

5G Network Simulation with Secure Syslog Integration

Midterm Presentation

Rishabh Kumar (cs25resch04002)

WNS

November 2025

Outline

- 1 Introduction
- 2 5G Network Simulation
- 3 UE Location Service
- 4 Secure Syslog Integration
- 5 Results & Analysis
- 6 Conclusion & Future Work

- **Goal:** Implement a comprehensive 5G network simulation with secure logging infrastructure
- **Key Components:**
 - 5G Standalone (SA) network using OpenAirInterface
 - UE Location Service for tracking devices
 - Secure syslog integration with TLS encryption
- **Use Cases:**
 - Network monitoring and analysis
 - Security audit and compliance
 - Real-time location tracking

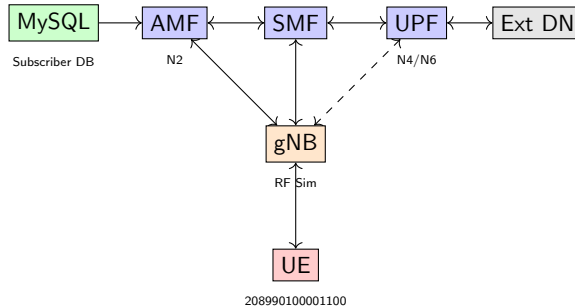
Challenges in 5G Networks:

- Complex architecture
- Security concerns
- Log management at scale
- Real-time monitoring needs

Our Solution:

- Simulated 5G environment
- Centralized secure logging
- Location tracking service
- Automated analysis tools

5G Architecture



Control Plane (Public Net)

- Subnet: 192.168.71.128/26
- MySQL: .131
- AMF: .132
- SMF: .133
- UPF: .134
- gNB: .140
- UE: .150

User Plane (Traffic Net)

- Subnet: 192.168.72.128/26
- UPF: .134
- External DN: .135
- Protocols: GTP-U

Technologies: Docker, OpenAirInterface v2.1.10, RF Simulator

Deployment Process

```
1 # Create Docker networks
2 docker network create --subnet=192.168.71.128/26 \
3   rfsim5g-oai-public-net
4 docker network create --subnet=192.168.72.128/26 \
5   rfsim5g-oai-traffic-net
6
7 # Deploy 5G Core
8 docker-compose up -d mysql oai-amf oai-smf \
9   oai-upf oai-ext-dn
10
11 # Deploy RAN
12 docker-compose up -d oai-gnb
13
14 # Deploy UE
15 docker-compose up -d oai-nr-ue
```

Listing 1: Network Setup Commands

Verification Results

gNB Connection Status

Status: Connected | Global Id: 0x0E00 | gNB Name: gnb-rfsim

UE Registration

5GMM-REGISTERED | IMSI: 208990100001100 | Cell: 0000e014e

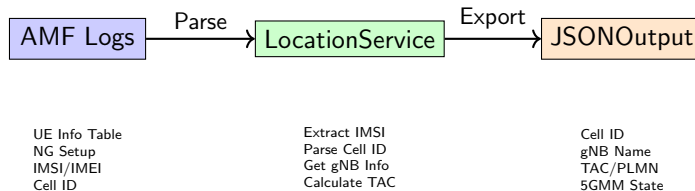
Connectivity Test

5 packets transmitted, 5 received, 0% packet loss

Achievement

✓ Full 5G SA network operational with end-to-end connectivity

Location Service Architecture



Location Service Implementation

```
1 class UELocationService:
2     def get_amf_logs(self):
3         """Fetch AMF container logs"""
4         cmd = ["docker", "logs", "rfsim5g-oai-amf"]
5         result = subprocess.run(cmd, capture_output=True)
6         return result.stdout.decode('utf-8')
7
8     def get_ue_location_by_imsi(self, imsi):
9         """Extract UE location from AMF logs"""
10        logs = self.get_amf_logs()
11        # Parse UE info table
12        ue_info = self.parse_ue_info_table(logs, imsi)
13        # Parse gNB information
14        gnb_info = self.parse_ng_setup_request(logs)
15        return {
16            'ue_identity': ue_info,
17            'network_location': {...},
18            'gnb_info': gnb_info
19        }
```

UE Identity:

- IMSI: 208990100001100
- GUTI
- RAN UE NGAP ID
- AMF UE NGAP ID

Network Location:

- Cell ID: 0000e014e
- TAC: 00 00 01
- PLMN: MCC=208, MNC=99

gNB Information:

- gNB ID: 0x0E00
- gNB Name: gnb-rfsim
- Status: Connected

Registration State:

- 5GMM State: REGISTERED
- UE IP: 12.1.1.2

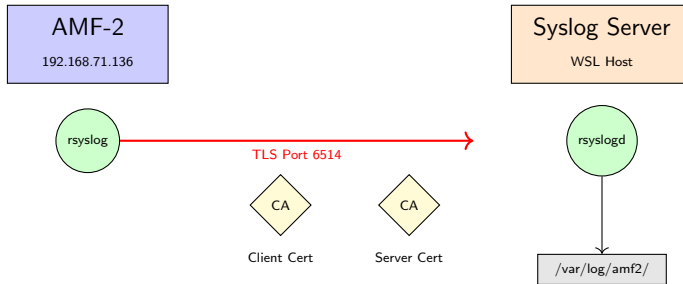
Note: All data extracted from real AMF logs, no mock data used

Usage Examples

```
1 # Get location by IMSI
2 python3 ue_location_service.py --imsi 208990100001100
3
4 # Get location by IMEI
5 python3 ue_location_service.py --imei 862104052096703
6
7 # Get all UE locations
8 python3 ue_location_service.py --all
9
10 # Export to JSON
11 python3 ue_location_service.py --imsi 208990100001100 \
12     --export ue_location.json
```

Listing 3: Location Service Commands

Secure Syslog Architecture



TLS Certificate Chain

- **CA Certificate:** Self-signed root CA (CN=Syslog CA)
- **Server Certificate:** CN=syslog-server
- **Client Certificate:** CN=amf-2
- **Key Size:** RSA 4096-bit
- **Validity:** 365 days

Authentication

- x509/name authentication mode
- Mutual TLS verification
- Certificate CN validation
- Only authorized clients (CN=amf-2) permitted

Rsyslog Configuration

```
1 # Load TLS module
2 module(load="imtcp"
3         StreamDriver.Name="gtls"
4         StreamDriver.Mode="1"
5         StreamDriver.Authmode="x509/name")
6
7 # Global TLS settings
8 global(
9     DefaultNetstreamDriver="gtls"
10    DefaultNetstreamDriverCAFile="/path/to/ca-cert.pem"
11    DefaultNetstreamDriverCertFile="/path/to/server-cert.pem"
12    DefaultNetstreamDriverKeyFile="/path/to/server-key.pem"
13 )
14
15 # Accept connections on port 6514
16 input(type="imtcp" port="6514" ruleset="amf2"
17        StreamDriver.Name="gtls"
18        PermittedPeer=["amf-2"])
```

Listing 4: Syslog Server Config (/etc/rsyslog.d/99-amf-secure.conf)

Container Configuration:

- Image: oai-amf:v2.1.10
- IP: 192.168.71.136
- AMF Set ID: 002
- N2 Port: 38413

Certificates Mounted:

- CA certificate
- Client certificate
- Client private key

Log Forwarding:

- Target: WSL host
- Port: 6514
- Protocol: TCP over TLS
- Format: RFC 5424

Custom Entrypoint:

- Install rsyslog
- Configure TLS
- Start rsyslogd
- Launch AMF

Verification Results

Syslog Server Status

rsyslogd listening on 0.0.0.0:6514 (TLS enabled)

AMF-2 Connection

TLS handshake successful | Peer verified: CN=amf-2

Log Reception

/var/log/amf2/amf2-2025-11-10.log created

566 bytes received from 192.168.71.136

Achievement

✓ Secure syslog operational with TLS encryption and authentication

Implementation Summary

Component	Status
5G Core (MySQL, AMF, SMF, UPF)	✓ Running
gNB (RF Simulator)	✓ Connected
UE (RF Simulator)	✓ Registered
End-to-End Connectivity	✓ Verified
UE Location Service	✓ Operational
Secure Syslog Server	✓ Running
AMF-2 with Syslog	✓ Forwarding
TLS Authentication	✓ Verified

Table: System Component Status

Key Achievements

① 5G Network Simulation

- Deployed complete 5G SA network using OpenAirInterface
- Successfully registered UE with 0% packet loss
- Verified all protocols (NGAP, PFCP, GTP-U, SCTP)

② UE Location Service

- Built Python service to extract location from AMF logs
- Support for IMSI/IMEI-based queries
- JSON export functionality
- Real data only, no mock coordinates

③ Secure Syslog Integration

- Deployed AMF-2 with TLS-encrypted log forwarding
- Implemented x509 certificate authentication
- Verified secure log transmission and storage

Technical Challenges & Solutions

Challenges:

- 1 Docker compose compatibility
- 2 IMEI extraction from logs
- 3 TLS certificate validation
- 4 Container networking

Solutions:

- 1 Modified YAML config
- 2 Regex pattern matching
- 3 Proper CN configuration
- 4 External network setup

Lesson Learned

Thorough log analysis and proper certificate management are crucial for secure 5G deployments

Project Deliverables

- ✓ **Working 5G network** with OAI (7 containers)
- ✓ **UE Location Service** (Python script with full documentation)
- ✓ **Secure Syslog Server** with TLS encryption
- ✓ **AMF-2 Deployment** with log forwarding
- ✓ **Documentation:** README, setup guides, API docs
- ✓ **Git Repository:** 4 commits with version control

Repository: github.com/username/WNS (to be pushed)

Lines of Code: 1500+ (Python, YAML, Shell, Config)

Future Enhancements

① Advanced Monitoring

- Real-time dashboard (Grafana/Kibana)
- ML-based anomaly detection
- Automated alerting system

② Scalability

- Multi-UE scenarios (10+ devices)
- Load balancing across AMFs
- Kubernetes deployment

③ Security

- SIEM integration
- Threat detection rules
- Compliance reporting

④ Geographic Features

- Cell database integration
- Real GPS coordinates
- Location visualization

References

-  3GPP TS 23.501: System Architecture for the 5G System
-  OpenAirInterface Software Alliance <https://openairinterface.org/>
-  RFC 5424: The Syslog Protocol
-  RFC 5425: Transport Layer Security (TLS) Transport Mapping for Syslog
-  Docker Documentation <https://docs.docker.com/>

Thank You!

Questions?

Rishabh Kumar (cs25resch04002)
kumarrishabh73@gmail.com