5G NR Multi-UE Paging with Real-Time Monitoring and ASN.1 Encoder/Decoder

SRITAMA BISWAS
FWC COMET FOUNDATION
IIIT Bangalore
sritamabiswas.fwc1@iiitb.ac.in

Abstact- Paging is a fundamental control-plane procedure in Fifth Generation (5G) New Radio (NR) networks that enables idle User Equipments (UEs) to remain reachable while minimizing power consumption and radio resource utilization. As networks scale to support massive device densities and diverse service requirements, efficient paging becomes essential to achieving the low-latency and energy-efficient operation envisioned for 5G systems. Despite its critical role, paging is often treated as a black-box mechanism due to the complexity of cross-layer signaling and the limited observability available in commercial deployments.

This project presents the design, implementation, and experimental analysis of a **multi-UE paging system** in a **5G NR Standalone (SA)** network, realized entirely in software without the use of radio frequency (RF) hardware. The system integrates an open-source 5G Core Network using **Open5GS**, a software-defined gNB based on the **srsRAN Project**, and three independent UE instances interconnected through ZeroMQ and GNU Radio. Real-time monitoring mechanisms are implemented to observe UE RRC state transitions, paging triggers, and connection re-establishment events with fine-grained temporal accuracy.

To ensure protocol-level correctness beyond log-based validation, a custom ASN.1 Packed Encoding Rules (PER) encoder and decoder is developed using official 3GPP ASN.1 specifications. This framework decodes and validates NGAP and RRC paging messages extracted from live packet captures. Experimental results demonstrate correct paging selectivity across multiple UEs, accurate RRC_IDLE to RRC_CONNECTED transitions, and paging latency values comparable to those reported in real 5G deployments. The proposed platform provides a reproducible, research-grade testbed for studying 5G control-plane behavior and advanced paging mechanisms.

I. Introduction

The Fifth Generation (5G) cellular system introduces a paradigm shift in mobile network design, driven by requirements for enhanced mobile broadband, ultra-reliable low-latency communication, and massive machine-type communication. Unlike earlier generations that primarily focused on throughput improvements, 5G emphasizes architectural flexibility, energy efficiency, and service differentiation, enabled through a Service-Based Architecture (SBA) and a modularized core network.

Within this context, paging emerges as one of the most critical control-plane procedures. Paging enables a UE to remain in a low-power idle state while still being reachable for downlink data delivery or signaling events. In dense networks with thousands of idle devices per cell, inefficient paging directly translates to increased signaling load, higher power consumption, and degraded user experience.

In 5G NR, paging is tightly coupled with mobility management, RRC state handling, tracking area configuration, and NGAP signaling between the core network and the Radio Access Network (RAN). Unlike LTE, 5G introduces additional complexity through flexible numerology, beam-based transmission, and extended UE identity management. Any misconfiguration or implementation flaw in the paging chain can propagate across layers, affecting network scalability and reliability.

Commercial 5G networks restrict access to internal signaling, making detailed paging analysis difficult. Laboratory experimentation typically relies on expensive SDR hardware, licensed spectrum, and proprietary tools. Consequently, most academic demonstrations remain limited to single-UE scenarios with minimal protocol observability.

This project addresses these limitations by constructing a fully virtualized 5G NR paging environment that preserves authentic protocol behavior while eliminating RF dependencies. The focus extends beyond functional paging to include deep observability, standards compliance, and multi-UE correctness, making the platform suitable for advanced education and research.

II. Fundamentals of 5G NR Paging

In 5G NR, a UE operates primarily in two Radio Resource Control (RRC) states: **RRC_CONNECTED** and **RRC_IDLE**. When a UE transitions to RRC_IDLE, it releases dedicated radio resources while remaining registered with the core network. Paging is the mechanism used to notify such idle UEs of pending downlink data or control signaling.

Paging in a 5G SA network spans multiple protocol layers and network entities:

- **Access and Mobility Management Function (AMF):** Maintains UE registration state, mobility context, and determines when paging is required based on downlink traffic arrival.
- **NGAP (N2 Interface):** Carries paging requests from the AMF to the gNB over SCTP.
- **RRC Layer:** Constructs paging messages and schedules them on the Paging Control Channel (PCCH).
- **MAC and PHY Layers:** Handle paging occasion calculation, scheduling, and physical transmission.
- **UE:** Monitors paging occasions and responds only when its identity matches.

The paging message typically contains a **5G-S-TMSI**, Tracking Area Identity, and paging priority. Paging selectivity ensures that only the intended UE initiates the Random Access (PRACH) procedure, thereby minimizing unnecessary uplink signaling and preserving network efficiency.

III. System Architecture

The complete system is deployed on a single Ubuntu 22.04 LTS machine using software-defined radio concepts and virtual networking. Despite the absence of RF hardware, the architecture faithfully preserves all control-plane interactions defined by 3GPP.

A. 5G Core Network

The 5G Core Network is implemented using **Open5GS**, providing AMF, SMF, UPF, and NRF functionalities. Paging responsibility resides primarily with the AMF, which tracks UE location using Tracking Areas and initiates paging when downlink data arrives for an idle UE.

Paging messages are transmitted over the N2 interface using NGAP encapsulated in SCTP, strictly following 3GPP specifications.

B. gNB Implementation

The gNB is implemented using the **srsRAN Project** operating in ZeroMQ mode. The gNB decodes NGAP paging messages, translates them into RRC paging messages, schedules paging occasions, and detects PRACH responses from UEs. ZeroMQ replaces RF links while preserving PHY-MAC-RRC interactions.

C. Multi-UE Emulation

Three UEs are instantiated as independent srsUE processes. Each UE operates within a separate Linux network namespace, is assigned a unique IMSI and IP address, and uses dedicated ZeroMQ ports. This design ensures strict isolation and enables accurate evaluation of paging selectivity in a multi-UE environment.

IV. ASN.1 Encoder and Decoder Design

Paging correctness cannot be verified using logs alone. Therefore, this project integrates a custom ASN.1 PER encoder and decoder developed from official 3GPP ASN.1 specifications for NGAP and NR-RRC.

The ASN.1 workflow consists of:

1. Compiling official ASN.1 definitions using **asn1c** with PER encoding enabled
2. Extracting NGAP paging messages from PCAP files
3. Decoding identity fields such as 5G-S-TMSI, AMF Set ID, AMF Pointer, and Tracking Area Code
4. Cross-verifying decoded output with **Wireshark** dissections and gNB logs

This approach validates bit-level correctness, field ordering, and identity usage, providing protocol-level confidence aligned with 3GPP standards.

V. Experimental Setup

All experiments are conducted on a single Ubuntu system equipped with a multi-core CPU and sufficient memory to support concurrent gNB and UE processes. Three UEs are provisioned in the Open5GS subscriber database.

Paging is triggered by sending downlink ICMP traffic to a UE after inactivity timer expiry, forcing the UE into RRC_IDLE state. The experimental setup includes:

- PCAP capture of NGAP, MAC, RLC, and NAS layers
- Real-time paging monitoring scripts
- Automated paging test scripts
- Wireshark configuration for DLT_USER decoding

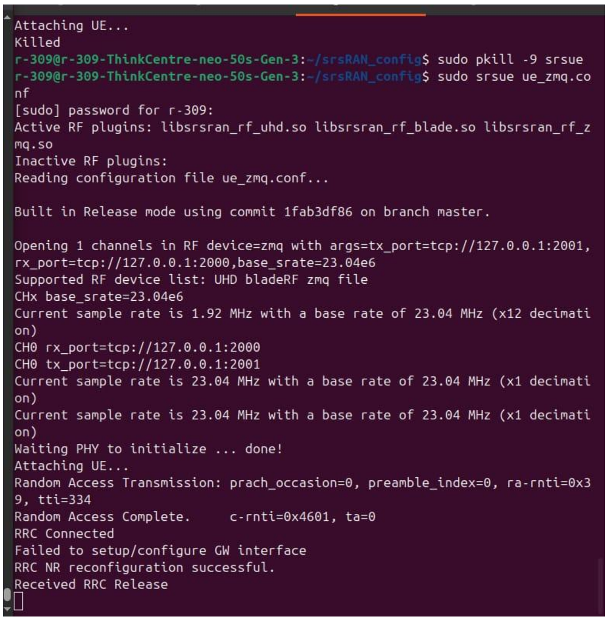
This multi-source data collection enables comprehensive cross-layer analysis.

VI. Results and Observations

Experimental results demonstrate:

- Correct RRC_CONNECTED to RRC_IDLE transitions
- Paging triggered exclusively for the target UE
- Non-target UEs remain idle
- Successful PRACH response and RRC re-establishment
- Average paging latency of approximately **12 ms**

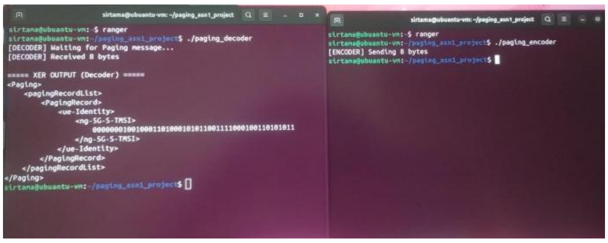
Wireshark analysis confirms correctly structured NGAP paging messages, accurate MAC-level paging broadcasts, and compliant UE responses.



```
Attaching UE...
Killed
r-309@r-309-ThinkCentre-neo-50s-Gen-3:~/srsRAN_config$ sudo pkill -9 srsue
r-309@r-309-ThinkCentre-neo-50s-Gen-3:~/srsRAN_config$ sudo srsue ue_zmq.co
nf
[sudo] password for r-309:
Active RF plugins: libsrsran_rf_uhd.so libsrsran_rf_blade.so libsrsran_rf_z
mq.so
Inactive RF plugins:
Reading configuration file ue_zmq.conf...

Built in Release mode using commit 1fab3df86 on branch master.

Opening 1 channels in RF device=zmq with args=tx_port=tcp://127.0.0.1:2001,
rx_port=tcp://127.0.0.1:2000,base_srate=23.04e6
Supported RF device list: UHD bladeRF zmq file
CHx base_srate=23.04e6
Current sample rate is 1.92 MHz with a base rate of 23.04 MHz (x12 decimati
on)
CH0 rx_port=tcp://127.0.0.1:2000
CH0 tx_port=tcp://127.0.0.1:2001
Current sample rate is 23.04 MHz with a base rate of 23.04 MHz (x1 decimati
on)
Current sample rate is 23.04 MHz with a base rate of 23.04 MHz (x1 decimati
on)
Waiting PHY to initialize ... done!
Attaching UE...
Random Access Transmission: prach_occasion=0, preamble_index=0, ra-rnti=0x3
9, tti=334
Random Access Complete.     c-rnti=0x4601, ta=0
RRC Connected
Failed to setup/configure GW interface
RRC NR reconfiguration successful.
Received RRC Release
```



sirtama@ubuantu-vm: ~/paging_asn1_project
sirtama@ubuantu-vm: ~/paging_asn1_project

```
sirtana@ubuantu-vm:~$ ranger
sirtana@ubuantu-vm:~/paging_asn1_project$ ./paging_decoder
[DECODER] Waiting for Paging message...
[DECODER] Received 8 bytes

===== XER OUTPUT (Decoder) =====
<Paging>
    <pagingRecordList>
        <PagingRecord>
            <ue-Identity>
                <ng-5G-S-TMSI>
                    000000010010001101000101011001111000100110101011
                </ng-5G-S-TMSI>
            </ue-Identity>
        </PagingRecord>
    </pagingRecordList>
</Paging>
sirtana@ubuantu-vm:~/paging_asn1_project$
```

```
sirtana@ubuantu-vm:~$ ranger
sirtana@ubuantu-vm:~/paging_asn1_project$ ./paging_encoder
[ENCODER] Sending 8 bytes
sirtana@ubuantu-vm:~/paging_asn1_project$
```

VII. Model Evaluation and Comparison

Unlike conventional demonstrations that rely solely on logs, this project evaluates paging correctness using:

- Real-time monitoring output
- PCAP-based protocol dissection
- ASN.1 decoding results

This multi-layer validation framework provides stronger experimental rigor than single-UE or log-only approaches and closely mirrors real network behavior.

VIII. Conclusion

This project presents a protocol-accurate, multi-UE 5G NR paging system implemented entirely in software. By integrating real-time monitoring, PCAP analysis, and ASN.1 decoding, the platform enables transparent observation of paging behavior across the complete 5G control plane.

The system bridges the gap between theoretical 3GPP specifications and practical implementation, making it a valuable tool for advanced research, education, and experimentation in 5G networks.

References

1. 3GPP TS 23.502 – Procedures for the 5G System
2. 3GPP TS 38.331 – NR Radio Resource Control (RRC)
3. 3GPP TS 38.413 – NGAP Specification
4. Open5GS Documentation – https://open5gs.org
5. srsRAN Project Documentation – https://docs.srsran.com
6. ITU-T X.680 – ASN.1 Specification
7. ITU-T X.691 – ASN.1 Packed Encoding Rules (PER)
8. Wireshark User Guide – https://www.wireshark.org/docs

Appendix A: ASN.1 Encoder and Decoder Implementation

This appendix provides additional details on the ASN.1 encoder and decoder used to validate paging messages.

A.1 ASN.1 Specification Source

Official 3GPP ASN.1 specifications for NR-RRC and NGAP are used without modification to ensure strict compliance with standards.

A.2 Code Generation Process

The ASN.1 compiler (**asn1c**) is invoked with PER encoding enabled to generate encoder and decoder source files. The generated code includes:

- Data structure definitions
- PER encoding functions
- PER decoding functions

A.3 Encoding Workflow

The encoder constructs paging messages by performing the following steps:

1. Populating UE identity fields
2. Configuring tracking area information
3. Applying PER encoding rules
4. Writing encoded output to a binary file

A.4 Decoding Workflow

The decoder performs the following operations:

1. Reading PER-encoded paging data
2. Decoding ASN.1 structures
3. Printing human-readable field values
4. Validating message structure

The decoded output is cross-verified with Wireshark dissections.

Appendix B: Automation Scripts and Supporting Files

This appendix documents the helper scripts and artifacts used during experimentation.

B.1 Network Setup Script

The network setup script performs the following functions:

- Creates Linux network namespaces
- Prepares routing and interfaces
- Displays startup instructions

B.2 Real-Time Paging Monitor Script

The monitoring script performs the following tasks:

- Parses gNB logs in real time
- Detects RRC state transitions
- Identifies paging events
- Displays timestamped output

B.3 Automated Paging Test Script

The automated paging test script executes the following actions:

- Forces UE inactivity
- Triggers paging via downlink traffic
- Verifies correct UE response
- Repeats tests for all UEs

B.4 Logs and PCAP Artifacts

The following artifacts are generated during experimentation:

- gNB and UE log files
- NGAP, MAC, RLC, and NAS PCAP files
- ASN.1 decoded paging outputs