomial/Gaussian naïve Bayes
Lopez-Martin M. et al. [86]	Wi-Fi	Internal	TCP, UDP flow	N/A	N/A	CNN, RNN
Qazi Z. et al. [87]	Wi-Fi	Internal	Traffic logs	N/A	N/A	Decision trees
3G/4G Networks						
Mongkolluksamee S. et al. [88]	3G	Internal	TCP, UDP flow	N/A	N/A	RF
Trinh H. D. et al. [68]	4G	External	IQ samples	●	○	SVM, LR, k-NN, MLP, CNN, RNN
Wang J. et al. [89]	4G	Internal	Traffic logs	○	○	LSTM + AE
Feng J. et al. [90]	3G/4G	Internal	Traffic logs	○	○	RNN
Wang X. et al. [91]	4G	Internal	Traffic logs	○	○	GNN
Zhang C. et al. [92]	3G/4G	Internal	Public dataset	○	○	CNN+LSTM
5G Networks						
Fei X. et al. [93]	5G NSA	Internal	TCP, UDP flow	●	●	LGBM
Yoon G. et al. [94]	5G SA	External	IQ samples	●	○	CNN
Wu X. et al. [95]	5G	Internal	TCP, UDP flow	○	○	CNN
Astrakhantsev A. A. et al. [96]	5G	Internal	Public dataset	○	○	RF, ANN, k-NN, AdaBoost
Pell R. et al. [97]	5G SA	External	Traffic logs	○	○	k-NN, SVM, RF
Fan L. et al. [98]	5G	Internal	Traffic logs	○	○	Decision tree, RF, AdaBoost
Islam M. R. et al. [99]	5G	External	Traffic logs	○	○	RF, Extra Trees
Our Work	**5G SA**	**External**	**IQ samples**	●	●	**CNN, RNN, TCN**

[1] Specifies whether the attacker is internal (e.g., a compromised network entity) or external (e.g., an OTA attacker outside the network).

[2] Refers to the origin of the data used for traffic analysis.

[3] Indicates the physical layer channels investigated in the study: ●: Analyzed, ○: Not analyzed.

al. [87] analyzed logs of 40 popular apps within an SDN, using decision trees and achieving 94%.

3G and 4G Networks: 3G and 4G environments pose greater data capture challenges, yet NTC remains feasible. Mongkolluksamee et al. [88] used RF on 3G TCP/UDP flows, achieving an F-measure of 0.95. Trinh et al. [68] leveraged OWL tool [41] to decode DCI from the PDCCH in 4G LTE, comparing SVM, Linear Regressor (LR), k-Nearest Neighbors (k-NN), Multi-layer Perception (MLP), CNN, and RNN, with CNN reaching 98% accuracy using MCS and TBS features. Wang et al. [89], using a large China Mobile dataset, integrated an Auto-Encoders (AE)-based model for spatial representation and Long Short-term Memory s (LSTMs) for temporal dynamics, achieving Mean Squared Error (MSE) under 0.15; Feng et al. [90], analyzing traffic from thousands of 3G/4G BSs, employed an RNN to reduce Normalized Root Mean Square Error (NRMSE) below 0.07. Zhang et al. [92], using dense urban cellular data from [104], combined CNNs and LSTMs, attaining NRMSE less than 0.19. Finally, Wang et al. [91], processing 4G BS data, used a Graph Neural Network (GNN) to surpass 79% accuracy.

5G Networks: The advent of 5G networks introduces new challenges and opportunities for NTC due to increased encryption and complex architectures. Fei et al. [93], examined real-time traffic classification in 5G NR Non Stand-Alone (NSA) scenarios using diverse app traffic datasets. By extracting features from both the PDCCH and PDSCH, they addressed encrypted flows and trained an Light Gradient Boosting Machine (LGBM) model, achieving 95% accuracy. Yoon et al. [94] analyzed 5G Stand-Alone (SA) networks by capturing mobile

traffic using *5GSniffer* [29] to decode DCI messages from the PDCCH. Extracted features included MCS, TBS, inter-DCI intervals, and transmission direction (uplink or downlink). A CNN model trained on these features achieved over 98% accuracy for individual app classification and over 99% for app categories. Wu et al. [95] proposed *CLPREM* for real-time traffic prediction in 5G networks. By collecting TCP and UDP flows from smartphones and utilizing a CNN, they optimized performance in terms of accuracy. Astrakhantsev et al. [96] addressed real-time traffic classification by comparing algorithms like Artificial Neural Networ (ANN), RF, k-NN, and AdaBoost using a labeled dataset [105], achieving accuracies above 82% across all models. Pell et al. [97] tested supervised learning techniques including k-NN, RF, and SVM for service classification using features from traffic metadata obtained from 5G CN logs. The RF classifier achieved the highest accuracy of 96.9%. Fan et al. [98] proposed a traffic fingerprinting method focusing on packet payload length transitions. Data from uplink traffic of energy metering terminals connected via 5G were used to train Decision Tree, RF, and AdaBoost classifiers. The RF classifier achieved the best precision of around 90%. Islam et al. [99] used QXDM [34] tool to extract 5G traffic logs and reveal the application's behavior. The RF and Extra trees classifiers reach an accuracy of 94% and 90%, respectively.

Other Networks and Public Datasets: Several studies utilize public datasets or are network-agnostic but contribute significantly to NTC. Yao et al. [82] proposed *SAMPLES*, a framework to identify mobile apps based on appID and Hypertext Transfer Protocol (HTTP) header information. They collected data from selected apps dealing in HTTP communications, building a repository of appIDs and names. Tested on 15 million flows from $700,000$ apps, it achieved accuracy higher than 72% but was limited to unencrypted HTTP traffic without learning models. Wang et al. [78] proposed a CNN model for classifying encrypted traffic using data from [76], achieving over 80% accuracy. Lotfollahi et al. [54] introduced a deep packet classification approach utilizing a CNN on the *ISCX* VPN-nonVPN dataset [76], achieving precision of 0.93 and recall of 0.94. Iliyasu et al. [77] presented a semi-supervised learning technique using a Deep Convolutional Generative Adversarial Networ (DCGAN), applied to the same dataset, achieving classification accuracies of 89%. Aceto et al. [79] investigated deep learning models for mobile traffic classification using network flows from *RedIRIS* [103], with a CNN achieving 85.70% accuracy. Miller et al. [80] differentiate TCP traffic transmitted over an encrypted *OpenVPN* channel from normal traffic. Using features extracted with *NetMate* [106] and an MLP, they achieved a precision of 92.9%.

3.3 Methodology

The primary objective of our study is to determine whether specific temporal traffic patterns exist for commonly used applications. We focused on five apps of two different categories: Amazon, eBay, and Shein, representing the "e-commerce" category; Netflix and Amazon Prime Video, representing the "video-streaming" category. To analyze the impact of the

scheduling algorithm in the identification of application-specific traffic patterns, we considered both SU and MU scenarios.

As shown in Figure 3.1, the proposed methodology is structured into three main stages: data collection, data processing, and classification. IQ samples were captured from a private 5G SA network in the first stage. Then, in the second stage, features were extracted and selected using *Golden Sniffer*, followed by pre-processing. Finally, in the classification stage, the processed features were used to train and test three deep learning models: CNN, RNN, and TCN.

CNNs were originally designed for image processing but have been adapted to handle one-dimensional sequential data, such as time-series or traffic patterns. They are highly effective in extracting localized patterns through convolutional operations, making them ideal for detecting short-term temporal dependencies or bursts of activity. By sliding convolutional filters across the input, CNNs learn spatially or temporally invariant features, enabling robust pattern detection within short temporal windows. These filters act as feature detectors, capturing localized dependencies which are critical for tasks involving brief or localized traffic bursts. Pooling layers, such as max-pooling or average-pooling, reduce data dimensionality by focusing on the most significant features, thereby improving computational efficiency while discarding redundant information. Although CNNs excel at identifying localized patterns with high computational efficiency and parallel processing capabilities, they are inherently spatially constrained. This restricts their ability to capture long-term dependencies or global patterns necessary for understanding complex behaviors spanning multiple timesteps. Despite this limitation, CNNs remain an excellent choice for tasks focusing on short-range temporal dependencies.

RNNs, on the other hand, are designed to maintain a memory of past inputs, making them particularly effective in capturing long-term temporal dependencies in sequential data. Unlike CNNs, RNNs process sequences step-by-step, enabling the network to model dependencies over extended time periods. This capability is essential for analyzing recurring patterns or user behaviors that unfold across interactions with multiple applications. However, RNNs face several challenges, including vanishing gradient problems, which can impair their performance when processing long sequences, and, due to their sequential processing nature, higher computational costs, and slower training times. It results that their scalability can be limited for real-time or large-scale traffic analysis.

TCNs address some of the limitations of both CNNs and RNNs, by combining the efficiency of convolutions with the ability to model long-range temporal dependencies. Using dilated convolutions, TCNs expand the receptive field exponentially with depth, allowing them to capture patterns over extended time periods without significantly increasing the number of parameters. This makes TCNs computationally efficient and suitable for parallel processing, unlike RNNs. Additionally, the use of causal convolutions ensures that predictions at any timestep depend only on past inputs, preserving the sequential nature of the data. While TCNs are highly effective in balancing short- and long-term dependency modeling, they re-

quire careful architectural design and tuning of dilation rates to perform optimally, which add complexity to their implementation.

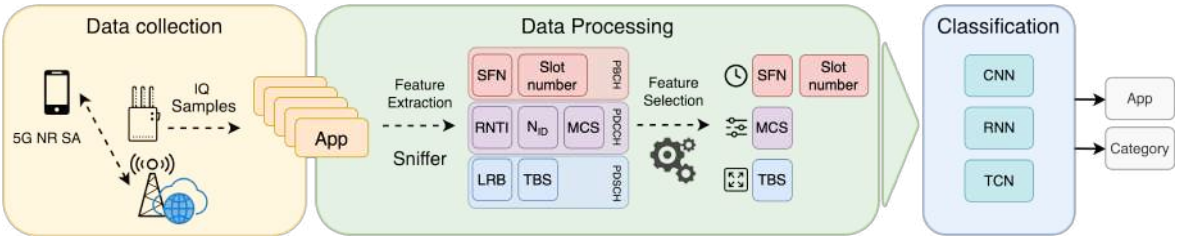


Figure 3.1: The workflow pipeline consists of three main stages: (i) data collection, where IQ samples are captured from a 5G NR SA network; (ii) data processing, involving feature extraction and selection using our sniffer; and (iii) classification, where extracted features are used to train and test deep learning models such as CNN, RNN, and TCN.

3.3.1 Experiemental Dataset Creation

The dataset creation process, depicted in Figure 3.1 as *data collection*, involved the following steps: (i) Attachment of a COTS UE to our private 5G SA network. (ii) Recording of IQ samples during a PDU session of a specific app. (iii) Processing of IQ samples for subsequent processing.

5G Testbed Implementation: The experimental setup for deploying the 5G SA network, as presented in Figure 3.2, comprised a workstation running Ubuntu 22.04, utilizing the open-source 5G software stacks of Open5Gs[50] and srsRAN_Project[49] for implementing the CN and the RAN of our private 5G Frequency Division Duplex (FDD) cell in the N28 band. srsRAN enables the implementation of 5G networks with realistic configurations. An Ettus Universal Software Radio Peripheral (USRP) N310[52] SDR was employed as the Radio Unit (RU), and a OnePlus 9 5G, Google Pixel 6a and Google Pixel 8a, Android smartphones, served as COTS UEs.

Trace acquisition: During the experiments, we ensured to use a single application at a time, that is the most common users' behavior. Since our sniffer is not yet optimized for real-time operation, we recorded IQ samples of the Radio Frequency transmissions between the BS and the COTS UE in sc16 file format. This was achieved using another SDR USRP N310 [52] and the `uhd_rx_samples_to_file` script from the USRP Hardware Driver (UHD) library.

We generated about 2400 sessions lasting 60 seconds, resulting in a total of 40 hours of traffic. For the SU scenario, 300 sessions per app were captured with only one UE connected to the 5G network, thereby maintaining a consistent RNTI, with the assumption that the UE was downloading data from a single app during each session. In contrast, for the e-commerce category in the MU scenario, three UEs were connected simultaneously: the target UE exclusively downloaded data from a specific app, while the other two UEs accessed apps at random. To minimize correlations between consecutive records, we enabled airplane mode, cleared caches, and powered down the base station every five sessions. Additionally,

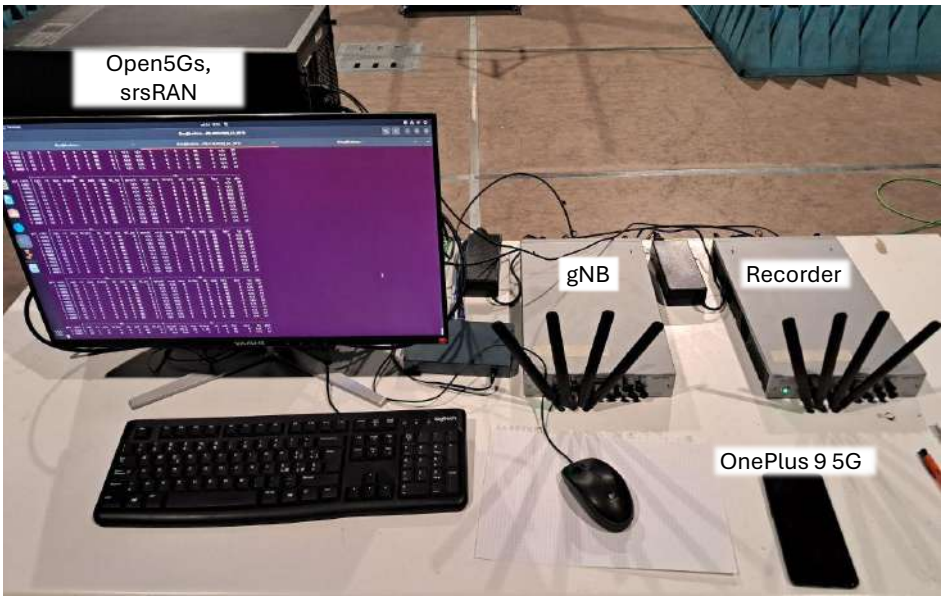


Figure 3.2: 5G Experimental Setup: Featuring an SDR-based srsRAN gNB (USRP N310), SDR sniffer (USRP N310), Open5GS CN, and the OnePlus 9 5G COTS UE.

we simulated various user actions and recorded traffic under both ideal and poor channel conditions, with the ideal case conducted inside a semi-anechoic chamber to filter out external interference.

3.3.2 Processing Pipeline

Data collection and pre-processing: As shown in the *data processing* phase of Figure 3.1, the collected IQ samples were processed using our *Golden Sniffer* to extract relevant features. The sniffer decoded the PBCH, PDCCH, and PDSCH. Decoding the PBCH provided the SFN and Slot Number, while the PDCCH yielded the RNTI, the n_{ID} of the cell, and the DCI bitstring. From the PDSCH, it is possible to obtain the LRB allocated to a user, thus enabling computation of the TBS as per [4]. Additionally, the encrypted payload of physical data within a slot was captured but not used.

Feature Selection and Analysis: To construct the dataset for training, validation, and testing of the deep learning classification models, six key features were initially selected: SFN, Slot Number, MCS, LRB, TBS, and RNTI. The SFN and Slot Number served as time references, while MCS, LRB, and TBS provided information about channel quality and the amount of downloaded data. As SFN and Slot Number were temporal indicators, and LRB and TBS were highly correlated [4], the correlations between MCS, LRB, and TBS were calculated to confirm, identify and remove redundant information. The computed correlation matrix is presented in Table 3.3. Based on this analysis, only SFN, Slot Number, MCS, and TBS, releted to the target RNTI, were retained for the final dataset, focusing only on non-redundant and critical features.

Data Processing: To recognize specific temporal traffic patterns, we initially worked by generating a temporal serie of feature vectors with a time granularity of 1ms (correspond-

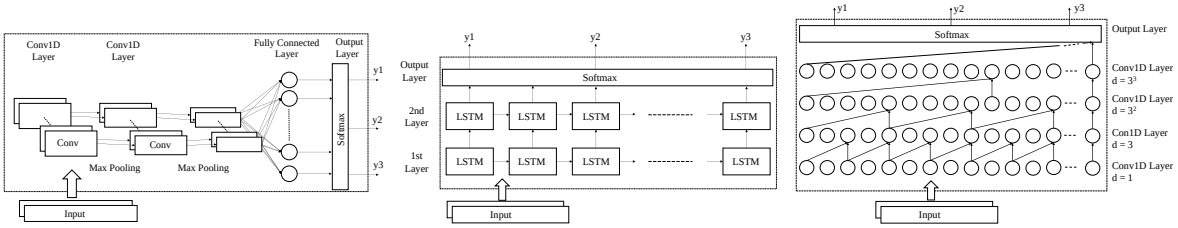
Table 3.3: Correlation matrix among MCS, LRB, and TBS features.

	MCS	LRB	TBS
MCS	1	0.86	0.8
LRB	0.86	1	0.91
TBS	0.8	0.91	1

ing to one sub-frame); when the sub-frame was not including allocations for the target user, we included a feature vector with null components. This approach introduced a significant number of zeros, particularly in e-commerce app data, which could negatively impact model training by causing issues such as class imbalance. To address this problem, we reduced the time granularity from 1ms to 10ms (equivalent to the duration of one frame), by averaging the values within each time interval. This frame aggregation not only reduced the prevalence of zero entries, but also minimized the impact of multi-user resource scheduling. While averaging proved effective in our tests, it has the potential to cause information loss in certain scenarios. For example, if only one slot within a frame is occupied, the mean value may no longer accurately reflect this detail. Although this was not an issue in our case, information loss can be mitigated by adding an additional feature specifying the number of non-empty slots per frame.

In order to classify applications, we organized the temporal serie of feature vectors into blocks of 500 samples (lasting 5 seconds, being each sample corresponding to one frame). Indeed, we assume that this time interval is reasonable for capturing user interactions with applications; for example, a user of a generic e-commerce app might perform an action every 5 seconds. We used the same block lenght for both e-commerce and video-streaming applications.

Finally, we divided each dataset into three parts: 20% was used for testing, and the remaining 80% was further divided into validation (20%) and training (80%) sets.



(a) CNN architecture (b) RNN architecture (c) TCN architecture

Figure 3.3: Comparison of deep learning model architectures: (a) CNN architecture, (b) RNN architecture, (c) TCN architecture.

Deep Learning Classification models: The classification phase, depicted in Figure 3.1, involved supervised learning using three deep learning models: CNN, RNN, and TCN. The experiments were performed on a Workstation running Linux Mint 22 Cinnamon with a Linux kernel version 6.8.0-49-generic. The workstation was equipped with an Intel Core i9-9900KF

CPU @3.60GHz (8 cores), 64GB of RAM, and an NVIDIA GeForce RTX 2080 Ti GPU (TU102). The models were implemented using TensorFlow v.2.17.0, a flexible and efficient framework for training neural networks. For all models, the input data were structured in the format $(n\_timesteps, n\_features)$, where $n\_timesteps$ represents the length of each data block (500 timesteps corresponding to 5 seconds), and $n\_features$ denotes the number of input features at each timestep, which in our case were the MCS and the TBS. The One-Dimensional CNN (1D-CNN) model, in Figure 3.3a, was designed with an architecture comprising two Conv1D layers, each with 128 filters with a kernel size of 5 and Rectified Linear Unit (ReLU) activation functions, followed by max-pooling layers to reduce dimensionality. A dense layer with 64 units was added before the output layer. The RNN model, in Figure 3.3b, consisted of two recurrent layers, each containing 64 units and utilizing ReLU activation functions. The TCN model, in Figure 3.3c, consisted of four Conv1D layers with dilations to allow the network to reach back in time to earlier inputs. Each Conv1D layer had 32 filters and a kernel size of 4, stacked sequentially with causal padding to preserve the temporal order of the data. A Global Average Pooling layer was incorporated to reduce the temporal dimension by averaging the activations across the entire sequence, resulting in a fixed-size representation for each sample. The output layer of all models was a fully connected layer with softmax activation to produce probability distributions over the output classes. The models were compiled using the Adam optimizer and trained using the categorical cross-entropy loss function. The evaluation metric for model performance was accuracy, and hyperparameter optimization was implemented using a grid search approach to systematically explore combinations of parameters.

3.4 Experimental Results

Our experiments aim to demonstrate the performance of our classification method for both individual applications within the same category and across different categories, in both SU and MU scenarios. To achieve this, we initially evaluated the performance of existing methods, specifically [94], using our SU dataset. Subsequently, we also addressed the MU scenario.

The method presented in [94] is applicable when using srsRAN_4G [107], or when considering a 4G LTE scenario, as in [68]. In these cases, the gNB (or eNB) allocates the entire bandwidth to a single UE. However, under real-world conditions, the FDA DCI field becomes ambiguous. Indeed decoding the data channel is essential for accurately interpreting it and computing the Transport Block Size (TBS), which is precisely what our method accomplishes. Consequently, in realistic conditions, the TBS feature is not available using the approach in [94].

Moreover, the utilization of the 5GSniffer [29], which captures DCIs at a ratio of 1:20 as stated in [94], presents two main limitations: firstly, it requires significantly more time to create the dataset, thus to perform classification, and secondly, it only provides a relative temporal reference between slots of collected DCIs. In contrast, as seen in Subsection 2.4.3,

our OTA 5G Sniffer captures over 93% of DCI events, allowing us to drastically reduce data acquisition time and provide an absolute time reference within the frame.

Using our dataset, we evaluated performance across four different scenarios, summarized in Table 3.4:

1. **Baseline (Case I):** Represents the state of the art in real-world conditions with no TBS feature, a 1:20 DCI capture rate (on average), and relative time (`REL_TIME`) between captured DCIs. A 1D-CNN model is used for classification.

2. **Improvement 1 (Case II):** Assumes the 5GSniffer [29] misses only 7% of DCIs (like *Golden Sniffer*), still excluding the TBS feature but including MCS and `REL_TIME`. Classification is performed using a 1D-CNN.

3. **Improvement 2 (Case III):** Builds on Case II by introducing the TBS feature while maintaining `REL_TIME`. A 1D-CNNs model is employed.

4. **Final Approach (Case IV):** Extends Case III by eliminating the `REL_TIME` feature and transforming the event-based data trace (where an event represents the DCI capture) into a temporal series of features sampled every 10ms. Classification is performed using 1D-CNN, RNN, and TCN models.

Table 3.4: Models and Feature Configurations Across Cases.

Case	Features	Model
I	MCS, REL_TIME	1D-CNN
II	MCS, REL_TIME	1D-CNN
III	MCS, TBS, REL_TIME	1D-CNN
IV	MCS, TBS, ABS_TIME	1D-CNN, RNN, TCN

3.4.1 Classification Performance

Case I and II achieved modest classification accuracies. Specifically, the accuracy for app category classification was below 68%, and individual app classification was approximately 60%. These outcomes underscore the limitations of excluding the TBS feature and relying on sparse DCI captures.

Case III aimed to enhance classification performance by incorporating the TBS feature while maintaining the relative timing `REL_TIME` feature. The inclusion of TBS provides additional context regarding the volume of data transmitted, which is crucial for distinguishing between different traffic types. This enhancement led to substantial improvements, with classification accuracy increasing by 30% for app categories, reaching approximately 91%, and by 11% for individual apps, achieving around 71%. This significant boost highlights the importance of incorporating TBS in traffic analysis.

Case IV introduced further enhancements by replacing the `REL_TIME` feature with absolute timing (`ABS_TIME`), which accounts for empty slots. This modification provides an

absolute temporal reference within each frame, enabling more accurate temporal pattern analysis. Additionally, in this case, we expand our model repertoire to include RNN and TCN alongside CNN.

The comprehensive approach in Case IV yielded remarkable classification performance:

- **App Category:** CNN, RNN, and TCN models attained accuracies of 94%, 97%, and 98%, respectively.
- **Individual App:** The same models achieved accuracies of 90%, 94%, and 97%, respectively.

These results demonstrate the effectiveness of using absolute timing and leveraging advanced neural network architectures for traffic classification. All case results are summarized in Table 3.5.

Table 3.5: Classification accuracy across different cases.

Case	App Categories (%)	Individual Apps (%)
I	66	60
II	68	60
III	91	71
IV	94, 97, 98	90, 94, 97

3.4.2 Comparative Analysis: Impact of Model Choice and Granularity

To validate our architectural and methodological choices, we performed a comprehensive comparison against traditional Machine Learning baselines (SVM, RF, MLP) and evaluated the impact of temporal granularity (1 ms vs. 10 ms) on classification performance.

Table 3.6 summarizes the results. First, we observe that Classical ML models consistently underperform compared to Deep Learning architectures. While temporal aggregation (10 ms) significantly aids statistical classifiers, improving RF accuracy from 41.0% to 67.0%, they fail to reach competitive levels. The best-performing baseline, MLP with aggregation, caps at 80.0%, confirming that statistical features alone are insufficient to fully capture the encrypted traffic patterns.

Second, the results highlight the inherent robustness of Deep Learning models. Unlike classical baselines, architectures like RNN and TCN maintain high accuracy even when fed with raw, noisy slot-level data (1 ms), achieving 91.0% and 88.0% respectively. This demonstrates their superior ability to extract features from high-frequency scheduling sequences.

However, the proposed 10 ms aggregation strategy proves critical for maximizing reliability. By filtering out scheduler-induced jitter, aggregation provides a consistent performance boost across all models, pushing the TCN accuracy from 88.0% to 97.0%. This conclusively proves that the optimal profiling framework requires both advanced sequential modeling to capture long-range dependencies and frame-level aggregation to maximize the signal-to-noise ratio.

Table 3.6: Comprehensive Performance Comparison: Impact of Model Architecture (ML vs. DL) and Temporal Granularity (1 ms vs. 10 ms) on Classification Accuracy.

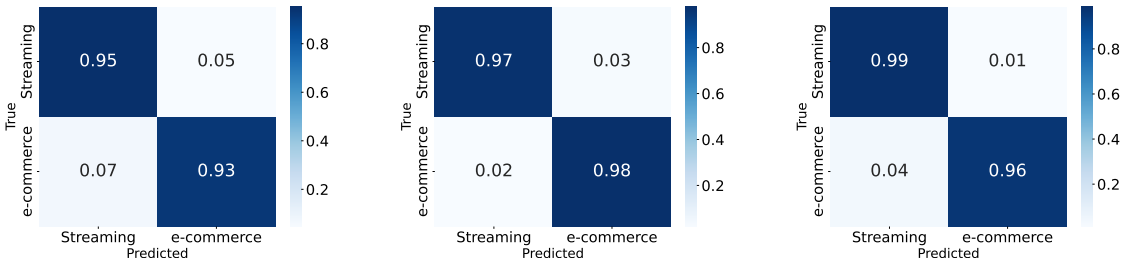
Model Type	**Algorithm**	**Individual App Accuracy (%)**	
		Raw (1 ms)	**Aggregated (10 ms)**
Classical ML	SVM	69.0	74.0
	RF	41.0	67.0
	MLP	73.0	80.0
Deep Learning	1D-CNN	82.0	90.0
	RNN	91.0	94.0
	TCN	88.0	97.0

3.4.3 Classification Analysis

In this subsection, we present the performance of our classification framework at two granularity levels: (i) app category classification (e-commerce vs. video-streaming), and (ii) individual app classification (Amazon, eBay, and Shein). We show confusion matrices derived from the final approach (Case IV) using the CNN, RNN, and TCN models, and we highlight how throughput analysis helps explain the observed classification performance.

App Category Classification: Figure 3.4 presents confusion matrices for classifying app categories (e-commerce and video-streaming) using CNN, RNN, and TCN models under Case IV, achieving classification accuracies of 94%, 97%, and 98%, respectively. Each model consistently achieves high precision and recall across both categories, indicating that the learned representations capture general patterns rather than overfitting to a single class. In particular, the darkest diagonal cells reflect an almost perfect classification of both e-commerce and video-streaming traffic. These results suggest that all three models effectively generalize to broader app categories, making them well-suited for real-world traffic classification tasks.

To better understand models' performance, Figure 3.5 compares the mean throughput of a typical "e-commerce" session and a "video-streaming" session (each lasting approximately 60 seconds). The mean throughput is computed by filtering the instantaneous throughput with a first-order AR filter (time constant = 1 s). Notably, the video-streaming trace clearly shows the gNB buffering mechanism: after an initial idle phase, the throughput ramps up during buffering and does not return to zero by the end of the acquisition. In contrast, e-commerce traffic remains at very low throughput for most of the session. However, in some time intervals (e.g. around 25s), the two throughput traces are comparable: this means that classification cannot simply work on short-term throughput measurements, while the overall pattern is easily recognizable. The distinct throughput patterns of the two app categories correlate with the high classification accuracy observed in Figure 3.4.



(a) CNN (b) RNN (c) TCN

Figure 3.4: Confusion Matrices for App Category Classification Using Different Neural Network Architectures under the Final Approach (Case IV) in SU Scenario. (a) CNN, (b) RNN, and (c) TCN models demonstrate the ability to distinguish between "e-commerce" and "video-streaming" traffic with high precision and recall.

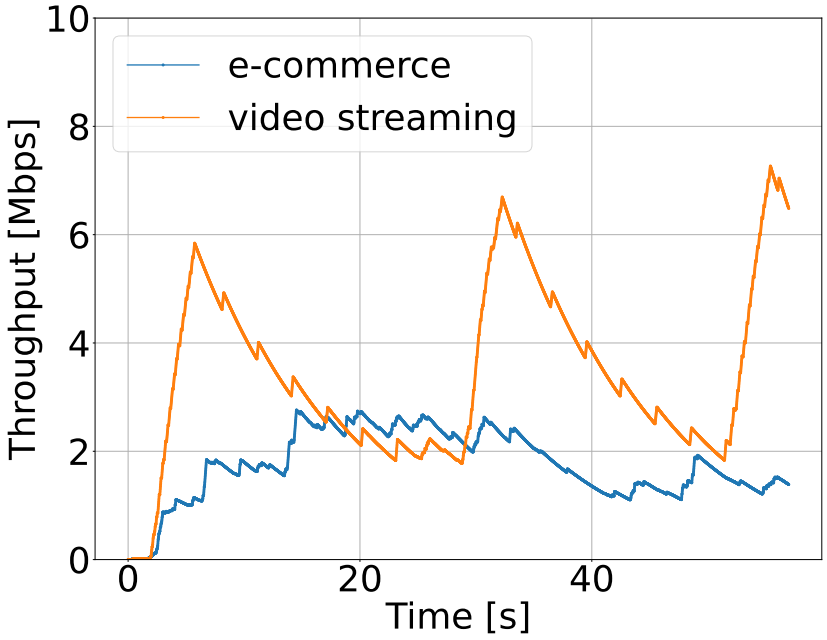
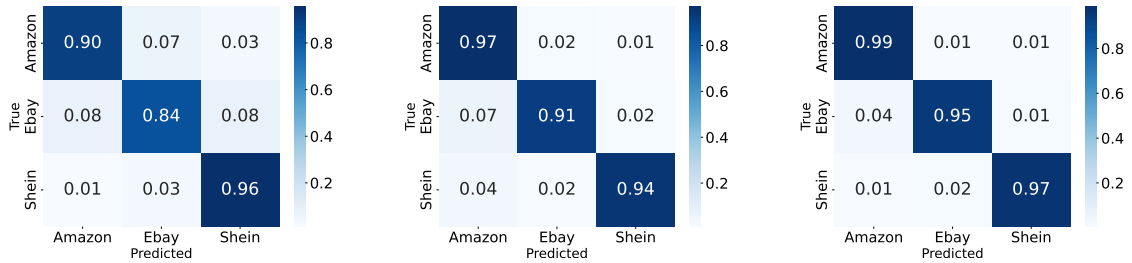


Figure 3.5: Mean throughput comparison for "e-commerce" and "video-streaming" sessions (both $\approx 60\,\mathrm{s}$). The throughput is filtered by a first-order AR filter (time constant = 1 s).

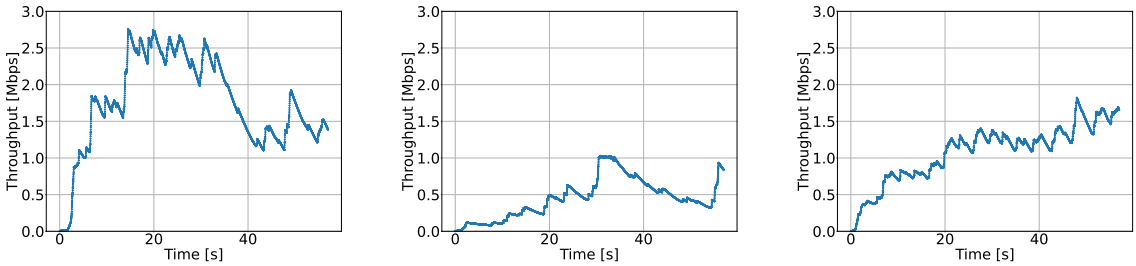
Individual App Classification: Figure 3.6 shows the confusion matrices for individual app classification (Amazon, eBay, and Shein) under Case IV using CNN, RNN, and TCN models, achieving 90%, 94%, and 97%, respectively. All three networks attain high accuracy, correctly identifying the vast majority of samples from each app. Notably, the TCN model exhibits the fewest off-diagonal entries, indicating fewer misclassifications. This suggests that temporal convolutional structures are particularly adept at capturing subtle, app-specific traffic patterns. Consequently, the TCN model stands out as the most effective choice for application-level traffic classification in dynamic, real-world scenarios. Also in this classification scernario, mean throughput of the three applications is quite different, allowing to the models to classify the apps' temporal pattern with an high accuracy. Figure 3.7 shows the throughput trends for Amazon, eBay, and Shein, computed using a first-order AR fil-

ter. Amazon and eBay exhibit similar temporal trends, though eBay's average throughput is closer to Shein's. These subtle differences provide sufficient signal for the models to achieve high accuracy, with the TCN model particularly excelling in recognizing these nuances.



(a) CNN (b) RNN (c) TCN

Figure 3.6: Confusion Matrices for Individual Application Classification under the Final Approach (Case IV) in SU Scenario. The matrices for (a) CNN, (b) RNN, and (c) TCN models illustrate performance in correctly identifying traffic patterns from individual apps (e.g., Amazon, eBay, and Shein).



(a) Amazon (b) Ebay (c) Shein

Figure 3.7: Mean value of throughput for three different "e-commerce" app. (a) Amazon, (b) Ebay, and (c) Shein. Values are obtained from istantaneous throughput values with a order-1 AR filter with time constant 1 second.

3.4.4 Multi-user scenario

In practical 5G deployments, the occurrence of a cell serving only a single active UE is highly uncommon. Consequently, we extended our evaluation to analyze a MU scenario. Initially, the performance of the previously trained models was assessed on the MU dataset. As evidenced by the confusion matrices in Figure 3.8, all models exhibited a significant degradation in performance. This degradation is primarily attributable to modifications in the BS scheduling algorithms, which in turn alter the temporal traffic patterns that the models were originally trained on. Figure 3.9 shows the difference of the resources assignment between both SU and MU scenario. In particular, the two plots represent the instantaneous resources assignment in 10 seconds of a Amazon session, in the two cases. In the first scenario, the figure shows immediate and steady resource allocation for each request. This is an evidence that delays or buffers do not exist: all user's requests are satisfied with no latency and therefore require no prioritization. In a MU scenario, however, the plot shows a

different TBS variation: there are both resources buffering and sudden bursts of allocation. This is due to the fact that there are many users in the cell, and their resources are managed according to their Quality of Service (QoS) requirements and priority levels of activity. For this reason, the resources of the user under study are available but delayed and downloaded in bulk when higher-priority activities are satisfied. To address this issue, we retrained the CNN, RNN, and TCN models using a comprehensive combined dataset that integrates SU and MU scheduling strategies. As illustrated in Figure 3.10, the retrained models achieved high classification accuracy during testing. These results clearly demonstrate that incorporating datasets reflecting diverse scheduling strategies facilitates robust and accurate application classification. Table 3.7 summarizes the performance metrics from both experimental setups.

Table 3.7: Performance Comparison of Classification Accuracy: SU Trained Models versus Retrained Models on a Combined Dataset adding MU data.

Model	SU Accuracy (%)	Retrained Accuracy (%)
CNN	47	90
RNN	33	88
TCN	40	93

(a) CNN (b) RNN (c) TCN

Figure 3.8: Confusion Matrices for Individual App Classification in a Realistic MU Scenario using SU Trained Models. The figures for (a) CNN, (b) RNN, and (c) TCN models reveal a substantial performance degradation when models trained under SU conditions are applied to MU data.

3.5 Countermeasure

The experimental results conclusively demonstrate that by integrating the TBS feature and leveraging absolute timing information, even when UEs operate with dynamic BWPs, we achieved a substantial increase in classification accuracy. The adoption of advanced deep learning models, particularly TCN, further enhanced performance, especially in individual app classification where temporal dependencies play a crucial role. The confusion matrices reveal that our models achieve high precision and recall across all categories and individual apps, indicating robust generalization capabilities. The TCN model, in particular, exhibits near-perfect classification for app categories and excels in distinguishing individual apps with

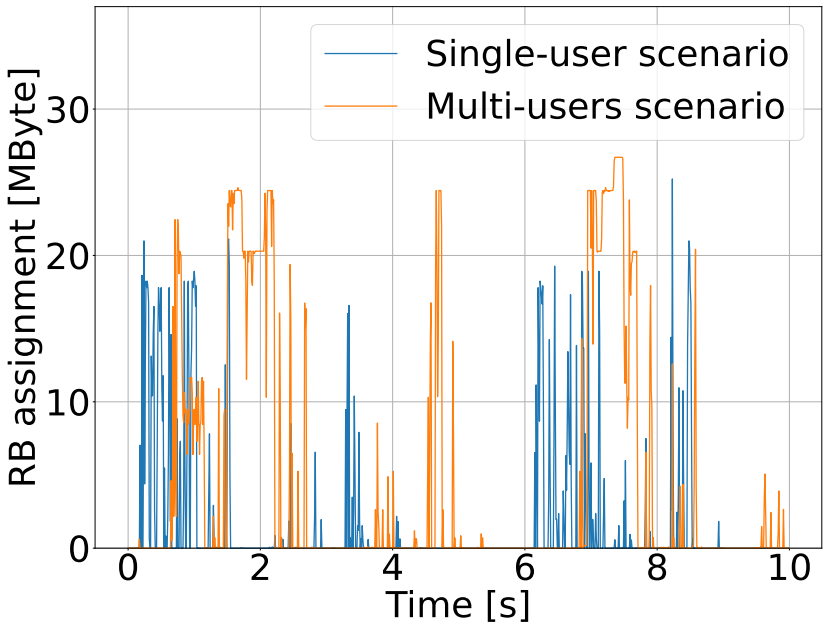
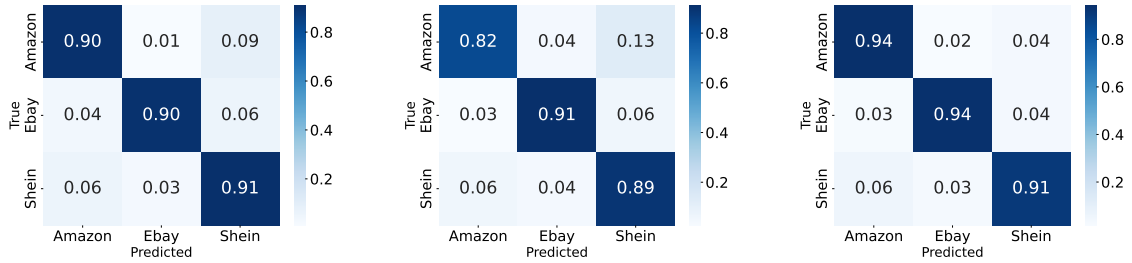


Figure 3.9: RB assignment over time for Amazon in SU and MU scenarios, illustrating steady allocation in SU and bursty, delayed allocation in MU due to shared scheduling.



(a) CNN (b) RNN (c) TCN

Figure 3.10: Confusion Matrices for Individual App Classification in a Realistic MU Scenario using Models Retrained on a Combined Dataset (SU and MU). Confusion matrices for (a) CNN, (b) RNN, and (c) TCN models indicate significant improvements in classification accuracy after retraining to incorporate diverse scheduling strategies.

minimal misclassifications. These findings highlight the critical importance of comprehensive feature extraction and the selection of appropriate neural network architectures in traffic flow confidentiality and classification tasks within 5G networks.

To address this issue, we propose a countermeasure based on strategically inserting zeros within the PDCCH DM-RS symbols in the frequency domain. This, emulating deep fading, disrupts blind DM-RS reconstruction explained in Subsection 2.3.2 (Figure 3.11), causing attacker PDCCH decoding to fail, whereas legitimate UEs, leveraging known scrambling parameters, still reconstruct the DM-RS and decode with limited performance impact. This can be achieved by modulating the amplitude of the frequency domain PDCCH REs at the gNB.

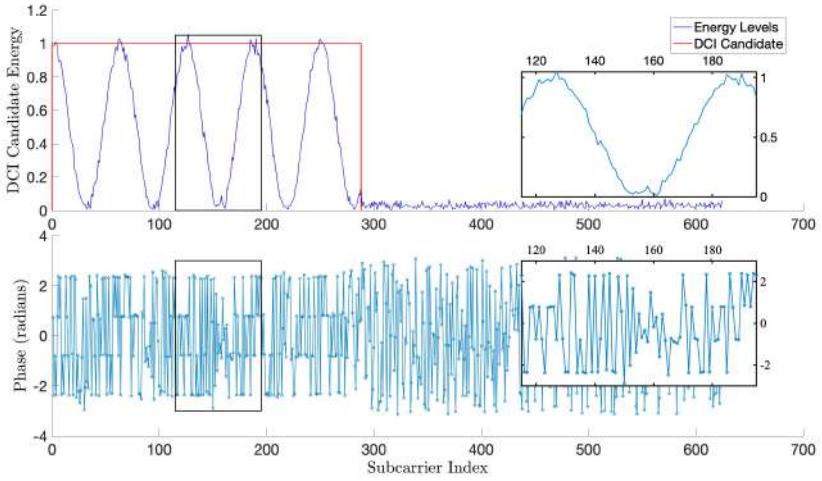


Figure 3.11: DCI candidates: (top) energy levels with a red-highlighted candidate; (bottom) IQ symbol phases revealing unpredictable phases at zero crossings.

3.5.1 Validation

We performed both MATLAB simulations and experimental validation with srsRAN and COTS UEs, to analyze the trade-off between security effectiveness and signal quality degradation. For each AL, we identified the optimal number of inserted zeros that render *Golden Sniffer* ineffective (that normally achieves over 93% DCI-decoding rate, Subsection 2.4.3) and the associated SNR penalty for a 40-bit DCI, yielding the minimal-penalty configurations shown in Table 3.8. The results show that lower ALs incur a higher penalty due to their higher coding rate, making the polar decoder more sensitive to inserted zeros. Nevertheless, legitimate UEs experience a limited impact on channel quality with up to 7 dB SNR degradation, while *Golden Sniffer* becomes ineffective. This embodies a standard security-reliability trade-off and demonstrates the practical feasibility of enhancing privacy without substantial degradation of network performance.

Furthermore, the approach is robust against MIMO attackers. Since the countermeasure targets the PDCCH, which is transmitted isotropically, a multi-antenna attacker receives the same artificially faded signal on all antennas and cannot use spatial processing to undo the effect.

Table 3.8: Minimal-penalty zero-insertion configurations per AL and DCI duration, yielding a 100% sniffer-failure rate while legitimate UEs keep decoding.

AL	DCI Duration 1		DCI Duration 2		DCI Duration 3	
	# Zeros	Penalty [dB]	# Zeros	Penalty [dB]	# Zeros	Penalty [dB]
1	9	6	3	7*	2	6*
2	9	3	2	4	1	4*
4	5	2	9	2	2	2
8	9	1	5	2	3	1
16	18	1	9	1	6	1

* result in DCIs with less than 16 DM-RS symbols, immune to our sniffing approach.

3.6 Conclusion

Traffic flow confidentiality, namely the ability to prevent an adversary from exploiting observable patterns in encrypted traffic timing and volume for revealing sensitive user-behavioral information, has historically received limited attention in cellular systems, likely because intercepting 5G traffic over the air was long assumed to be prohibitively difficult. However, with recent advancements in 5G New Radio sniffing technologies, we believe it is crucial to revisit solutions for ensuring traffic flow confidentiality over the air interface.

Findings confirm that deep learning techniques can effectively infer application behavior patterns from encrypted 5G communications. Experimental results show highly accurate classification performance, achieving up to 98% accuracy in distinguishing application categories, reaching 97% accuracy in identifying specific applications, and—after targeted re-training—up to 93% accuracy in more complex MU scenarios. These outcomes highlight that encryption alone is insufficient to obscure sensitive patterns and underscore the urgent need to strengthen traffic flow confidentiality.

Having established the feasibility of passive PHY-layer inference attacks, the next chapter moves one step further by examining *active* adversarial strategies. Specifically, we leverage a frame-synchronized attacker architecture to design a reactive jammer capable of targeting ongoing transmissions with symbol-level precision. This transition from passive observation to active disruption highlights how the same reverse-engineering capabilities that enable traffic analysis can also be repurposed to compromise availability, thereby exposing additional vulnerabilities in 5G's PHY-layer design.

Chapter 4

Hijacking the Beam: A Selective Jamming Attack on Channel State Information

This chapter shows a novel downgrade attack for 5G networks able to dramatically reduce the efficiency of MIMO links, while remaining virtually invisible to standard intrusion monitors. By opportunistically injecting noise precisely aligned with silent reference signals in the radio frames (ZP-CSI-RS), we show that an attacker can corrupt the feedback mechanism implemented at the UE for channel estimation, forcing the BS to downgrade or disable spatial multiplexing. This mechanism is much simpler than pilot spoofing schemes and dramatically harder to defend: indeed, silent reference signals cannot be protected by integrity mechanisms and their sparsity keeps the average power of the injected noise difficult to detect.

We design a robust mechanism for synchronizing the attacker transmissions to the silent reference signals, and quantify the latency of the attack under different SNR and cell load conditions. The approach has been implemented in a SDR testbed and experimentally validated in a private network, by demonstrating a success probability higher than 82% on two different commercial smartphones and a throughput degradation up to 70%.

4.1 Introduction

5G networks significantly enhanced the capabilities of mobile communication systems, providing increased throughput, reduced latency, and support for massive device connectivity. The high throughput capabilities of 5G are largely achieved through advanced communication techniques like MIMO [108, 109, 110], as already discussed in Subsection 1.2.4. However, the complexity of these PHY-layer mechanisms introduces new attack surfaces that can be exploited by malicious actors [111, 112]. The efficiency of MIMO in downlink transmissions heavily relies on accurate Channel State Information (CSI) feedback sent from the UE to the gNB for adapting the transmission strategy to the current radio conditions. This feedback is based on two key estimates at the UE: the number of spatial layers that can support

parallel transmissions and the interference level. Special reference signals are periodically transmitted in the radio frames for enabling these estimates: NZP-CSI-RS, which serve as pilot signals for estimating the channel matrix; ZP-CSI-RS, which deliberately leave certain resource elements empty to allow the UE to measure interference and noise [2, 4, 5, 6].

Despite of their potential benefits, it has been demonstrated that MIMO systems have several vulnerabilities: Singular Value Decomposition precoding and power allocation assume that the transmitter and/or receiver rely on accurate estimates of the channel matrix $\mathbf{H}$ (or its subspaces). In OFDM systems, $\mathbf{H}$ varies not only by time but also by subcarrier. When the estimates are wrong, the parallelization breaks down and capacity can be severely degraded.

Previous literature [113, 114] shows that an *intelligent* adversary corrupting channel estimates can be far more damaging than wideband noise jammers. Classic attack primitives include: (i) *pilot/sounding jamming* that corrupts the estimated $\mathbf{H}$ [115, 116, 117], (ii) *feedback manipulation* that induces the transmitter to precode along wrong singular vectors [118], and (iii) model-specific tactics such as "opposite water-filling" (mis-allocating power toward weak modes) or rank-reduction via structured contamination during training [119, 120]. These ideas have been analyzed and demonstrated by simulations and SDR-based experiments. Importantly, prior jamming literature that treats the adversary as *unaware* of protocol details (broadband/white-noise jamming) underestimates the real risk; protocol-aware, synchronized attackers targeting channel state estimates can achieve larger degradation for less power.

In this chapter, a novel attack exploits ZP-CSI-RSs to force a MIMO downgrade. The structured and periodic nature of CSI-RS transmissions in 5G (introduced in Subsection 1.2.4), particularly the mapping of NZP and ZP reference signals, enables both the pilot and feedback manipulation attacks. It is enough that an attacker is synchronized to the network and aware of the configuration of these signals, for potentially performing these attacks. In particular, the proposed attack is a very simple feedback manipulation attack. Rather than hacking the feedback message, the goal is corrupting the feedback estimation process by injecting noise on the ZP-CSI-RS. By selectively jamming the ZP-CSI-RS resources, an attacker can artificially inflate the perceived interference at the UE. This causes the UE to report a lower channel quality and, most importantly, a lower rank indicator, forcing the gNB to reduce the number of transmission layers. An attacker may have two objectives for pursuing a downgrade: it can act as a Denial of Service (DoS) attack by degrading the target's throughput, and more critically, it simplifies the complex multi-layer signal into a single-layer transmission that is significantly easier to intercept and decode.

To the best of our knowledge, this attack has not been addressed in previous studies on 5G MIMO vulnerabilities, which have largely concentrated on attacking active reference signals such as pilot jamming or spoofing [121, 122, 123, 124]. Attacks on active pilots are usually complex, because the limited duration of pilot signals, combined with spatial filtering in beamformed systems, impose stringent requirements on the adversary such as high transmission power, precise time and frequency synchronization, and accurate beam alignment to the

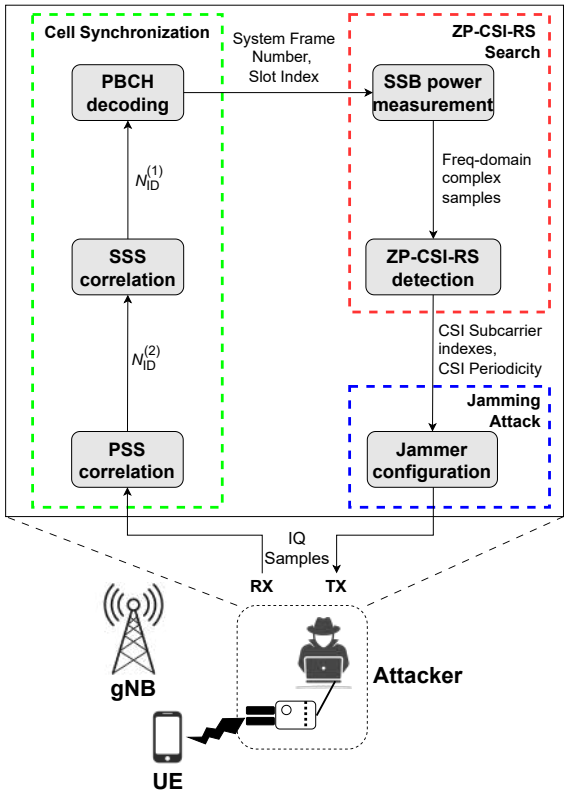


Figure 4.1: Workflow summarizing the adversary operations, including cell synchronization, ZP-CSI-RS detection, and jammer configuration.

UE [125]. Conversely, our attack is much more effective: the adversary must first identify the positions of ZP-CSI-RS in the time-frequency grid, which only requires cell synchronization and channel monitoring, and then maintain time and frequency synchronization with the gNB OFDM grid to ensure that interference is injected in the desired resource elements. Moreover, the attack is stealthy and low-power, as it does not interfere with active transmissions or require beam alignment. It is also robust: the network cannot distinguish between legitimate environmental interference and targeted manipulation and small synchronization errors in time only slightly reduce its effectiveness. A summary of the adversary operations (detailing cell synchronization, ZP-CSI-RS detection and jammer alignment) is provided in Figure 4.1.

4.2 Related Work

Massive MIMO in 5G networks has an intrisic security against passive eavesdropping, but also potential vulnerabilities in case of active eavesdroppers and jammers [121].

Spoofing attacks represent a well known vulnerability for CSI estimation in MIMO systems. The idea is adding synchronized transmissions at legitimate reference signals, leading to incorrect channel estimates [122]. In [123], the authors implement and evaluate two attacks: a CSI-based sniffing attack that allows malicious clients to eavesdrop on concurrent transmissions, and a power manipulation attack that enhances the attacker's capacity at the expense of other users. Even the very high directionality of the channel does not prevent a

well-timed pilot replica from poisoning CSI [126]. Mitigation strategies include secure feedback mechanisms or a beamforming-based validation mechanism for CSI estimates [124].

Jamming attacks. Another problem is related to *pilot contamination* due to interference generated by nearby cells or jammers. These attackes can substantially reduce the overall spectral efficiency of massive MU-MIMO systems [122], but can also be mitigated by opportunistic detection strategies. The susceptibility to interference in 5G NR-based MIMO links is studied in [127], where the authors analyze performance degradation under smart jamming scenarios. By evaluating Alamouti-OSTBC schemes in the presence of interference, the study quantifies the BER–SIR trade-off and mathematically assesses spectrum impairment in dense deployments. In [128], a detection mechanism is introduced, leveraging discrepancies between two channel estimators (LS and MMSE). Another approach proposed in [129] exploits pilot-partitioning mechanisms and channel correlation to reconstruct CSI. Another potential problem analyzed in in [130] is due to CSI-RS aging and feedback compression, which can be mitigated with predictive models using Kalman filters.

Most of the previously mentioned attacks have been demonstrated with analytical models or simulations because of the practical complexity of their implementation.

4.3 Jamming attack

In order to detect the ZP-CSI-RS-bearing symbols, the method proposed in Sub-subsection 2.3.4.2 is adopted. Once the ZP-CSI-RS periodicity and pattern is detected, the attack is designed to inject interference only on those specific time–frequency REs, without interfering with other signals.

In order to successfully carry out the attack, the signal transmitted by the attacker must be aligned with the signal received from the gNB, at the UE. This puts some constraint on the scenario, specifically that the attacker must be able to get an estimate of the relative shift between the signal received by the gNB and the signal it will transmit, from the point of view of the victim(s).

We implemented our jammer based on the open source srsRAN-UE[107], tweaking the TA parameter. Commercial handsets set this parameter initially during the Random Access procedure, and then update it based on feedback from the gNB, with the objective that their transmitted signal is received within the correct receive interval at the gNB. Conversely, in a SDR setup, the TA must compensate for a platform specific delay. We empirically found that a TA of 230μs reliably compensates the processing pipeline of the SDR used in our setup, an Ettus USRP B210 [131].

In our experimental evaluation with low-power devices, the gNB, attacker and victim nodes were physically close to each other, so that propagation delays are negligible, and the TA serves almost exclusively to counteract the latency introduced by the radio hardware itself. Small errors in setting this parameter do not affect the method, as long as the received signal is aligned within the size of the cyclic prefix, acting as a guard interval in OFDM.

OFDM-based transmissions, especially SDR-based ones, must also maintain proper time and frequency alignment. The maximum tolerable frequency error is on the order of $1/100$th of the subcarrier spacing, and larger errors severely affect the orthogonality between the subcarriers. The fine frequency alignment with the gNB is maintained by every device by monitoring the Synchronization Signals (SS) in the frequency domain. Small frequency errors appear as a constant phase shift between the PSS and the SSS. Precise time alignment can also be maintained by monitoring the phase of the received PSS and the SSS, as a time drift appears as a linear phase shift across the bandwidth.

Unlike other jamming attacks, the discontinuous operation of the proposed jammer allows keeping the synchronization with the cell. Indeed, our solution only transmits a very time-selective signal with a very low duty cycle equal to the reciprocal of the detected periodicity, so that tracking of the synchronization signals remains feasible and practical. Conversely, a jammer attacking the synchronization signals of a cell would not be able to keep time-alignment with the cell itself, as its clock would drift and there would be no straightforward solution to keep tracking the broadcast signals and maintain synchronization.

4.3.1 SDR-based Setup

On a SDR-based setup, the crystal oscillator error must have an initial value low enough for the UE to get initial synchronization with the gNB. External reference signals or a calibration method can be used to this purpose.

In this subsection, we present our error measurement strategy for a free-running SDR platform by comparison with a Global Positioning System Disciplined Oscillator (GPSDO)-locked device. We consider SDRs where the oscillators used to synthesize the mixer Local Oscillator (LO) frequency and ADC sampling rate may be different: in particular, we denote by δ_t and δ_f the sampling rate and LO frequency relative errors, respectively.

We consider two possible setups. The first one employs the GPSDO-locked SDR as transmitter (signal generator) and the free-running SDR as measurement device, while the second switches the roles, i.e. the free-running SDR transmits a tone at nominal frequency and the GPSDO-locked SDR measures the actual received frequency.

In the first setup, shown in Figure 4.2a, a sinusoid at frequency f_0 from a GSPDO-locked device is received by a free-running SDR. We denote by f_0' the apparent incoming frequency at the receiver, and f_c its nominal LO frequency. Since the free-running oscillator is affected by a relative frequency error δ_f, the receiver LO is actually $f_c' = (1+\delta_f)f_c$ and the incoming signal at frequency f_0 is mixed to $f_0 - (1+\delta_f)f_c$. This frequency is further mixed with a digitally generated complex exponential of nominal frequency $f_c - f_0$. Due to the sampling rate relative error, the actual mixing frequency gets multiplied by $1+\delta_t$, so that the frequency at the input of the spectral estimation block is $-f_0\delta_t + f_c(\delta_t - \delta_f)$. The spectral analysis block is itself affected by the sampling rate error itself, so that the measured frequency error results

$$f_0' - f_0 = \frac{-f_0\delta_t + (\delta_t - \delta_f)f_c}{1+\delta_t}. \tag{4.1}$$

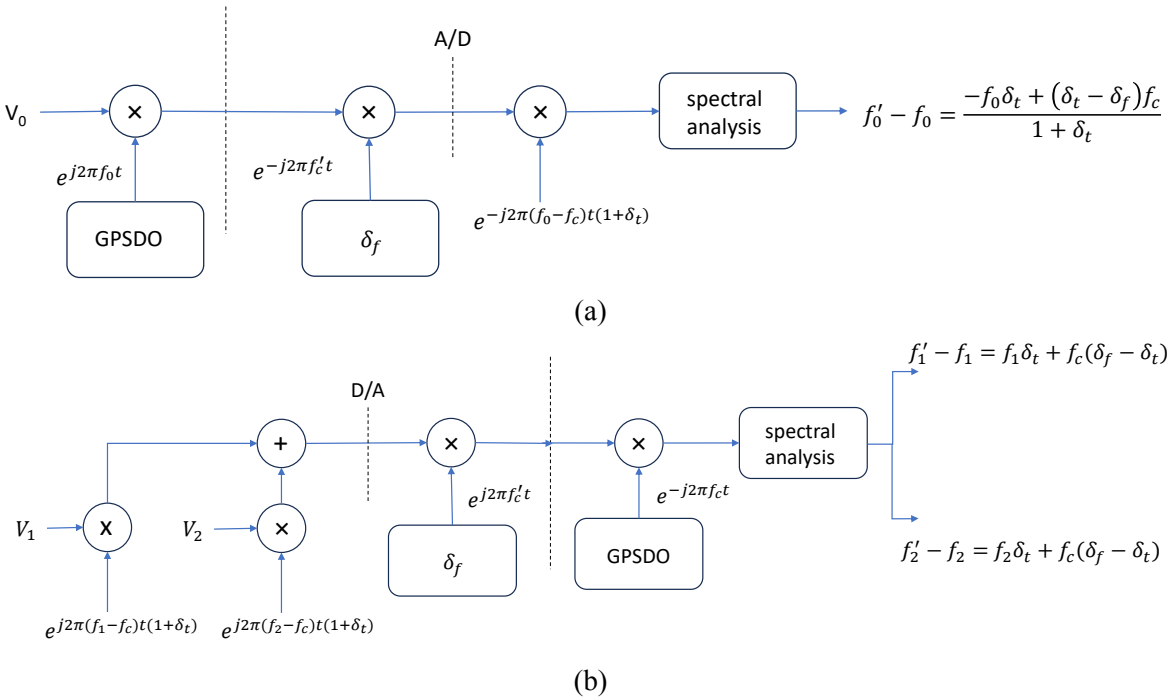


Figure 4.2: Signal flow graph for the apparent frequency errors for two tones measured at: (a) free-running SDR with a GPSDO-locked transmitter, (b) GPSDO-locked receiver from a free-running SDR transmitter.

In the second setup, shown in Figure 4.2b, the free-running SDR transmits a tone with nominal frequency f_1 and the GPSDO-locked SDR measures the actual received frequency f_1'. In this case, we obtain:

$$f_1' = f_c(1+\delta_f) + (f_1 - f_c)(1+\delta_t). \tag{4.2}$$

Note that in both equations (4.1) and (4.2), the unknowns δ_f, δ_t and the implementation dependent LO frequency f_c appear. In order to cancel the dependency from f_c, we add another tone at the nominal frequency f_2, and measure the actual received frequency f_2', modeled as follows:

$$f_2' = f_c(1+\delta_f) + (f_2 - f_c)(1+\delta_t). \tag{4.3}$$

Observe that, taking the difference between (4.3) and (4.2), we obtain

$$f_2' - f_1' = (f_2 - f_1)(1+\delta_t), \tag{4.4}$$

which leads to the estimate of the relative sampling frequency error δ_t

$$\hat{\delta}_t = \frac{f_2' - f_1'}{f_2 - f_1} - 1 = \frac{(f_2' - f_2) - (f_1' - f_1)}{f_2 - f_1}. \tag{4.5}$$

Note that (4.5) allows the estimation of δ_t directly, without knowledge of δ_f and/or f_c. In

Figure 4.3: GPSDO-locked N210 and free-running B210 radio used in the experimental setup.

order to maximize the measurement accuracy, the largest possible value for $f_2 - f_1$ (the highest supported sampling rate) should be used. Once an estimate of δ_t is available, it can be substituted in any of (4.1), (4.2) or (4.3). In turn, if f_c is known, an estimate of δ_f may be obtained. In many cases the LO frequency and the sampling rates are derived from the same crystal oscillator and $\delta_f = \delta_t$ so it is not necessary to estimate δ_f or make assumptions about f_c.

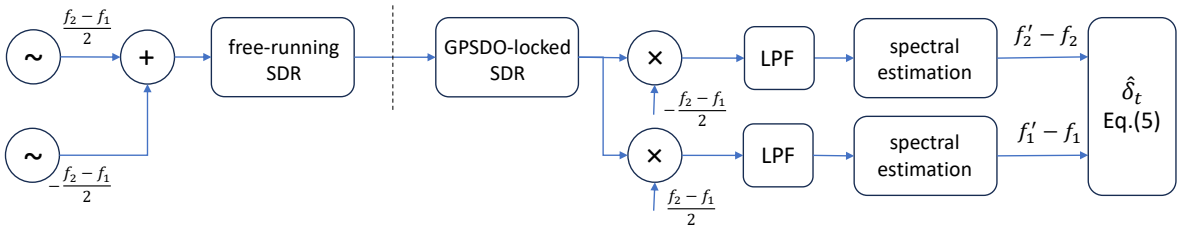


Figure 4.4: Flowgraph of the experimental setup for the measurement of the relative error according to equation (4.5).

Hardware. The hardware employed for the experimental validation of this study is constituted by a PC host, running Ubuntu 22.04.3 LTS 64-bit, a GPSDO-locked SDR (specifically, an Ettus Research N210 with the internal GPSDO option), and several free-running SDR's such as the one shown in Figure 4.3. In particular, we measured the crystal oscillator error of four different USRP B210 devices, which we intend to adopt for a 5G wireless cellular network testbed deployment based on srsRAN[49].

Software. In order to validate experimentally the proposed error estimation method, we used the GNURadio framework [132], which provides a collection of predefined signal processing blocks that can be combined and configured to implement the scenario shown in Figure 4.4. In particular, the proposed scheme calculates the maximum amplitude of the FFT of the difference signal to obtain an estimate of the sampling frequency error as described in Equation 4.5. The relevant parameters used in the realization of the flowgraph are reported in Table 4.1. The sampling rate, 25M-SPS, is the maximum supported by the N210. The fre-

quency span, low pass filters cutoff and stopband frequencies have been chosen as a tradeoff between frequency span and filter order. The tuning frequency falls in the band n2 [133]. We operated in the n2 band as we found that no mobile network operator was active, and setting the transmit power as low as possible for a successful experiment. The decimation rate was chosen in order to be able to cope with relative frequency errors of 4.006ppm at 1950MHz, which is twice the expected maximum relative error between two free-running devices with $\pm$2ppm *a priori* stability [131]. The FFT frequency bin size, with smoothing, is 0.95 Hz, and results in a ppm estimate resolution of approximately 0.5 ppb.

Table 4.1: Parameters for the signal flow graph of figure 4.4.

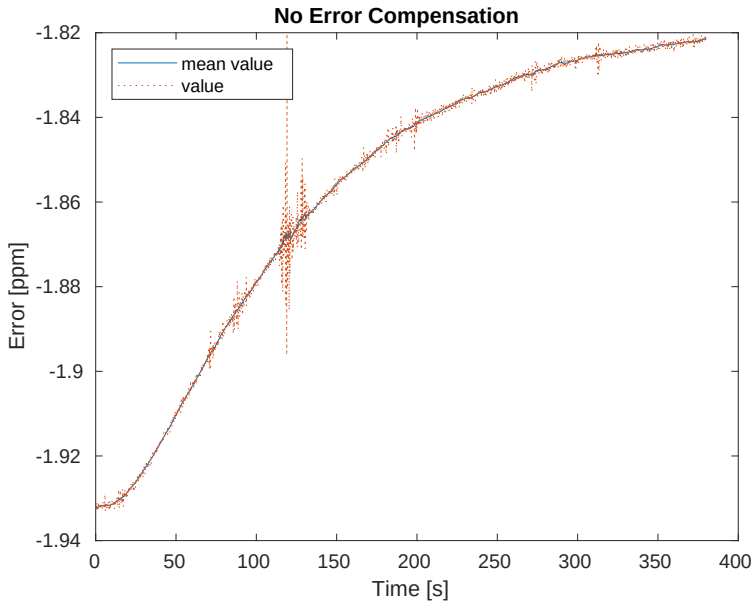
Sampling rate	25 MSps
Tuning frequency	1950 MHz
Frequency span $f_2 - f_1$	20 MHz
LPF stopband frequency	5MHz
LPF cutoff frequency	2.5MHz
Decimation rate	1600
FFT size	2048
Smoothing factor	8

Error compensation. The Ettus USRP B210 platform we consider in this work allows for almost arbitrary setting of the LO frequency and sampling rates. In fact, it is based on the Analog Devices AD9361 [134], featuring fractional-N frequency synthesis with a large modulus, resulting in a LO and sampling rate resolution of 1ppb (the datasheet reports 2.4Hz at a LO frequency of 2.4GHz).

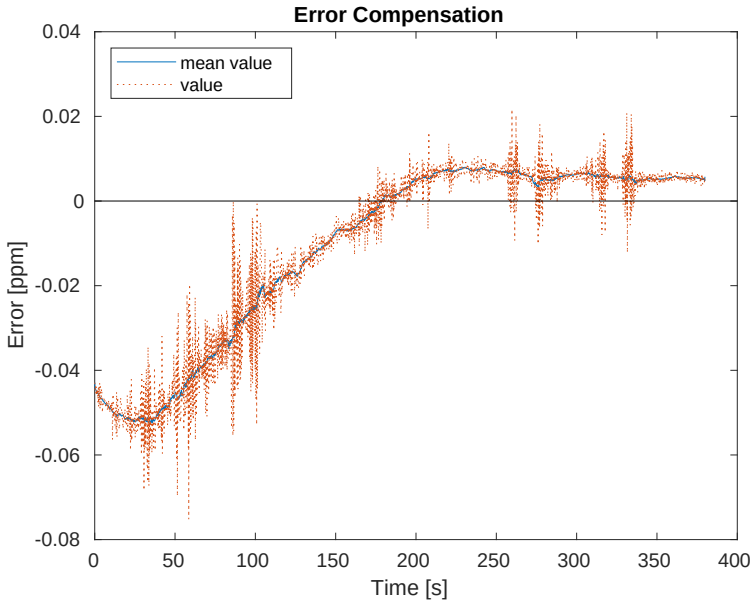
Once an estimate for $\hat{\delta}$ is obtained, we use it in a modified version of the UHD, specifically the low level driver functions `_tune_helper` and `_tune_bbvco` in lib/usrp/-common/ad9361_driver/ad9361_device.cpp, used respectively for tuning the mixer LO frequency and the ADC/DAC sampling rates. In particular, both the requested tuning frequency and sampling rate are divided by $1 + \hat{\delta}$. In UHD, we modified these low-level functions so that the correction is hidden to higher abstraction layers, and no other modification is required in either other UHD library functions or UHD users, i.e. the applications.

Device Calibration. As a demonstration of the achieved calibration, we report in Figures 4.5a and 4.5b the measured errors as a function of time after the switch on, before and after calibration of an Ettus Research B210 SDR, with serial 30AD308. As visible in both figures, the measured errors show a significant transient behaviour with a time constant larger than a hundred seconds. For this reason, the estimate has to be evaluated after the internal temperature of the device has stabilized. We waited approximately seven minutes for each estimate. This thermal transient could be probably shortened by adding a thermal dissipator on the AD9361 (and the FPGA) chips.

Note that the maximum measured error resulted -1.93 ppm, which is consistent with what is stated on the USRP datasheet [131]. The estimated relative error δ we used for Figure 4.5b was -1.82 ppm, which we used to modify the functions `_tune_helper` and `_tune_bbvco` in UHD as stated in the previous section. Inspection of Figure 4.5b shows how, once thermal



(a) δ trend before error compensation.



(b) δ trend after error compensation.

equilibrium is reached, the calibrated device satisfies the 3GPP requirements [47]. Also note how the steady state value in the figure is approximately 0.01 ppm, hinting at a better estimate of -1.81 ppm. We report in Table 4.2 the measured relative errors for four Ettus Research B210 USRPs we tested in our lab. All the measured errors were consistent with the Ettus datasheets.

4.3.2 Noise waveform synthesis

Waveform synthesis for filling just the ZP-CSI-RS subcarriers has been performed as follows. The jammer transmits interference on $N_{\text{noise}} = N_{\text{CSI}}^{\text{RB}} \cdot N_{\text{RB}}$ subcarriers; specifically, for every symbol identified as containing the ZP-CSI-RS according to the detected periodicity and offset, complex Gaussian noise values are generated and mapped to the subcarriers

Table 4.2: Relative error estimates for the devices under test

USRP	Serial Number	δ
B200	30AD308	-1.82 ppm
B200	3128FED	0.83 ppm
B200	3129046	0.85 ppm
B200	30AD345	-0.73 ppm

corresponding to the ZP-CSI-RS.

Let $\mathbf{X} \in \mathbb{C}^{N_{\text{FFT}}}$ the interference vector in the frequency domain, the jamming signal is constructed by assigning independent complex gaussian noise samples to $N_{\text{CSI}}^{\text{RB}}$ subcarriers starting from k' in every RB. All the remaining entries of $\mathbf{X}$ are set to zero. The time domain jamming signal $\mathbf{x} \in \mathbb{C}^{N_{\text{FFT}}}$ is then generated via inverse DFT, as $\mathbf{x} = \text{ifft}(\text{fftshift}(\mathbf{X}))$, and addition of the required cyclic prefix, i.e. a minimum of 9/128 of the duration of the symbol [2]. At this point, the jammer simply copies the temporal samples into the next transmit buffer of the SDR and waits for the hardware scheduler to release them over the air. We do this only for the symbols that the cell has reserved for ZP-CSI-RS. Whenever an upcoming symbol does not contain ZP-CSI-RS, we enqueue a buffer filled with zeros, so that it is feasible to maintain synchronization to the incoming received signal at the same frequency. The srsRAN-UE[107] uses transmit buffers with a size equal to one slot duration.

4.4 Experimental Validation and Results

This section presents the impact of the proposed ZP-CSI-RS jamming attack on a private 5G SA network (see Figure 4.6). The CN was deployed using the Open5GS project[50], while the gNB was implemented with srsRAN_Project [49]. The gNB radio unit was realized with an Ettus USRP N310 SDR[52], configured to operate in the FDD band n2[135] with a 20 MHz channel bandwidth centered at 1970 MHz, $\mu = 0$, and a 4×4 MIMO configuration, and a transmission power of gNB and jammer as low as possible for a successful experiment. In order to satisfy the clock stability requirements for gNB [47], the SDR has a GPSDO module. A Google Pixel 6a and a Samsung Galaxy S25 served as off-the-shelf UEs.

The use of a controlled environment was necessary to ensure full compliance with regulatory and ethical constraints. In fact, performing jamming or interception experiments on public mobile networks is strictly prohibited due to legal restrictions and the potential violation of user privacy and service availability.

4.4.1 Validation of ZP-CSI-RS Detection Method

Before assessing the impact of the jamming attack, we experimentally validated the effectiveness and reactivity of the method for detecting ZP-CSI-RSs, proposed in Sub-subsection 2.3.4.2. In general, since the position of ZP-CSI-RS symbols is unknown and periodic, the detection process requires observing multiple frames to reliably identify the correct symbol. In our case, the ZP-CSI-RS periodicity is known to be 20 slots from the srsRAN_Project

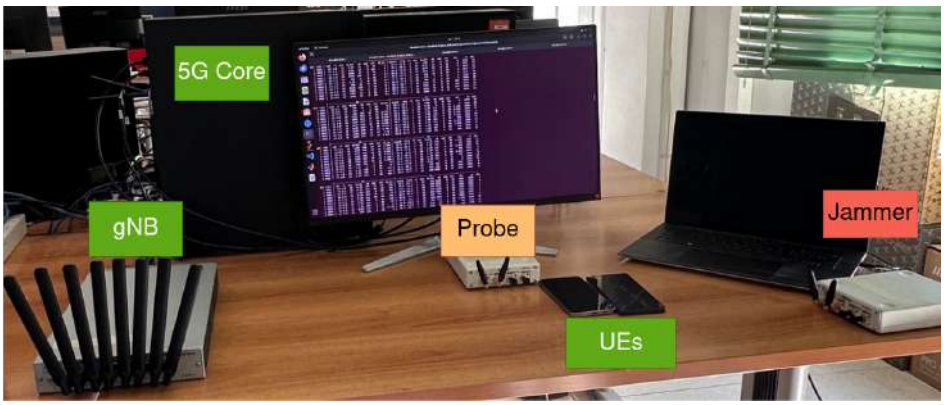


Figure 4.6: Experimental setup for 5G network implementation and testing. The 5G network components – Core Network, gNB, Pixel 6a, Samsung Galaxy S25 – are highlighted in green. The red box indicates the jammer used to perform ZP-CSI-RS attack. The yellow box marks the network probe, which is employed to monitor network performance and behavior during the experiments.

configuration. Therefore, we collected 50 acquisition traces, each lasting 500 ms (250 times the ZP-CSI-RS periodicity) in order to detect symbols also in low traffic conditions and in different BS scheduling configurations. To assess how traffic load influences the reactivity of the detection method, we varied the downlink bitrate to generate slot occupancies of approximately 20%, 60% and 100%, corresponding to an average number of 2, 6 and 10 occupied slots per frame for numerology $\mu = 0$. For each load level, the detection Algorithm 1 described in Sub-subsection 2.3.4.2 was applied to 50 acquisition traces.

We focused on the convergence time of the algorithm, i.e. the delay in terms of average number of frames required to successfully identify the correct estimate of the ZP-CSI-RS period and offset (by setting the number of additional observations c equal to 0). A detection is considered successful if the candidate symbol and its periodicity match the true configuration of the ZP-CSI-RS symbol.

As highlighted in Figure 4.7, the convergence time is inversely correlated with slot occupancy. Bar heights represent the convergence time at 20%, 60%, and 100% of load, shown as average with error bars indicating standard deviation. Under full load conditions (100%), the algorithm typically converges 1.5 times the CSI-RS periodicity (0.5 periods for the first detection and one more period for the second, according to Algorithm 1 in Sub-subsection 2.3.4.2). Theoretical outcomes from the model are depicted by the red dashed line in the figure, showing agreement with the experimental results. This experiment demonstrates that the attack is executable in real-time, even more so since the periodicity of the ZP-CSI-RS remains constant unless the gNB configuration is changed.

Experimental evaluation of the effectiveness of the blind detection method showed that:

- the detection algorithm identifies the ZP-CSI-RS position l' within the slot accurately in all cases.
- the number of REs per symbol $N_{\mathrm{CSI}}^{\mathrm{RB}}$, and the offset within the RB k' configuration parameters were always estimated correctly.

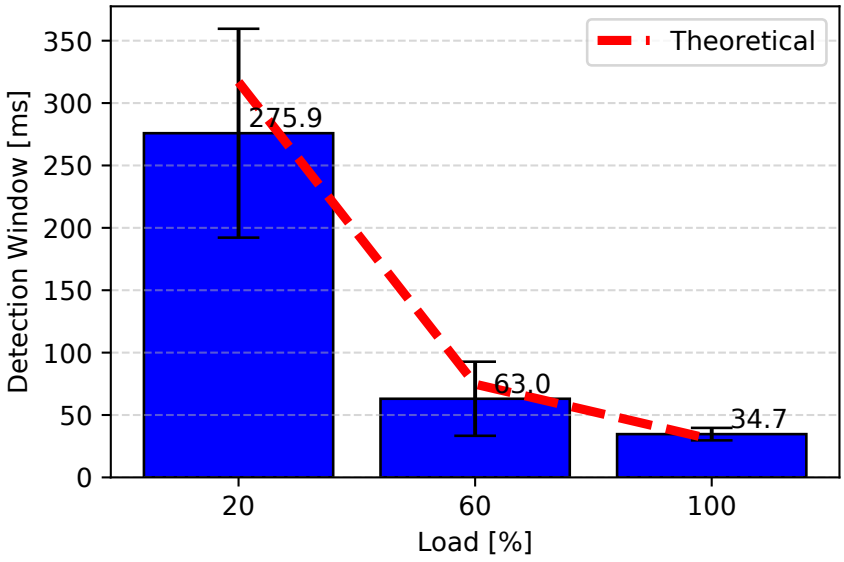


Figure 4.7: Average and standard deviation of the delay for correct detection of ZP-CSI-RS symbol periodicity and offset. The periodicity used in this experiment is 20ms.

Finally, we quantify the probability of correctly identifying a ZP-CSI-RS for different SNR values of the gNB-to-probe link. Also in this case, we vary the load scenarios to assess the impact of a different average reception power in the frame. As shown in Figure 4.8, the probability of correctly identifying the ZP-CSI-RS symbol increases with SNR, with an asynthotic value lower or almost equal to the load. Indeed, this probability representing the p value used in our model, is given by the product of finding a busy RB and applying with success the ZP-CSI-RS detection scheme. When the ZP-CSI-RS probability is maximized (for high SNR values), the probability approaches the normalized load. For low SNR values, the ZP-CSI-RS detection scheme does not work as noise dominates symbol power estimation, and the probability gradually decreases to zero.

The figure illustrates comparable behavior between the smartphone models, depicted by a continuous line for the Google Pixel 6a and a dashed line for the Samsung Galaxy S25. The discrepancy observed between the two handset models in Figure 4.8 can be attributed to the device-specific PMI feedback, which works better for one of the two smartphones. This, combined with the imperfect electromagnetic coincidence between the measurement probe and the antenna phones, results in a gNB-to-probe channel of different quality in each case.

4.4.2 Attack Demonstration an Impact on Throughput

We now analyze the effect of the attack on a UE implemented with a commercial smartphone. Figure 4.9 shows the CSI reports sent by the UE to gNB over time, together with the estimated error rates. Despite of the extremely low error rate, when the attack begins (approximately at $t = 11s$), the CSI report is immediately affected, with a reduction of CQI, MCS and RI parameters. The figure has been obtained by tuning the jamming power so that the increase

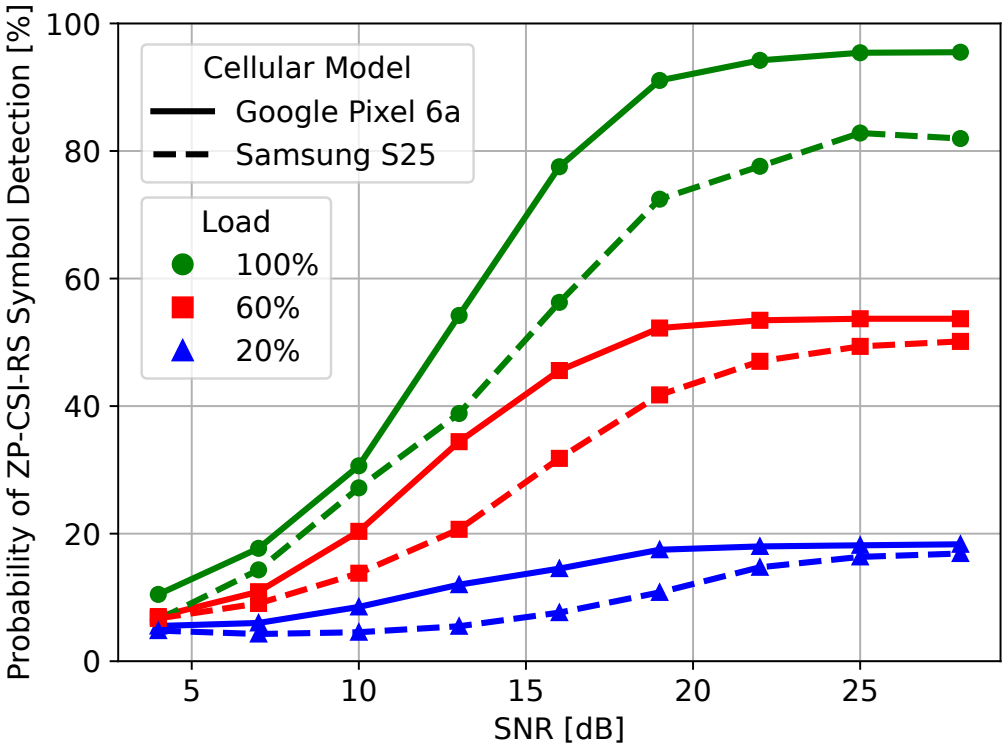


Figure 4.8: Probability of correctly identifying a ZP-CSI-RS symbol as a function of the gNB-to-probe link SNR under different cell load conditions and for two smartphone models: Google Pixel 6a (solid line) and Samsung Galaxy S25 (dashed line).

in wide-band interference-plus-noise power observed by a probe beside the handset is equal to 28dB. We characterize such an increment as the jamming factor.

From the figure, we observe that after a couple of seconds, the RI goes to 1. We remind that the RI reflects the number of spatial layers used for transmission and can take integer values from 1 up to the maximum supported by the UE and gNB, typically determined by their hardware capabilities (in our case 4). This reduction is due to the UE CSI reporting scheme, which decides to reduce the number of layers for increasing the transmission power available in each layer and compensating the overestimated interference level. Consequently, both MCS and CQI parameters progressively enhance to their optimal values, while the RI stays fixed at 1 without adjustment.

The attack also has an immediate and measurable effect in the degradation of the link throughput, as observed in the Figure 4.10. The plots show the bitrate measured by one of the smartphones performing a speed test. At approximately $t = 11\,\text{s}$, when the jammer is activated, the throughput drops. The figure also shows the attack ability of fine-tuning the level of MIMO degradation by adjusting the transmission power of the jammer. By increasing the interference level perceived by the UE, the attacker can effectively hijack the RI feedback, thus progressively reducing the number of selected spatial layers from 4 to 1, i.e. degrading the connection to a single-layer SISO. As a direct consequence, the available throughput decreases significantly, dropping from roughly 240 Mbps to about 70 Mbps. These results confirm that the jammer successfully disrupts the ZP-CSI-RS reporting process, which in

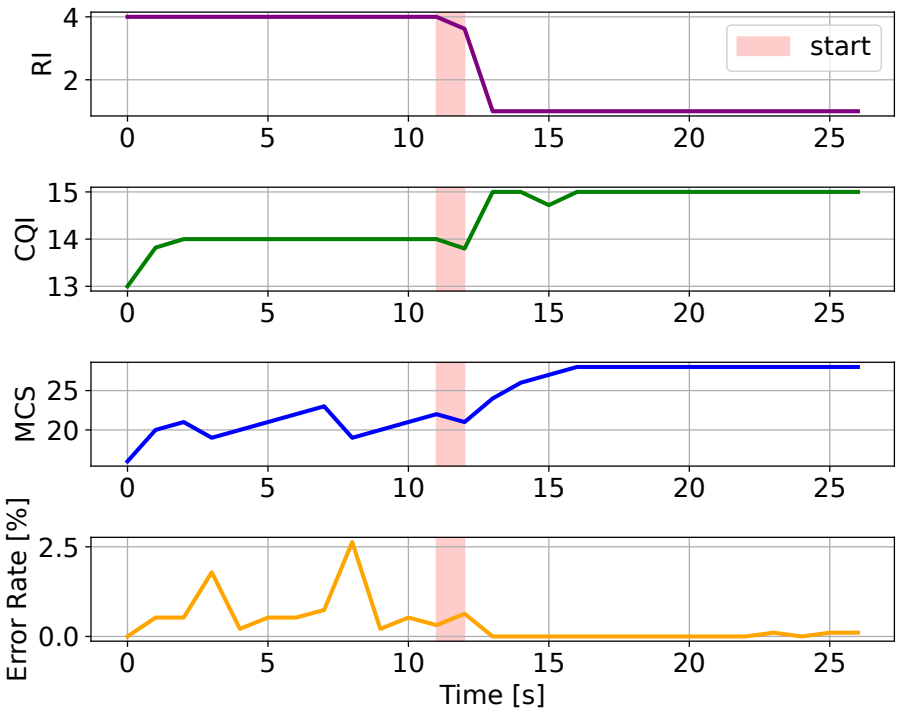


Figure 4.9: CSI reports and error rates of the UE over time: a ZP-CSI-RS attack starts at $t \approx 11s$.

turn causes degraded link adaptation by the gNB.

4.5 Conclusions

This chapter introduced and experimentally validated a practical MIMO downgrade attack that exploits ZP-CSI-RS-based interference estimation to coerce the network into a reduced spatial-rank operating point. We implemented the attacker on a USRP B210 and integrated a blind ZP-CSI-RS configuration detector into an open-source UE stack; the detector correctly identifies ZP configurations with probability above 82% and, when combined with targeted symbol-level jamming, forces a robust fall-back from higher MIMO ranks to a single-layer transmission mode.

The MIMO to SISO downgrade is a pivotal enabler for subsequent passive inference: our current *Golden Sniffer* prototype reliably recovers per-UE parameters (RNTI, scrambling seeds, DCI contents) in single-layer scenarios, but is not yet designed for multi-layer decoding. By reducing the cell to single-layer operation, the attacker therefore creates the exact observational conditions under which *Golden Sniffer* can operate effectively on a real cell. In other words, the active jammer and the passive sniffer are complementary: the jammer simplifies the PHY-layer multiplexing that hinders blind recovery, and the sniffer leverages that simplification to extract the control-plane metadata required for finer attacks.

Building on this insight, the next chapter assembles these components into an end-to-end adversary: the frame-synchronized sniffer is used to capture DCI messages and recover per-

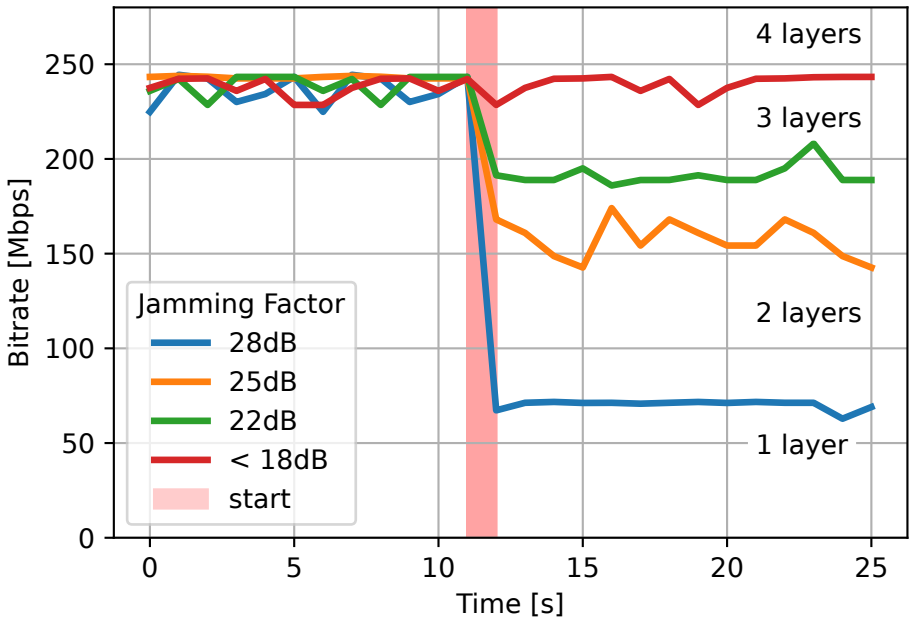


Figure 4.10: Impact of the ZP-CSI-RS jamming attack on downlink bitrate (and number of spatial layers) during a downlink speed test. The jamming factor reports the measured interference and noise power increase.

UE parameters, while a synchronized, frame-aligned transmitter is used to inject forged control information once the sniffer has located the target resources. This combination demonstrates a practical escalation path from passive observability to active control-plane manipulation, and motivates the defensive measures discussed later in the thesis.

Chapter 5

Active Depletion: Inducing High-Power States via DCI Spoofing

This chapter introduces *Silent Drain*, a practical Denial-of-Energy attack targeting commercial 5G UE. Mobile devices implement a rich set of energy-saving mechanisms, for example RRC state transitions (RRC_IDLE, RRC_INACTIVE, RRC_CONNECTED), Discontinuous Reception and scheduler-driven adaptations, that are explicitly designed to reduce average power draw while preserving user experience. These mechanisms, and their continued refinement across 3GPP releases, are central to 5G's energy-efficiency agenda and are widely used in operator networks and device stacks.

At the device level, the power consumption of a smartphone's cellular subsystem depends strongly on radio state, monitoring duty cycle, uplink activity, MCS and MIMO usage, and scheduler decisions; empirical studies therefore commonly profile energy as a function of RRC state transitions, DRX cycles and transmission parameters to build realistic power models.

Building on empirical power-profiling results, we observe that the very mechanisms designed to reduce energy use can be intentionally disabled or coerced into higher-consumption behaviour by manipulating control-plane signaling. In practice, carefully crafted control messages and scheduling manipulations can prevent a UE from entering low-power modes (or force it to remain in high-activity states), increase monitoring duty cycles or induce prolonged uplink activity, thereby neutralizing DRX/RRC power savings and producing sustained, measurable increases in power draw.

5.1 Introduction

Energy efficiency is a fundamental design goal of 5G networks, crucial for extending the operational lifetime of battery-powered mobile devices and IoT nodes [136]. To achieve this, the 5G NR standard integrates mechanisms such as lean carrier design, flexible Radio Resource Control (RRC) states, and Discontinuos Reception (DRX) [6, 5]. These mechanisms significantly reduce idle power consumption by minimizing always-on transmissions

and enabling long sleep cycles.

At the device level, the power consumption of a UE is strongly influenced by its connection state and by the parameters it receives over the PHY-layer control channel. The RRC protocol defines three main states. In *RRC IDLE*, the UE performs only basic cell search and paging reception, consuming minimal power. The *RRC INACTIVE* state retains the network context while reducing radio activity, enabling faster resumption of data transfer. In *RRC CONNECTED*, the UE maintains an active link with the gNB, sustaining full radio and baseband operation.

The transitions between these states are regulated by inactivity timers and explicit control messages [6], which are implicitly influenced by received DCI messages on the PDCCH. A received scheduling grant or an uplink grant will typically reset inactivity timers and schedule immediate or near-term UE activity. Thus, the timing and content of DCI have a direct effect on the UE's radio duty cycle and, consequently, on its instantaneous and average power consumption.

A central mechanism for energy saving in RRC_CONNECTED is DRX, which allows the UE to cycle between *on* (monitoring PDCCH) and *off* (sleep) intervals according to configured timers and cycles. Denoting by T_{on} the aggregate time spent awake per DRX cycle and T_{cycle} the total cycle period, the awake duty cycle is $\eta = \frac{T_{\text{on}}}{T_{\text{cycle}}}$. A simple model for the mean power consumption is therefore $P_{\text{avg}} = \eta P_{\text{active}} + (1-\eta) P_{\text{sleep}}$, where P_{active} and P_{sleep} are empirically measurable device power levels in active and sleep sub-states. Because $P_{\text{active}} \gg P_{\text{sleep}}$, even modest increases in η (for example caused by more frequent wake-ups or longer on-durations) lead to noticeable increases in P_{avg}.

PHY-layer parameters further modulate energy consumption. The MCS and the number of spatial layers (MIMO rank) jointly determine the transmission time and the baseband processing load. Higher MCS values reduce airtime per bit (thus potentially lowering the radio transmission energy), but higher-order modulations and multi-layer MIMO require more complex signal processing (higher baseband CPU usage and RF chain activation), which increases instantaneous power. The effect on energy depends on the trade-off between reduced transmission duration and increased processing power; in practice, there exist operating points where the device remains in a high-power state for longer due to repeated retransmissions or scheduler-driven uplink bursts.

These observations identify concrete adversarial levers. Starting from detailed power-profiling, an adversary can craft control-plane signaling that (i) prevent the UE from entering low-power RRC states by continually issuing or replaying scheduling grants that keep inactivity timers active, (ii) increase η by forcing more frequent PDCCH monitoring (e.g., by creating perceived downlink activity or frequent uplink grants), and (iii) induce unfavorable PHY-configurations (e.g., repeated low-MCS assignments or artificially fragmented uplink allocations) that raise baseband processing or retransmission rates. Concretely, the adversary does not need to decrypt higher-layer RRC messages: manipulating or injecting DCI-like control messages (that pass CRC checks masked by the RNTI) or replaying previ-

ously observed control-plane patterns can be sufficient to bias the UE's state machine toward high-consumption behaviour.

From a measurement point of view, the attack design relies on three empirical components: (1) *profiling*: accurate characterization of P_{active}, P_{sleep} and their dependence on RRC/-DRX/MCS/MIMO; (2) *trigger mapping*: correlating specific control-plane events (e.g., grant types, timing offsets, MCS signals) with instantaneous power increments; and (3) *closed-loop validation*: verifying that injected or replayed signaling produce the expected sustained increase in P_{avg} over relevant time scales. These components also inform the detection thresholds and the selection of efficient attack primitives that maximize energy impact while minimizing conspicuous traffic.

Importantly, these attack primitives exploit predictability rather than cryptographic weakness: control messages on the PDCCH are validated using CRC masking with the RNTI and scrambling sequences, which provide integrity checks but are not a substitute for cryptographic authentication. Where an attacker can observe or guess the required masking parameters (or coerce the network into simpler operating points), the adversary can operate within the protocol's decoding semantics to produce the desired UE behaviour.

5.2 Related Work

The security framework for 5G networks has been a central topic in standardization bodies, resulting in an architecture designed to support a wide range of services, from massive Internet of Things (IoT) deployments to Vehicle-to-Everything (V2X) communications [137]. This architecture introduces new security functions and procedures intended to provide a robust defense against a variety of threats [138]. However, as we already know, despite the continuous development of new countermeasures, the inherent openness of the wireless medium means that the PHY-layer remains susceptible to attacks that can undermine network performance and security. A significant consequence of such attacks, focus of this work, is the impact on the energy consumption of network devices.

Energy efficiency is a foundational requirement for 5G systems, particularly given the projected scale of mobile devices. Physical layer attacks are a direct method for executing such service denials. Wang et al. [139] provide a comprehensive survey of physical layer security in 5G networks, categorizing threats into active and passive types. Active attacks, such as jamming and spoofing, force legitimate devices to engage in energy-intensive operations, including increasing transmission power, performing complex signal processing for interference cancellation, or engaging in repeated retransmissions, all of which directly deplete power reserves.

Specific attack vectors have been studied to understand their impact on energy resources. Flooding attacks, a form of DoS, are examined in the context of dense wireless sensor networks, a scenario analogous to massive IoT in 5G [140]. This work shows that flooding a network with superfluous packets forces legitimate nodes to needlessly process incoming

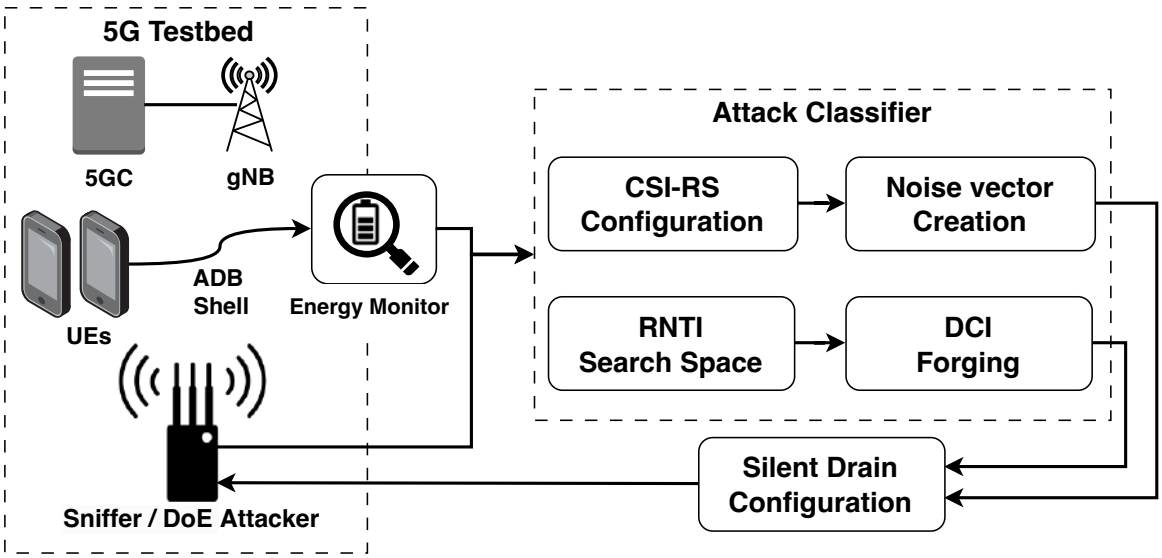


Figure 5.1: Workflow of the proposed method.

data, consuming significant energy and reducing network lifetime. A more targeted approach is the Depletion of Battery (DoB) attack[141]. In this work authors, investigate this threat in 5G-connected UAV networks, demonstrating how an adversary can exploit high-priority control plane messages, such as the Packet Forwarding Control Protocol (PFCP), to keep a device's processor active and prematurely drain its battery life. This highlights how vulnerabilities in higher-layer protocols can be leveraged to mount an attack with direct physical layer consequences on energy consumption as described in[15].

The potential for applying machine learning to identify recurring network issues, as explored in [142], for predicting technical ticket reopening, suggests a possible direction for detecting patterns associated with persistent energy-draining attacks. Finally, Maiwada et al. establish a direct connection between network attacks and energy inefficiency, identifying Distributed Denial of Service (DDoS) attacks as a primary cause of excessive power drain [136].

5.3 Methodology

Our methodology follows a two-phase approach (see Figure 5.1): (1) systematic profiling of UE energy consumption under controlled network configurations, and (2) targeted exploitation of the most energy-demanding conditions through a DoE attack.

5.3.1 Phase 1: Energy Profiling

The goal of Phase 1 is to identify, in a reproducible and measurable way, the network configurations and radio parameters that maximize the UE's energy consumption per delivered bit. The profiling is performed in a controlled testbed (see Section 4.4) to ensure repeatability.

The UE power is measured via Android Debug Bridge (ADB) Shell to log timestamps and instantaneous current consumption. We fix Radio Frequency conditions (signal strength,

noise floor) and disable unrelated background services on the UE to minimize external power fluctuations.

5.3.1.1 Profiling scenarios and variables

Profiling methodically evaluates the following network configurations, each chosen due to its known effect on UE power.

- *RRC States:* RRC_IDLE, RRC_INACTIVE and RRC_CONNECTED; transitions between these states are provoked by controlled traffic bursts and explicit detach/attach procedures.
- *DRX Configuration:* multiple DRX cycles and on-duration timers (short, medium, long); we include extreme cases (DRX disabled) to observe worst-case duty cycles.
- *Scheduling Policy:* representative scheduler behaviours (Round Robin, Proportional Fair) are emulated on the testbed to study effects of grant timing and fairness on energy.
- *Traffic Patterns:* medium (web-like flows) and high intensity (saturated `iPerf` uplink/-downlink) to measure both idle and saturated regimes.
- *Modulation and Coding Scheme:* representative low and high MCS points (e.g., QPSK vs 64QAM) to observe tradeoffs between airtime and processing load.
- *MIMO Configuration:* SISO and 2x2 MIMO (single vs multi-layer) to quantify baseband processing overhead.

For each combination of the above variables we run the selected traffic profile, capture the power trace for a fixed interval (120 s), and store the aligned trace for offline analysis.

5.3.1.2 Metric

We normalize performance using an *energy-per-bit* metric: $\gamma_e = \frac{P_{\text{avg}}}{R_b}$ [J/bit], where P_{avg} is the average power measured during the run and R_b is the achieved application-level bit rate measured at the IP layer.

Outcome of Phase 1: The profiling identifies configurations with the highest γe, such as: (1) Connected state with saturated uplink, (2) DRX disabled, (3) low MCS under high traffic, (4) suboptimal scheduling patterns, and (5) single-layer transmissions. These insights directly inform our attack design.

These observations directly inform Phase 2: the design of attack primitives that (i) prevent low-power transitions, (ii) increase the awake duty cycle, and (iii) force PHY configurations that maximize baseband cost. The following section describes how we translate profiling findings into a practical DoE attack while controlling for detectability and protocol conformance.

5.3.2 Phase 2: Attack Design and Implementation

5.3.2.1 Threat Model

The goal is to force the UE into high-energy states identified in Phase 1 without delivering useful data. We assume an attacker with: (1) An SDR capable of transmitting 5G NR-compliant downlink signals. (2) Synchronization with the target cell via PSS/SSS decoding. (3) Knowledge of the UE's RNTI and Search Space through a passive sniffer. Using a synchronized framework, it is possible to inject forged DCI messages targeting the victim's RNTI. DCI messages are transmitted in the correct PDCCH Search Spaces to:

- Maintain the UE in *RRC CONNECTED* by replaying valid-looking DCIs at regular intervals.
- Assign uplink resources with custom MCS and RB allocations.

5.3.2.2 Attack Scenarios

The attack scenarios presented in this work are:

1. *RRC Connection Keep-Alive*: periodic DCI replays prevent inactivity timers from expiring.
2. *Persistent Uplink Drain*: forged DCI messages to trigger continuous transmissions, persisting even after UE detachment due to retransmission loops.

Link to Phase 1 The parameters to forge injected DCI, in both scenarios, are chosen based on the worst-case energy conditions observed in profiling (e.g., DRX disabled, low MCS in a single-layer cell configuration). This ensures that each forged allocation maximizes the UE's power draw.

5.4 UE Energy Consumption

This section presents a systematic measurement of UE energy consumption in various configurations of 5G networks. The objective is twofold: (1) quantify the impact of RRC states, DRX cycle, scheduling policy, MCS level, and MIMO configuration on power draw, and (2) identify worst-case configurations that can later be exploited in targeted DoE attacks. Each experiment lasts about 120 seconds, and each mechanism for energy efficiency is tested under two traffic conditions: moderate (with flood-ping) and intensive (with iPerf).

5.4.1 RRC State Analysis

To estimate the energy footprint of each RRC state, we evaluated the power absorbed by the smartphone during:

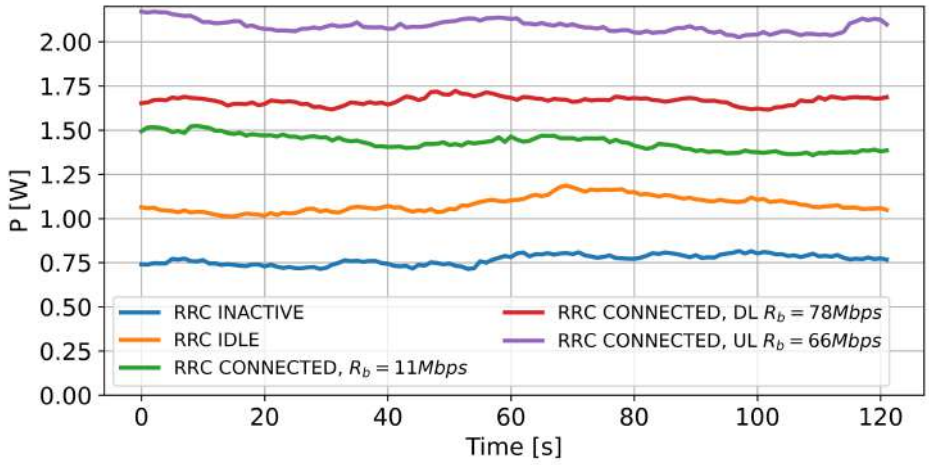


Figure 5.2: Impact of RRC state and traffic load on energy consumption.

- `RRC IDLE`: In this case, we are measuring the power that smartphone needs to stay-on and do cell search. We can reach this state by stopping all traffic sources and waiting for the inactivity timer to expire.
- `RRC INACTIVE`: This is the state reached by the smartphone when all traffic is stopped, but the inactivity timer is not expired yet.
- `RRC CONNECTED`: We evaluate this state under three traffic conditions to highlight the impact of different load levels:
  - **Moderate traffic (flood ping)**: Periodic ICMP echo requests were sent from the gNB to the UE, generating downlink/uplink activity (about 11Mbps in both directions, single-layer), keeping the UE in connected mode.
  - **Downlink saturation (iPerf DL)**: We generate continuous UDP downlink traffic using iPerf, saturating the UE's receive capacity, with about 76Mbps in single-layer downlink.
  - **Uplink saturation (iPerf UL)**: Similarly, the UE transmits UDP data at full capacity using iPerf in uplink mode, measuring about 66Mbps in single-layer uplink.

Although we did not directly capture state transitions, the steady-state current profiles provide insight into the energy characteristics of each RRC mode and how they scale with traffic intensity in `RRC CONNECTED`. Figure 5.2 reveals a clear gradient in current draw, with idle and inactive states consuming significantly less power. As expected, the `RRC Connected` state incurs the highest energy drain, particularly under saturated traffic conditions. In particular, the difference between moderate and high traffic (both DL and UL) in `RRC Connected` demonstrates the substantial impact of data activity on the power consumption of the UE, the UL saturation case showing the highest energy drain.

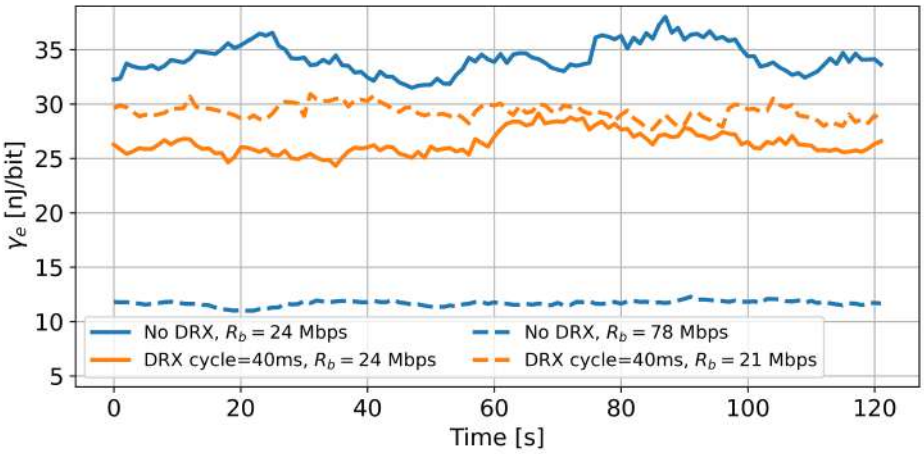


Figure 5.3: Impact of DRX on γ_e with moderate traffic (solid line), high traffic (dashed line) scenarios.

5.4.2 Impact of DRX Configuration

To evaluate the energy-performance tradeoff introduced by DRX, we measured the current consumption of the UE under two operational modes:

- **DRX enabled:** the UE is configured to wake up for 10ms every 40ms; the configuration in srsRAN could be done using these parameters (specifically, in `drx` section): `long_cycle`= 40ms, `on_duration_time` = 10ms.
- **DRX disabled (always-on):** the UE remains continuously active by setting `long_cycle` = 0.

As shown in Figure 5.3, enabling DRX in moderate traffic scenario leads to a notable lower γ_e, even with a relatively short cycle of 40ms. While longer DRX cycles could potentially yield larger energy savings, such configurations are not supported in our testbed: values above 40ms result in de-synchronization between the UE and gNB in `srsRAN`, resulting in a higher required γ_e in our experiments. In saturated downlink traffic conditions, enabling DRX can reduce instantaneous throughput to a fraction of its maximum, as data can only be scheduled during on-duration periods. As a result, the same volume of data is delivered over a longer time, extending the total UE's active period.

5.4.3 Scheduling Policy Impact

We evaluated the energy implications of two resource allocation strategies implemented in the `srsRAN` gNB scheduler: RR (Round Robin) and Proportional Fair (PF). RR is the default scheduler in `srsRAN`, which allocates resources uniformly among all connected UEs, regardless of channel conditions or QoS requirements. Although RR offers strict fairness in terms of time-domain resource allocation, it does not consider radio link quality or traffic priority, which can lead to inefficient spectrum usage or degraded service for latency-sensitive flows. To address these limitations, we activated the PF scheduler by enabling the

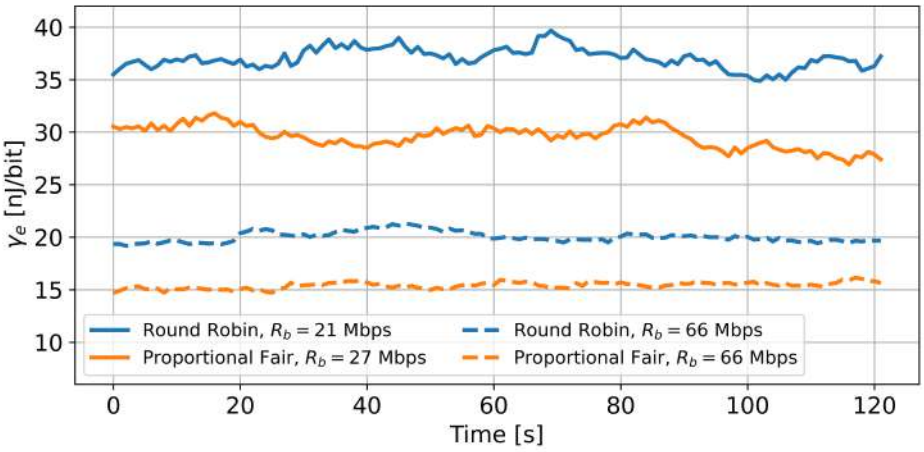


Figure 5.4: Impact of scheduling on γ_e with moderate traffic (solid line), high traffic (dashed line) scenarios.

`sched_expert_cfg` section in the gNB configuration file.

In both cases, a second UE was connected to the same gNB to produce independent traffic (e.g. video streaming), ensuring non-trivial scheduling behavior beyond single-UE allocation. Current consumption was recorded under stable radio and application-layer conditions, identical across both scheduling configurations.

The experiments expose that, (Figure 5.4), the PF scheduler achieves better efficiency, in both traffic scenarios, with an improvement of up to 10nJ/bit in moderate traffic and approximately 5nJ/bit under iPerf traffic. Note that the measured γ_e is lower in the high-load iPerf case, suggesting that sustained and predictable scheduling leads to more stable RF behavior and fewer state transitions, thereby reducing waste.

5.4.4 Modulation Coding Scheme Impact

To evaluate the energy impact of different MCS, we forced the UE to operate at fixed uplink and downlink MCS levels by configuring the `pusch` and `pdsch` section of the gNB. We considered two modulations: QPSK (MCS = 9) and 64-QAM (MCS = 28). In both traffic scenarios, higher MCS levels consistently lead to better energy efficiency. As shown in Figure 5.5, the use of lower MCS values increases γ_e due to prolonged airtime per bit, which extends RF chain activity and contributes to higher energy expenditure.

- `min_mcs = max_mcs = 9` (QPSK)
- `min_mcs = max_mcs = 28` (64-QAM)

The results show as under low traffic conditions, the energy gap reaches up to 5nJ/bit in favor of the high MCS setting. This difference becomes even more pronounced under high-load traffic, where the low-MCS configuration incurs up to 30nJ/bit more than the high-MCS case.

These findings indicate that higher-order modulation not only improves spectral efficiency but also leads to better energy performance, regardless of traffic intensity.

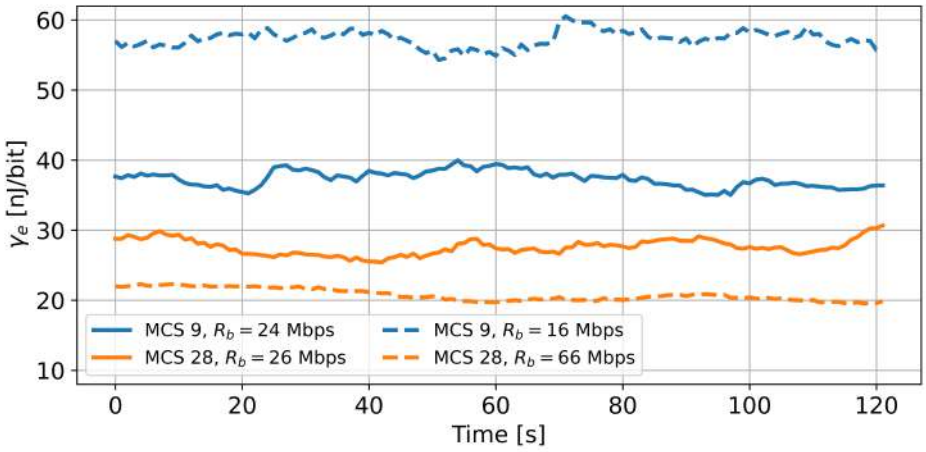


Figure 5.5: Impact of MCS on γ_e with moderate traffic (solid line), high traffic (dashed line) scenarios.

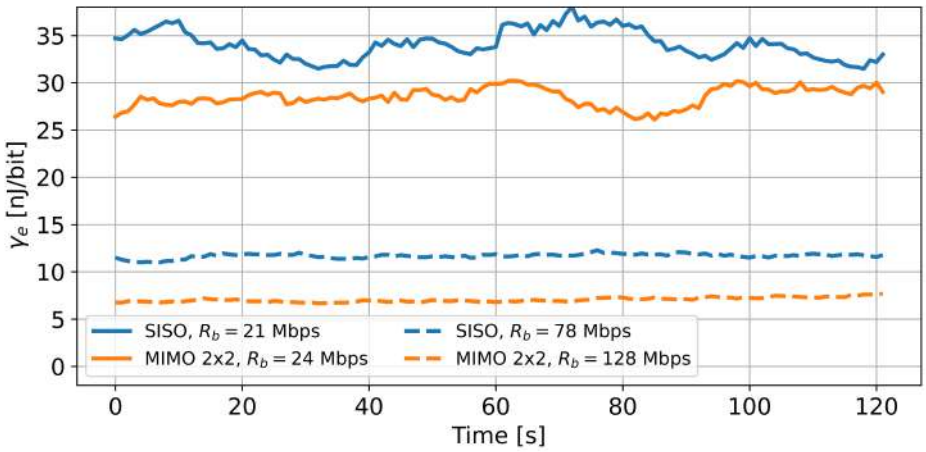


Figure 5.6: Impact of number of layer used on γ_e with moderate traffic (solid line), high traffic (dashed line) scenarios.

5.4.5 MIMO Configuration

To assess the energy impact of spatial diversity, we examined the behavior of the UE under two MIMO configurations. Specifically, we tested a SISO configuration and a MIMO 2x2 configurations.

Regardless of the configuration, we verified via low-level logs that the UE consistently transmits using a single antenna, while the number of receiving antennas varies depending on the channel quality [5].

Contrary to the common assumption that MIMO configurations lead to higher energy consumption due to the activation of multiple RF chains, our efficiency analysis shows the opposite trend. As illustrated in Figure 5.6, we observed a reduction of approximately 5nJ/bit when using MIMO, indicating that the ability to transmit more data in less time outweighs the potential cost of activating additional hardware.

Summary of Profiling Results: Table 5.1 highlights configurations considered highly exploitable.

Parameter	Configuration	Exploitability
RRC State	Connected + UL Saturation	High
DRX	Depends on traffic conditions	Medium
Scheduling	Round Robin	Low
MCS	Low (QPSK)	High
MIMO	Single-Layer	High

Table 5.1: Summary of profiling results and exploitability assessment.

Those marked as high exploitability represent worst-case conditions that substantially increase the UE's energy consumption. These empirical results are strictly linked to the parameter choices in the *Silent Drain* attack.

5.5 Silent Drain Attacks

Based on the worst-case energy configurations identified in Section 5.4, we implement *Silent Drain*, a family of DoE attacks targeting different physical layer parameters of a 5G UE. The common objective is to keep the UE in highly energy-inefficient states without delivering useful data.

The attack setup is the same used in Section 4.4. Summarizing, the tool enables the following operations:

- **Cell Search and Synchronization:** The attacker can synchronize with the target cell via PSS/SSS and PBCH decoding.
- **System Frame and Slot Timing Estimation:** After successful synchronization, the tool estimates the System Frame Number (SFN) and slot index, which are essential to ensure that signaling messages are transmitted with precise timing.
- **Transmission of Custom Signals:** Once synchronization and timing estimation are complete, the attacker can transmit noise for rank downgrade (already presented in Chapter 4), spoofed or replayed DCI messages targeting the RNTI associated with the victim UE.

The knowledge of the UE's RNTI and cell configuration is due to *Golden Sniffer*, presented in Chapter 2. DCI messages are pre-computed offline using MATLAB and transmitted in real time by the attacker. Moreover, to maximize the energy impact of the attack, we spoofed both DCI 1_0 and DCI 0_0 messages. This induces the UE to unnecessarily activate its radio chains, either to transmit or to listen for data, thereby increasing power consumption even in the absence of actual traffic.

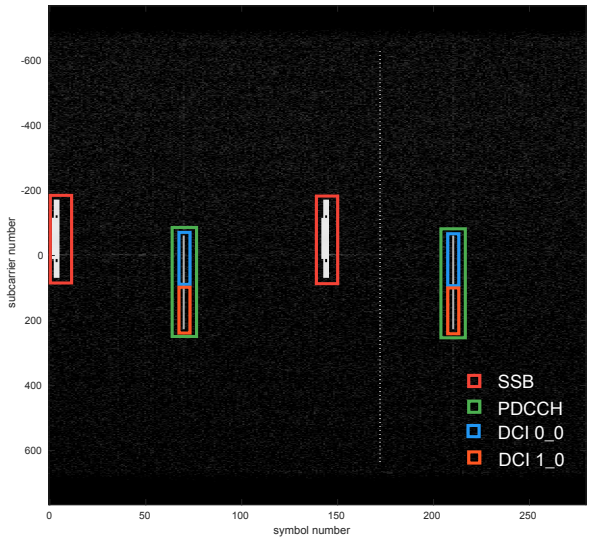


Figure 5.7: DCI replays transmitted every 10ms.

5.5.1 Variant 1: Persistent Connected State

Objective: Prevent the UE from entering low-power RRC states (Inactive/Idle) by periodically injecting forged DCIs.

Initial Conditions: The UE is connected to a single-layer cell, operating in `RRC CONNECTED` state with DRX disabled and RR scheduling. In this baseline configuration, the UE would normally transition to lower-power states after the inactivity timer expires if no data activity is detected.

Mechanism: Every 10 ms (see Figure 5.7), corresponding to the minimum DRX cycle allowed by the standard, the attacker transmits a forged allocation for the victim's RNTI, ensuring that the inactivity timer never expires.

Impact: Although no actual data is transmitted, the UE interprets the fake downlink signaling and tries to decode the downlink data, and the fake uplink grants are legitimate, triggering dummy scheduling activity. In fact, we empirically observe that upon receiving each forged uplink grant, the UE attempts a transmission, even if there is no data to send. This confirms that uplink activity is stimulated by the grant, without requiring application-layer triggers, amplifying the energy impact of the attack.

5.5.2 Variant 2: MCS and Resource Allocation Manipulation

Objective: Force the UE into low spectral efficiency modes while consuming maximum resources.

Initial Conditions: The UE is connected to a SISO cell with DRX disabled and RR scheduling. At the application layer, the UE is generating a traffic with $R_b = 11$ Mbps.

Mechanism: Excluding slot 0 to avoid interfering with the SSB, in every slot (see Figure 5.8) the attacker injects, into the UE's Search Space, forged DCI messages that:

- Set a low MCS index (e.g., QPSK), which reduces spectral efficiency and increases the airtime required per bit.

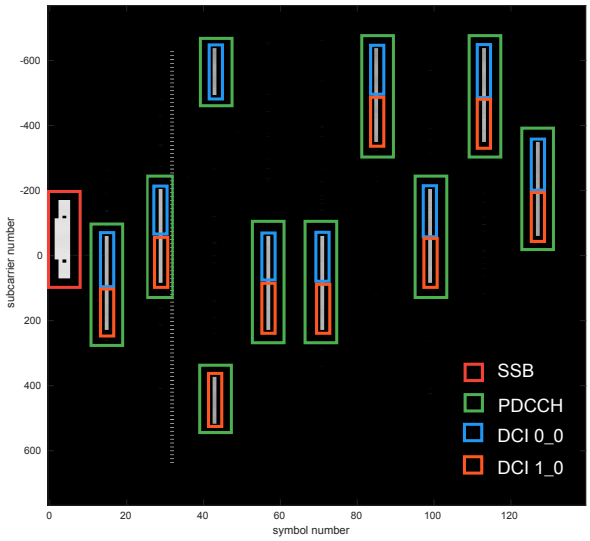


Figure 5.8: Forged DCI 0_0 and DCI 1_0 messages periodically injected, triggering uplink transmissions.

- Set the Frequency Domain Resource Assignment (FDA) field to allocate all available RBs, forcing the UE to process the maximum channel bandwidth in each slot.

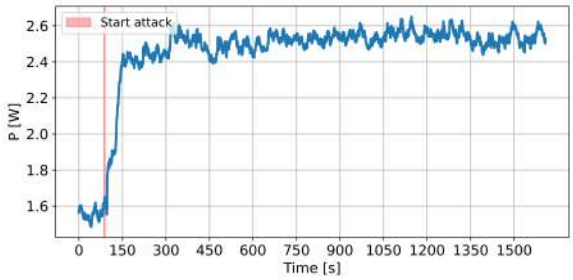
Due to the attacker's proximity, the forged PDCCH messages override the legitimate gNB control channel, effectively saturating all uplink and downlink scheduling opportunities.

Impact: The UE processes large volumes of control and data channel resources inefficiently. Upon receiving each forged uplink grant, the UE transmits on the allocated resources, leading to de-synchronization with the legitimate gNB and, eventually, AMF-triggered disconnection. However the UE enters a persistent retransmission loop, non-compliant with 3GPP standards, repeatedly sending the same transport blocks. Unexpectedly, retransmissions continue indefinitely, over 30 minutes in our experiments, even after the UE has lost connection to the serving cell.

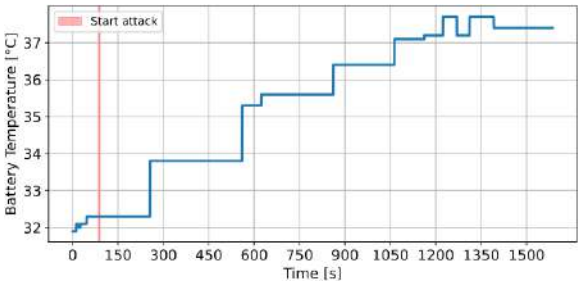
Figure 5.9a reports the measured *power* consumption during this phase (not γ_e), as the uplink R_b during the disconnected retransmission loop is not under our control. In principle, the uplink throughput could be inferred by spectrum analysis, as the uplink channel appears fully occupied during the loop, but we do not explicitly measure it.

The measured power is anomalously high, as also reflected in the increase of battery temperature (Figure 5.9b), remaining over +1 W compared to the `RRC Connected` baseline. This highlights a vulnerability in the UE's retransmission logic: in the absence of gNB acknowledgments, the UE continues its efforts indefinitely. Such behavior can be exploited to drain the battery with minimal attacker effort, without requiring core network interaction.

To verify that this phenomenon was not due to a device-specific firmware, we repeated the attack on a second commercial device, a Samsung Galaxy S25. The results were consistent, confirming that the observed behavior is not limited to a single model or vendor.



(a) Absorbed average power



(b) Battery temperature

Figure 5.9: Effects on: (a) Absorbed mean power and (b) Battery temperature during a *Silent Drain* Attack.

5.6 Conclusion

In this chapter, we showed that systematic energy profiling of a commercial 5G UE can reveal specific network configurations that significantly increase power consumption. Based on these measurements, we designed the *Silent Drain* attack, which injects forged DCI to force the UE into or maintain it within such high-consumption states. In our testbed, these conditions sustained $\approx +1\ W$ additional power draw and prolonged uplink activity for over **30 minutes** after disconnection from the serving cell.

To mitigate such threats, we propose: (i) integrity protection or lightweight authentication of DCI messages to prevent spoofing; (ii) UE-side anomaly detection that correlates energy usage with expected traffic and operational states; (iii) gNB–side monitoring to identify abnormal grant patterns and terminate suspicious activity promptly.

Future work will focus on profiling to multiple devices and frequency bands, assess the feasibility of the attack in real world scenario. We also plan to investigate energy–aware intrusion detection methods capable of recognizing sustained high–power states unrelated to legitimate traffic.

Conclusions

This thesis has explored the practical attack surface of 5G Radio Interface by progressively combining passive PHY-layer observability, traffic-flow inference, and targeted active manipulation. The work began with the development of a purpose-built 5G sniffer capable of decoding both control-plane and user-plane channels, including PDCCH, PDSCH, and key reference signals such as ZP-CSI-RS and NZP-CSI-RS. This instrument provided fine-grained visibility into scheduling behavior, resource allocation, and channel-state reporting.

Building on this foundation, the thesis examined how decoded control- and user-plane metadata can be leveraged to perform passive traffic analysis. Using only observable PHY features, we showed that application-level behavior remains surprisingly distinguishable: traffic patterns could be classified reliably both within the same service category (e.g., Amazon vs. eBay vs. Shein) and across distinct categories (e.g., Shein vs. Netflix). These findings highlight that even in encrypted and privacy-oriented cellular stacks, side-channel leakage through control signaling still exposes meaningful behavioral fingerprints.

The experimental platform was then extended from observation to intervention. Leveraging the ability to sniff ZP-CSI-RS, we designed and executed a downgrade jamming attack targeting the PHY-layer. By selectively interfering with channel-state reporting, the system was coerced into adopting substantially worse transmission configurations, including a forced reduction of the spatial rank (e.g., from rank 4 down to rank 1). The resulting degradation in link quality and throughput demonstrates that finely timed, CSI-aware interference can have severe practical impact on 5G performance.

Finally, the same instrument was used to move from PHY-level manipulation to control-plane compromise. By aligning frame-synchronized actuation with decoded PDCCH information, we demonstrated a DCI spoofing attack capable of influencing scheduling behavior end-to-end. Beyond its security implications, this manipulation produced measurable secondary effects; in particular, altered smartphone energy consumption indicates that subtle control-plane tampering can propagate into operational and privacy-relevant consequences.

Overall, the trajectory of this work shows how passive leakage, traffic-pattern inference, and precise active manipulation compose into a coherent attack chain. The results emphasize that 5G systems, despite their architectural improvements, remain susceptible to adversaries who combine observability with timing-accurate actuation, offering insights that are both experimentally grounded and operationally significant.

Collectively, these contributions provide an integrated experimental methodology and a

set of empirical findings that uncover previously under-appreciated PHY and control-plane vulnerabilities in 5G deployments.

Practical Applicability and System Boundaries

To better delineate the operational boundaries of the proposed techniques, it is essential to explicitly consolidate the main assumptions and practical limitations encountered during this research. While the experimental evaluation proves the feasibility of the identified attack chain, its real-world execution is constrained by several factors:

- **Radio Environment Dependencies:** The reliability of the 5G NR sniffer and the accuracy of active signal injection are strictly dependent on favorable radio conditions. Achieving reliable synchronization and accurate parameter inference (e.g., for CSI-aware interference) requires a SNR and relative proximity to the target gNB or UE.
- **Scenario Dynamics:** The current methodology was primarily validated in controlled environments. Highly dynamic multi-user scenarios could introduce interference patterns that could affect the DCI detection rate and the overall success of real-time traffic inference.
- **Implementation Nature:** The developed sniffer currently operates in a non-real-time mode. While this is sufficient for demonstrating vulnerabilities and performing post-processing analysis, moving toward an ”on-the-fly” attack would require further algorithmic optimization and hardware acceleration to handle the strict timing requirements of the 5G PHY-Layer.
- **Vendor-Specific Variability:** Several aspects of the 5G standard allow for vendor-specific configurations (e.g., CSI feedback computation). Such variability may introduce potential deviations in the behavior of commercial networks compared to the srsRAN-based testbed.

Open Iusses

While the work reported here is comprehensive within the controlled environments and scenarios considered, several important open issues remain that should guide future research and standardization efforts.

Countermeasures and mitigation. This thesis has already taken initial steps toward mitigation by analyzing how specific PHY-layer design choices influence passive observability. In particular, as discussed in Chapter 3, we showed that shaping reference-signal patterns (e.g., through cosine-squared masking of DCI resources) can significantly reduce the effectiveness of passive sniffing and traffic classification without fundamentally altering protocol operation. However, these measures remain preliminary. There is a critical need for a broader and systematic design and empirical evaluation of countermeasures, including lightweight

PHY-level integrity checks, reduced observable metadata encodings, robust scheduling policies, and cryptographic protections where feasible. The trade-offs between security, latency, spectral efficiency, and backward compatibility must be carefully characterized.

Anomaly detection and ML-based defenses. Several of the attacks demonstrated in this thesis induce distinctive PHY- and MAC-layer artifacts, such as abnormal scheduling persistence, rank downgrades, or atypical DRX behavior, which could in principle be exploited for detection. While this work does not implement a full detection pipeline, the experimental traces collected across Chapters 4 and 5 already suggest promising features for anomaly detection. Future research should focus on developing machine-learning-based detectors trained on such PHY signatures and scheduling patterns, producing labeled datasets, evaluating robustness against adaptive adversaries, and quantifying false-positive and false-negative trade-offs in realistic operational settings.

Energy–privacy nexus. This thesis has explicitly demonstrated that active control-plane manipulation can measurably alter the energy consumption profile of commercial smartphones, as shown in Chapter 5. The observed reduction in power draw under DCI spoofing highlights a previously unexplored side-channel at the intersection of energy efficiency, privacy, and security. A systematic study of power-consumption side-effects remains an open research direction. Understanding whether such effects can be reliably monitored by the network or the device itself is particularly relevant for future defense mechanisms.

Standardization and policy implications. Some of the vulnerabilities and mitigations identified in this work stem from fundamental design choices in the 5G NR physical and control layers. While certain countermeasures (e.g., reference-signal shaping or scheduling randomization) could be deployed locally, others may require changes to protocol specifications, such as minimizing observable control metadata or providing integrity protection for selected control messages. Translating these findings into deployable protections will require engagement with standardization bodies and mobile network operators.

Ethical and legal considerations. This thesis deliberately confines all active interference and spoofing experiments to controlled laboratory environments, using licensed hardware and isolated testbeds. Nevertheless, extending this line of work toward broader validation raises ethical and regulatory challenges. Future research should continue to emphasize safe, reproducible methodologies and, where possible, involve collaboration with industry partners and regulators when moving toward field trials.

Addressing these issues will both validate the practical significance of the vulnerabilities identified in this thesis and inform the development of effective defenses. By combining rigorous experimental methodology with cross-layer remediation strategies, future research can help harden next-generation cellular systems against the class of combined passive-observation and active-manipulation attacks explored here.

Bibliography

[1] 3GPP. *NG-RAN; Architecture description*. Technical Specification (TS) 38.401. 3rd Generation Partnership Project (3GPP), 2025.

[2] 3GPP. *NR; Physical Channels and Modulation*. Technical Specification (TS) 38.211. 18.3.0. 3rd Generation Partnership Project (3GPP), July 2024.

[3] 3GPP. *NR; Multiplexing and Channel Coding*. Technical Specification (TS) 38.212. 18.3.0. 3rd Generation Partnership Project (3GPP), July 2023.

[4] 3GPP. *NR; Physical Layer Procedures for Control*. Technical Specification (TS) 38.213. 18.4.0. 3rd Generation Partnership Project (3GPP), Sept. 2024.

[5] 3GPP. *NR; Physical Layer Procedures for Data*. Technical Specification (TS) 38.214. 18.4.0. 3rd Generation Partnership Project (3GPP), Sept. 2024.

[6] 3GPP. *NR; Radio Resource Control (RRC); Protocol Specification*. Technical Specification (TS) 38.331. 18.3.0. 3rd Generation Partnership Project (3GPP), Sept. 2024.

[7] Vuk Marojevic et al. "Performance Analysis of a Mission-Critical Portable LTE System in Targeted RF Interference". In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE. 2017, pp. 1–6.

[8] Mina Labib et al. "Enhancing the Robustness of LTE Systems: Analysis and Evolution of the Cell Selection Process". In: *IEEE Communications Magazine* 55.2 (2017), pp. 208–215.

[9] Marc Lichtman et al. "LTE/LTE-A Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation". In: *IEEE Communications Magazine* 54.4 (2016), pp. 54–61.

[10] Marc Lichtman et al. "Vulnerability of LTE to Hostile Interference". In: *2013 IEEE Global Conference on Signal and Information Processing*. Ieee. 2013, pp. 285–288.

[11] Andrea Paci et al. "FlashCatch: Minimizing Disruption in IMSI Catcher Operations". In: *18th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. WiSec 2025. Arlington, VA, USA: Association for Computing Machinery, 2025, pp. 124–135. ISBN: 9798400715303. DOI: 10.1145/3734477.3734705. URL: https://doi.org/10.1145/3734477.3734705.

[12] Shinjo Park. "Why We Cannot Win: On Fake Base Stations and Their Detection Methods". PhD thesis. Technische Universität Berlin, 2023.

[13] Ivan Palamà et al. “IMSI Catchers in the Wild: A Real World 4G/5G Assessment”. In: *Computer Networks* 194 (2021), p. 108137.

[14] Syed Rafiul Hussain et al. “Privacy attacks to the 4G and 5G cellular paging protocols using side channel information”. In: *Network and distributed systems security (NDSS) Symp.2019* (2019).

[15] Altaf Shaik et al. “New Vulnerabilities in 4G and 5G Cellular Access Network Protocols: Exposing Device Capabilities”. In: *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. 2019, pp. 221–231.

[16] Simon Erni et al. “AdaptOver: Adaptive Overshadowing Attacks in Cellular Networks”. In: *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. 2022, pp. 743–755.

[17] Hojoon Yang et al. “Hiding in Plain Signal: Physical Signal Overshadowing Attack on {LTE}”. In: *28th USENIX Security Symposium (USENIX Security 19)*. 2019, pp. 55–72.

[18] David Rupprecht et al. “Breaking LTE on Layer Two”. In: *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2019, pp. 1121–1136.

[19] Michal Harvanek et al. “Survey on 5G Physical Layer Security Threats and Countermeasures”. In: *Sensors (Basel, Switzerland)* 24.17 (2024), p. 5523.

[20] Norbert Ludant, Marinos Vomvas, and Guevara Noubir. “Unprotected 4G/5G Control Procedures at Low Layers Considered Dangerous”. In: *arXiv preprint arXiv:2403.06717* (2024).

[21] Mohamad Saalim Wani et al. “Security Vulnerabilities in 5G Non-Stand-Alone Networks: A Systematic Analysis and Attack Taxonomy”. In: *Journal of Cybersecurity and Privacy* 4.1 (2024), pp. 23–40.

[22] Norbert Ludant and Guevara Noubir. “SigUnder: A Stealthy 5G Low Power Attack and Defenses”. In: *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 2021, pp. 250–260.

[23] Youness Arjoune and Saleh Faruque. “Smart Jamming Attacks in 5G New Radio: A Review”. In: *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2020, pp. 1010–1015. DOI: 10.1109/CCWC47524.2020.9031175.

[24] Marc Lichtman et al. “5G NR Jamming, Spoofing, and Sniffing: Threat Assessment and Mitigation”. In: *2018 IEEE international conference on communications workshops (ICC Workshops)*. IEEE. 2018, pp. 1–6.

[25] Olaonipekun Oluwafemi Erunkulu et al. “5G Mobile Communication Applications: A Survey and Comparison of Use Cases”. In: *IEEE Access* 9 (2021), pp. 97251–97295.

[26] Ahmad Alalewi, Iyad Dayoub, and Soumaya Cherkaoui. “On 5G-V2X Use Cases and Enabling Technologies: A Comprehensive Survey”. In: *Ieee Access* 9 (2021), pp. 107710–107737.

[27] Alcardo Alex Barakabitze et al. “5G Network Slicing Using SDN and NFV: A Survey of Taxonomy, Architectures and Future Challenges”. In: *Computer Networks* 167 (2020), p. 106984.

[28] CJ Richards, JC McEachen, and M Tummala. “RNTI Recovery Optimization Utilizing Modified Polar Decoding and Syndrome Matching”. In: *2024 17th International Conference on Signal Processing and Communication System (ICSPCS)*. IEEE. 2024, pp. 1–10.

[29] Norbert Ludant, Pieter Robyns, and Guevara Noubir. “From 5G Sniffing to Harvesting Leakages of Privacy-Preserving Messengers”. In: *2023 IEEE Symp. on Security and Privacy (SP)*. IEEE, 2023, pp. 3146–3161. DOI: 10.1109/SP46215.2023.10179353.

[30] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation*. Technical Specification (TS) 36.211. 18.0.1. 3rd Generation Partnership Project (3GPP), Apr. 2024.

[31] Shijie Luo et al. “Sni5Gect: A Practical Approach to Inject aNRchy into 5G NR”. In: *34th USENIX Security Symposium (USENIX Security 25)*. 2025.

[32] Haoran Wan et al. “NR-Scope: A Practical 5G Standalone Telemetry Tool”. In: *Proceedings of the 20th International Conference on emerging Networking EXperiments and Technologies*. 2024, pp. 73–80.

[33] Keysight. *WaveJudge Wireless Analyzer Solutions*. Accessed on 2025-03-18. 2025. URL: %7Bhttps://www.keysight.com/us/en/products/wireless-analyzers/wavejudge-wireless-analyzer-solutions.html%7D.

[34] Qualcomm. *QXDM Professional™ Tool*. Accessed on 2025-03-18. 2020. URL: https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/80-n9471-1_d_qxdm_professional_tool_quick_start.pdf.

[35] Actix. *Actix Analyzer*. Accessed on 2025-03-18. 2020. URL: %7Bhttps://www.amdocs.com/sites/default/files/2021-07/Actix-Analyzer-Overview-datasheet_2020.pdf%7D.

[36] Infovista. *TEMS™ Investigation | Mobile Network Drive Testing*. Accessed on 2025-03-18. 2025. URL: %7Bhttps://www.infovista.com/products/tems-investigation/network-testing-and-troubleshooting%7D.

[37] thinkRF. *Autonomous & Continuous Spectrum Intelligence Platform*. Accessed on 2025-03-18. 2025. URL: %7Bhttps://thinkrf.com%7D.

[38] Swarun Kumar et al. “LTE Radio Analytics Made Easy and Accessible”. In: *ACM SIGCOMM Computer Communication Review* 44.4 (2014), pp. 211–222.

[39] Xiufeng Xie et al. “piStream: Physical Layer Informed Adaptive Video Streaming over LTE”. In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 2015, pp. 413–425.

[40] Robert Falkenberg, Christoph Ide, and Christian Wietfeld. “Client-Based Control Channel Analysis for Connectivity Estimation in LTE Networks”. In: *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE. 2016, pp. 1–6.

[41] Nicola Bui and Joerg Widmer. “OWL: A Reliable Online Watcher for LTE Control Channel Measurements”. In: *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. 2016, pp. 25–30.

[42] Robert Falkenberg and Christian Wietfeld. “FALCON: An Accurate Real-Time Monitor for Client-Based Mobile Network Data Analytics”. In: *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2019, pp. 1–7.

[43] Martin Kotuliak et al. “LTrack: Stealthy Tracking of Mobile Phones in LTE”. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022, pp. 1291–1306.

[44] Tuan Dinh Hoang et al. “LTESniffer: An Open-Source LTE Downlink/Uplink Eavesdropper”. In: *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 2023, pp. 43–48.

[45] Sangwoo Lee, Liuyi Jin, and Radu Stoleru. *Saflo: eBPF-Based MPTCP Scheduler for Mitigating Traffic Analysis Attacks in Cellular Networks*. 2025. arXiv: 2502.04236 [cs.NI]. URL: https://arxiv.org/abs/2502.04236.

[46] Gunwoo Yoon and Byeongdo Hong. *Scalable and Robust Mobile Activity Fingerprinting via Over-the-Air Control Channel in 5G Networks*. Submitted on 19 September 2024. 2024. arXiv: 2409.12572 [cs.NI].

[47] 3GPP. *NR; Base Station (BS) Radio Transmission and Reception*. Technical report (TR) 38.104. 19.1.0. 3rd Generation Partnership Project (3GPP), June 2025.

[48] J. Laurie Grinstead Charles M.; Snell. *Introduction to Probability*. American Mathematical Society, 1997.

[49] SRS. *Open source O-RAN 5G CU/DU solution from Software Radio Systems (SRS)*. Accessed on 2025-03-18. 2025. URL: %7Bhttps://github.com/srsran/srsRAN_Project%7D.

[50] Sukchan Lee. *Open5GS*. Accessed on 2025-03-18. 2025. URL: https://open5gs.org/.

[51] Rohde & Schwarz. *R&S®FSVA3000 Signal and spectrum analyzer*. 2025. URL: https://www.rohde-schwarz.com/cz/products/test-and-measurement/benchtop-analyzers/rs-fsva3000-signal-and-spectrum-analyzer_63493-601504.html.

[52] *USRP™ N310 Simplifying SDR Deployment*. Ettus Research. 2019. URL: https://www.ettus.com/wp-content/uploads/2019/01/USRP_N310_Datasheet_v3.pdf.

[53] 3GPP. *Study on Channel Model for Frequencies from 0.5 to 100 GHz*. Technical report (TR) 38.901. 19.0.0. 3rd Generation Partnership Project (3GPP), July 2025.

[54] Mohammad Lotfollahi et al. "Deep packet: A novel approach for encrypted traffic classification using deep learning". In: *Soft Computing* 24.3 (2020), pp. 1999–2012.

[55] Giuseppe Aceto et al. "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges". In: *IEEE Trans. on network and service management* 16.2 (2019), pp. 445–458.

[56] Pan Wang et al. "Datanet: Deep learning based encrypted network traffic classification in sdn home gateway". In: *IEEE Access* 6 (2018), pp. 55380–55391.

[57] Ly Vu, Cong Thanh Bui, and Quang Uy Nguyen. "A deep learning based method for handling imbalanced problem in network traffic classification". In: *Proceedings of the 8th Int. Symp. on information and communication technology*. 2017, pp. 333–339.

[58] Shahbaz Rezaei and Xin Liu. "Deep learning for encrypted traffic classification: An overview". In: *IEEE commun. magazine* 57.5 (2019), pp. 76–81.

[59] Ran Dubin et al. "I know what you saw last minute—encrypted http adaptive video streaming title classification". In: *IEEE Trans. on information forensics and security* 12.12 (2017), pp. 3039–3049.

[60] Alberto Dainotti, Antonio Pescape, and Kimberly C Claffy. "Issues and future directions in traffic classification". In: *IEEE network* 26.1 (2012), pp. 35–40.

[61] Shuyuan Zhao, Yongzheng Zhang, and Yafei Sang. "Towards unknown traffic identification via embeddings and deep autoencoders". In: *2019 26th Int. Conf. on Telecommunications (ICT)*. IEEE. 2019, pp. 85–89.

[62] Byeongdo Hong, Sangwook Bae, and Yongdae Kim. "GUTI Reallocation Demystified: Cellular Location Tracking with Changing Temporary Identifier." In: *NDSS*. 2018.

[63] Roger Piqueras Jover. "LTE security, protocol exploits and location tracking experimentation with low-cost software radio". In: *arXiv preprint arXiv:1607.05171* (2016).

[64] Oscar Lasierra et al. "Unmasking 5G Security: Bridging the Gap Between Expectations and Reality". In: *Authorea Preprints* (2024).

[65] Merlin Chlosta et al. "LTE security disabled: misconfiguration in commercial networks". In: *Proc. 12th conf. on security and privacy in wireless and mobile networks (WiSec'19)*. 2019, pp. 261–266.

[66] Andrea Paci, Matteo Chiacchia, and Giuseppe Bianchi. “5GMap: User-Driven Audit of Access Security Configurations in Cellular Networks”. In: *19th Wireless On-Demand Network Systems and Services Conf. (WONS)*. IEEE. 2024, pp. 97–104.

[67] Altaf Shaik et al. “Practical attacks against privacy and availability in 4G/LTE mobile communication systems”. In: *arXiv preprint arXiv:1510.07563* (2015).

[68] Hoang Duy Trinh et al. “Mobile traffic classification through physical control channel fingerprinting: A deep learning approach”. In: *IEEE Trans. on Network and Service Management* 18.2 (2020), pp. 1946–1961.

[69] Geir Hallingstad and Lasse Overlier. “Traffic Flow Confidentiality in a Future Network Enabled Capability Environment”. In: *2007 IEEE SMC Information Assurance and Security Workshop*. 2007, pp. 325–332.

[70] Per Carlén. “Traffic flow confidentiality mechanisms and their impact on traffic”. In: *2013 Military Commun. & Inf. Sys. Conf.* IEEE. 2013.

[71] Csaba Kiraly et al. “Traffic Flow Confidentiality in IPsec: Protocol and Implementation”. In: *The Future of Identity in the Information Society: Proceedings of the Third IFIP WG 9.2, 9.6/11.6, 11.7/FIDIS International Summer School on The Future of Identity in the Information Society, Karlstad University, Sweden, August 4–10, 2007*. Springer. 2008, pp. 311–324.

[72] Chaoyun Zhang, Xi Ouyang, and Paul Patras. “ZipNet-GAN: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network”. In: *13th Int. Conf. on emerging Networking EXperiments and Technologies (CoNext’17)*. 2017, pp. 363–375.

[73] Zhengyang Chen et al. “Automatic mobile application traffic identification by convolutional neural networks”. In: *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE. 2016, pp. 301–307.

[74] George Dean Bissias et al. “Privacy vulnerabilities in encrypted HTTP streams”. In: *Privacy Enhancing Technologies: 5th Int. Workshop, PET 2005, Cavtat, Croatia, May 30-June 1, 2005, Revised Selected Papers 5*. Springer. 2006, pp. 1–11.

[75] Vincent F Taylor et al. “Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic”. In: *2016 IEEE European Symp. on Security and Privacy (EuroS&P)*. IEEE. 2016, pp. 439–454.

[76] Gerard Draper-Gil et al. “Characterization of encrypted and vpn traffic using time-related”. In: *Proceedings of the 2nd Int. Conf. on information systems security and privacy (ICISSP)*. 2016, pp. 407–414.

[77] A. S. Iliyasu and H. Deng. “Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks”. In: *IEEE Access* 8 (2019), pp. 118–126.

[78] Wei Wang et al. “End-to-end encrypted traffic classification with one-dimensional convolution neural networks”. In: *2017 IEEE Int. Conf. on intelligence and security informatics (ISI)*. IEEE. 2017, pp. 43–48.

[79] Giuseppe Aceto et al. “Mobile encrypted traffic classification using deep learning”. In: *2018 Network traffic measurement and analysis Conf. (TMA)*. IEEE. 2018, pp. 1–8.

[80] Shane Miller, Kevin Curran, and Tom Lunney. “Multilayer perceptron neural network for detection of encrypted VPN network traffic”. In: *2018 Int. Conf. on cyber situational awareness, data analytics and assessment (Cyber SA)*. IEEE. 2018, pp. 1–8.

[81] Robert Falkenberg and Christian Wietfeld. “FALCON: An Accurate Real-time Monitor for Client-based Mobile Network Data Analytics”. In: *2019 IEEE Global Communications Conf. (GLOBECOM)*. Waikoloa, Hawaii, USA: IEEE, Dec. 2019. DOI: 10.1109/GLOBECOM38437.2019.9014096. arXiv: 1907.10110. URL: https://arxiv.org/abs/1907.10110.

[82] Hongyi Yao et al. “Samples: Self adaptive mining of persistent lexical snippets for classifying mobile application traffic”. In: *Proceedings of the 21st Annual Int. Conf. on Mobile Computing and Networking*. 2015, pp. 439–451.

[83] Ashwin Rao et al. “Using the middle to meddle with mobile”. In: *CCIS, Dec* (2013).

[84] Ghazi Al-Naymat et al. “CLASSIFICATION OF VOIP AND NON-VOIP TRAFFIC USING MACHINE LEARNING APPROACHES.” In: *J. of Theoretical & Applied Information Technology* (2016).

[85] Hasan Faik Alan and Jasleen Kaur. “Can Android applications be identified using only TCP/IP headers of their launch time traffic?” In: *9th ACM Conf. on security & privacy in wireless and mobile networks*. 2016, pp. 61–66.

[86] Manuel Lopez-Martin et al. “Network traffic classifier with convolutional and recurrent neural networks for Internet of Things”. In: *IEEE access* 5 (2017), pp. 18042–18050.

[87] Zafar Ayyub Qazi et al. “Application-awareness in SDN”. In: *Proceedings of the ACM SIGCOMM 2013 Conf. on SIGCOMM*. 2013, pp. 487–488.

[88] Sophon Mongkolluksamee, Vasaka Visoottiviseth, and Kensuke Fukuda. “Enhancing the performance of mobile traffic identification with communication patterns”. In: *2015 IEEE 39Th annual computer software and applications Conf.* Vol. 2. IEEE. 2015, pp. 336–345.

[89] Jing Wang et al. “Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach”. In: *IEEE INFOCOM 2017*. IEEE. 2017.

[90] Jie Feng et al. “Deeptp: An end-to-end neural network for mobile cellular traffic prediction”. In: *IEEE Network* 32.6 (2018), pp. 108–115.

[91] Xu Wang et al. “Spatio-temporal analysis and prediction of cellular traffic in metropolis”. In: *IEEE Trans. on Mobile Computing* 18.9 (2018), pp. 2190–2202.

[92] Chaoyun Zhang and Paul Patras. “Long-term mobile traffic forecasting using deep spatio-temporal neural networks”. In: *Proceedings of the Eighteenth ACM Int. Symp. on Mobile Ad Hoc Networking and Computing*. 2018, pp. 231–240.

[93] Xiao Fei, Philippe Martins, and Jialiang Lu. “Real-time Traffic Classification for 5G NSA Encrypted Data Flows With Physical Channel Records”. In: *2023 IEEE 98th Vehicular Technology Conf. (VTC2023-Fall)*. IEEE. 2023, pp. 1–6.

[94] Gunwoo Yoon and Byeongdo Hong. “Scalable and Robust Mobile Activity Fingerprinting via Over-the-Air Control Channel in 5G Networks”. In: *arXiv preprint arXiv:2409.12572* (2024).

[95] Xiaorui Wu and Chunling Wu. “CLPREM: A real-time traffic prediction method for 5G mobile network”. In: *Plos one* 19.4 (2024), e0288296.

[96] Andrii A Astrakhantsev et al. “Adjusting the parameters of machine learning algorithms to improve the speed and accuracy of traffic classification”. In: *Information and Telecommunication Sciences* (2023).

[97] Robert Pell et al. “Service Classification of Network Traffic in 5G Core Networks using Machine Learning”. In: *2023 IEEE Int. Conf. on Edge Computing and Communications (EDGE)*. IEEE. 2023, pp. 309–318.

[98] Mingrui Fan et al. “Traffic Fingerprints for Homogeneous IoT Traffic Based on Packet Payload Transition Patterns”. In: *Electronics* 13.5 (2024), p. 930.

[99] Md Ruman Islam et al. “Characterizing Encrypted Application Traffic through Cellular Radio Interface Protocol”. In: *2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*. IEEE. 2024, pp. 321–329.

[100] Google. *MonkeyRunner*. 2019. URL: https://developer.android.com/studio/test/monkeyrunner/.

[101] The Tcpdump Group. *Tcpdump*. https://www.tcpdump.org/. 2019.

[102] Citrix. *SSL Interception*. https://docs.citrix.com/en-us/netscaler-secure-web-gateway/12/ssl-interception.html.

[103] RedIRIS. *RedIRIS*. https://www.rediris.es/.

[104] Gianni Barlacchi et al. “A multi-source dataset of urban life in the city of Milan and the Province of Trentino”. In: *Scientific data* 2.1 (2015), pp. 1–15.

[105] Juan Sebastián Rojas. *IP Network Traffic Flows Labeled with 75 Apps*. URL: https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps.

[106] NetMate. *NetMate: A Network Traffic Analyzer*. https://www.netmate.org/.

[107] SRS. *srsRAN 4G*. URL: https://www.srsran.com/4g.

[108] Erik G. Larsson et al. "Massive MIMO for next generation wireless systems". In: *IEEE Communications Magazine* 52.2 (2014), pp. 186–195. DOI: 10.1109/MCOM.2014.6736761.

[109] Olakunle Elijah et al. "A Comprehensive Survey of Pilot Contamination in Massive MIMO—5G System". In: *IEEE Communications Surveys & Tutorials* 18.2 (2016), pp. 905–923. DOI: 10.1109/COMST.2015.2504379.

[110] David López-Pérez et al. "A Survey on 5G Radio Access Network Energy Efficiency: Massive MIMO, Lean Carrier Design, Sleep Modes, and Machine Learning". In: *IEEE Communications Surveys & Tutorials* 24.1 (2022), pp. 653–697. DOI: 10.1109/COMST.2022.3142532.

[111] Poonam Lohan et al. "From 5G to 6G Networks: A Survey on AI-Based Jamming and Interference Detection and Mitigation". In: *IEEE Open Journal of the Communications Society* 5 (2024), pp. 3920–3974. DOI: 10.1109/OJCOMS.2024.3416808.

[112] Abhishek Pagadala and Gulrej Ahmed. "Analysis of DDoS Attacks in 5G Networks". In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 2023, pp. 1–6. DOI: 10.1109/ICCCNT56998.2023.10307311.

[113] Pei Huang et al. "IS-WARS: Intelligent and Stealthy Adversarial Attack to Wi-Fi-Based Human Activity Recognition Systems". In: *IEEE Transactions on Dependable and Secure Computing* 19.6 (2022), pp. 3899–3912. DOI: 10.1109/TDSC.2021.3110480.

[114] Ning Gao et al. "Physical layer authentication under intelligent spoofing in wireless sensor networks". In: *Signal Processing* 166 (2020), p. 107272. ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2019.107272. URL: https://www.sciencedirect.com/science/article/pii/S0165168419303263.

[115] Myeongsu Han et al. "OFDM Channel Estimation With Jammed Pilot Detector Under Narrow-Band Jamming". In: *IEEE Transactions on Vehicular Technology* 57.3 (2008), pp. 1934–1939. DOI: 10.1109/TVT.2007.907314.

[116] Giang Quynh Le Vu, Hung Tran, and Kien Trung Truong. "Jammer Detection by Random Pilots in Massive MIMO Spatially-uncorrelated Rician Channels". In: *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*. 2021, pp. 440–445. DOI: 10.1109/NICS54270.2021.9701559.

[117] Chowdhury Shahriar, Robert McGwier, and T. Charles Clancy. "Performance impact of pilot tone randomization to mitigate OFDM jamming attacks". In: *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. 2013, pp. 813–816. DOI: 10.1109/CCNC.2013.6488553.

[118] Na Chen et al. “Scalable and flexible massive MIMO precoding for 5G H-CRAN”. In: *IEEE Wireless Communications* 24.1 (2017), pp. 46–52.

[119] Junyoung Nam, Giuseppe Caire, and Jeongseok Ha. “On the Role of Transmit Correlation Diversity in Multiuser MIMO Systems”. In: *IEEE Transactions on Information Theory* 63.1 (2017), pp. 336–354. DOI: 10.1109/TIT.2016.2615627.

[120] Yongpeng Wu et al. “Secure Massive MIMO Transmission With an Active Eavesdropper”. In: *IEEE Transactions on Information Theory* 62.7 (2016), pp. 3880–3900.

[121] Mohammad Amin Sheikhi and S. Mohammad Razavizadeh. “Security Vulnerability of FDD Massive MIMO Systems in Downlink Training Phase”. In: *2018 9th International Symposium on Telecommunications (IST)*. 2018, pp. 492–496. DOI: 10.1109/ISTEL.2018.8661082.

[122] Hessam Pirzadeh, S. Mohammad Razavizadeh, and Emil Björnson. “Subverting Massive MIMO by Smart Jamming”. In: *IEEE Wireless Communications Letters* 5.1 (2016), pp. 20–23. DOI: 10.1109/LWC.2015.2487960.

[123] Yu-Chih Tung et al. “Vulnerability and protection of channel state information in multiuser MIMO networks”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 775–786.

[124] Wei Xi et al. “Prevent CSI spoofing in uplink MU-MIMO transmission”. In: *Proceedings of the 1st Workshop on Context Sensing and Activity Recognition*. 2015, pp. 13–18.

[125] Qian Liu et al. “Disrupting MIMO communications with optimal jamming signal design”. In: *IEEE transactions on wireless communications* 14.10 (2015), pp. 5313–5325.

[126] Shiguo Wang et al. “Pilot spoofing detection for massive MIMO mmWave communication systems with a cooperative relay”. In: *Computer Communications* 202 (2023), pp. 33–41. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2023.02.014. URL: https://www.sciencedirect.com/science/article/pii/S0140366423000518.

[127] Avner Elgam, Yael Balal, and Yosef Pinhasi. “Study of 5G-NR-MIMO Links in the Presence of an Interferer”. In: *Electronics* 10.6 (2021), p. 732.

[128] Delong Liu, Wei Wang, and Yang Huang. “Pilot spoofing attack detection and channel estimation for secure massive MIMO”. In: *Electronics Letters* 60.21 (2024), e70074.

[129] Ti Ti Nguyen and Kim-Khoa Nguyen. “Pilot-partitioning protocol and anti-jamming methods in distributed massive MIMO systems”. In: *IEEE Transactions on Cognitive Communications and Networking* 9.5 (2023), pp. 1211–1225.

[130] Muhammad Karam Shehzad, Luca Rose, and Mohamad Assaad. “A novel algorithm to report CSI in MIMO-based wireless networks”. In: *ICC 2021-IEEE International Conference on Communications*. IEEE. 2021, pp. 1–6.

[131] *USRP™ B200/B210 Bus Series*. Ettus Research. URL: https://www.ettus.com/wp-content/uploads/2019/01/b200-b210%5C_spec%5C_sheet.pdf.

[132] Danilo Valerio. “Open source software-defined radio: A survey on gnuradio and its applications”. In: *Forschungszentrum Telekommunikation Wien, Vienna, Technical Report FTW-TR-2008-002* (2008).

[133] https://www.sharetechnote.com/html/5G/5G_FR_Bandwidth.html.

[134] *RF Agile Tranceiver AD9361*. Analog Device. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ad9361.pdf.

[135] 3GPP. *User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone*. Technical Specification (TS) 38.300. 18.2.0. 3rd Generation Partnership Project (3GPP), July 2025.

[136] Umar Danjuma Maiwada et al. “Energy Efficiency in 5G Systems: A systematic literature review”. In: *International Journal of Knowledge-Based and Intelligent Engineering Systems* (2024).

[137] Xiaowei Zhang, Andreas Kunz, and Stefan Schröder. “Overview of 5G Security in 3GPP”. In: *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. 2017.

[138] Xinsheng Ji et al. “Overview of 5G Security Technology”. In: *Science China Information Sciences* 61.8 (2018), p. 081301.

[139] Ning Wang et al. “Physical-Layer Security of 5G Wireless Networks for IoT: Challenges and Opportunities”. In: *IEEE Internet of Things Journal* (2019).

[140] Gaurav Soni and Kamlesh Chandravanshi. “Security Scheme to Identify Malicious Maneuver of Flooding Attack for WSN in 6G”. In: *2021 8th International Conference on Signal Processing and Integrated Networks*. 2021. DOI: 10.1109/SPIN52536.2021.9566066.

[141] Meenu Rani Dey et al. “Early Detection of Battery Depletion Attack in 5G-based UAV Networks”. In: *2024 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. 2024. DOI: 10.1109/ANTS63515.2024.10898930.

[142] Raman Batra, Varalakshmi S, and Shashikant Patil. “Enhancement of 5G N/W System for the use of ML Algorithm Based Ticket-Reopening System for the use of Attack Prediction”. In: *2024 1st International Conference on Innovative Sustainable Technologies for Energy, Mechatronics, and Smart Systems (ISTEMS)*. 2024, pp. 1–5. DOI: 10.1109/ISTEMS60181.2024.10560321.