



Vertiefungsprojekt

“Implementieren und Testen eines 5G-Core-Netzes mit Free5GC”

Eingereicht durch:	Thomas Zielasny
Matrikelnummer:	1177217
Studiengang:	Elektro- und Informationstechnik
Spezialisierung:	Information and Communication Technology
Hochschule:	Frankfurt University of Applied Sciences
Betreuer:	Prof. Dr.- Ing. Ulrich Trick
Zeitraum:	04.05.2022 – 28.06.2023 (Sommersemester 23)

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig verfasst und keine als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellenachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.



28.06.2023, Thomas Zielasny

Inhaltsverzeichnis

1	Einleitung	4
2	Theoretischer Hintergrund	5
2.1	Verwendete Technologien	5
2.1.1	VirtualBox und Ubuntu 20.04 (Server)	5
2.1.2	Go	5
2.1.3	MongoDB	5
2.1.4	T-Shark	6
2.1.5	Free5GC	6
2.1.6	UERANSIM	6
2.2	5G-System	6
2.2.1	Netzwerkfunktionen	6
2.2.2	Zugangsnetz	8
2.2.3	Benutzerequipment	8
2.3	Protokolle	8
2.3.1	Packet Forwarding Control Protocol (PFCP)	8
2.3.2	GPRS Tunneling Protocol (GTP)	9
3	Anforderungsanalyse	10
3.1	Generelle Zielsetzung	10
3.2	Klärung der Anforderungen	10
3.3	Angestrebtes Ziel	10
3.4	Anwendungsfälle	11
4	Realisierung	12
4.1	Aufsetzen der virtuellen Maschinen in VirtualBox	12
4.2	Installation von Free5GC	13
4.3	Testen der Kernnetzfunktionen	15
4.4	Vorbereitung der virtuellen Maschine für UERANSIM	16
4.5	Installation von UERANSIM	17
4.6	Registrierung des Benutzerequipments	18
4.7	Testen der Funktionalität	20
5	Zusammenfassung und Ausblick	24
6	Abkürzungsverzeichnis	25
7	Abbildungsverzeichnis	27
8	Referenzen	28
9	Anhang	30

1 Einleitung

Der im Rahmen des Vertiefungsprojekts erstellte Bericht befasst sich mit *Free5GC*. *Free5GC* ist eine Open-Source-Implementierung eines 5G-Core-Netzwerks, das auf dem 3GPP-Standard basiert. Wobei das 5G für die fünfte Generation der Mobilfunktechnologie steht und eine erhebliche Verbesserung der drahtlosen Kommunikation mit hoher Bandbreite, geringer Latenzzeit und Unterstützung einer großen Anzahl gleichzeitiger Verbindungen ermöglicht. Der *5G-Core* bildet das Herzstück des 5G-Netzwerks und verwaltet und koordiniert verschiedene Dienste und Ressourcen. Durch *Free5GC* wird Forschern und Entwicklern eine Plattform zur Verfügung gestellt, auf der sie neue Ideen, Protokolle und Techniken in Zusammenhang mit den 5G-Core-Netzwerken implementieren und testen können. Es bietet eine vollständige Implementierung der Funktionen und Protokolle, die in einem 5G-Core-Netzwerk erforderlich sind, welche durch den Benutzer anpassbar und erweiterbar sind. Es können unter anderem neue Routing- und Signalisierungsalgorithmen entwickelt werden, Netzwerkfunktionen virtualisiert und skaliert werden und Sicherheitslösungen getestet werden.

Dieser Bericht bietet eine Einführung in das Free5GC-Framework, beschreibt seine Architektur und Funktionen und diskutiert mögliche Anwendungsbereiche und Forschungsfragen. Außerdem wird die Implementierung des *Free5GC* in Verbindung mit dem *Radio-Access-Network (RAN)* und *User-Equipment (UE)* simuliert. Durch die Nutzung und Möglichkeiten, die diese Plattform bietet, können Wissenschaftler und Forscher einen wertvollen Beitrag zur Weiterentwicklung der 5G-Technologie leisten.

2 Theoretischer Hintergrund

Im theoretischen Hintergrund werden die verwendeten Technologien vorgestellt. Außerdem werden essenzielle 5G-Systemelemente vorgestellt und spezielle Protokolle, die im Zusammenhang mit der Technologie verwendet werden, näher beleuchtet.

2.1 Verwendete Technologien

Die verwendeten Technologien beschreiben Programmiersprachen, die im Rahmen der Projektarbeit verwendet wurden, Datenbanken zur Speicherung der Daten und andere Softwarelösungen, die verwendet wurden.

2.1.1 VirtualBox und Ubuntu 20.04 (Server)

VirtualBox ist eine Virtualisierungssoftware, die es Benutzern ermöglicht, mehrere Betriebssysteme auf einem Host-Rechner zu installieren und parallel auszuführen. Die von *Oracle* bereitgestellte Software erlaubt es, sogenannte virtuelle Maschinen (VMs) zu erstellen und zu konfigurieren, auf denen Betriebssysteme wie *Windows*, *Linux*, *macOS* und andere aufgesetzt werden können. Die Technologie basiert auf der Hardware-Virtualisierungsmethode, die in modernen Prozessoren vorhanden ist. Dadurch kann der Computer verschiedene Betriebssysteme gleichzeitig ausführen, die in isolierten virtuellen Umgebungen, den virtuellen Maschinen laufen. Außerdem wird eine Netzwerkkommunikation zwischen der Host-Maschine und den virtuellen Maschinen unterstützt. (vgl. Oracle o.D.)

Das Betriebssystem *Ubuntu 20.04 (Server)* ist eine spezielle Version von *Ubuntu*, das für den Einsatz als Server optimiert ist. Es eignet sich besonders als Plattform für den Betrieb von Webservern, da es bereits den *Apache http Server*, *Nginx* und andere Webserver-Software enthält, die Webanwendungen hosten können. Außerdem werden von dem Betriebssystem verschiedene Datenbankserver wie *MySQL*, *PostgreSQL* und *MongoDB* unterstützt. (vgl. Kurek, T. 2019)

2.1.2 Go

Die kompilierte Programmiersprache *Go* ist sehr leistungsstark und verbraucht wenig Speicherplatz, bietet eine native Unterstützung für Multithreading und bringt außerdem eine umfangreiche Standardbibliothek mit, daher eignet sie sich besonders für den Einsatz als *API-Server*. In diesem Zusammenhang bietet sie zusätzlich den Vorteil einer hohen Skalierbarkeit in einer Produktionsumgebung, wodurch komplexe Prozesse effizient parallel ausgeführt werden können. (vgl. Google o.D.) Sie bildet die Basis für die Software *Free5GC* und bearbeitet sämtliche Anfragen in dieser 5G-Kernumgebung.

2.1.3 MongoDB

Die dokumentenorientierte NoSQL-Datenbank, die entwickelt wurde, um skalierbare und flexible Datenlösungen zu unterstützen, bietet im Gegensatz zu relationalen Datenbanken eine Datenorganisation in JSON-ähnlichen Dokumenten, was eine einfache Handhabung von strukturierten

und unstrukturierten Daten ermöglicht. Als Datenbankbasis für *Free5GC* werden Funktionen wie Persistenz von Konfigurations- und Nutzerdaten, Zustandsverwaltung, Skalierbarkeit und Echtzeitdatenverarbeitung unterstützt. (vgl. MongoDB Inc. o.D.)

2.1.4 T-Shark

T-Shark ist ein Terminalprogramm und Teil des *Wireshark*-Pakets und wird für die Netzwerkprotokollanalyse verwendet. (vgl. Ross, J. 2020) Dadurch sollen die Netzwerkschnittstellen auf den Ubuntu-Servern überwacht, gefiltert und analysiert werden, um mögliche Fehler aufzudecken und ein besseres Verständnis für die Funktionalität des Kernnetzes zu bekommen.

2.1.5 Free5GC

Free5GC ist eine Open-Source-Implementierung eines 5G-Kernnetzes und stellt Kernfunktionen für die Verwaltung und den Betrieb des 5G-Kernnetzes bereit. Dazu gehören Funktionen wie Zugangskontrolle, Sitzungsverwaltung, Benutzer- und Steuerungsebene, Authentifizierung und Autorisierung. Als Open-Source-Implementierung eignet es sich besonders für die Erforschung der Funktionen und Protokolle und das Testen neuer Funktionalitäten. *Free5GC* ist in *Go* geschrieben und verwendet *MongoDB* als Datenbank zur Verwaltung von Daten. *Free5GC* stellt den Kern eines 5G-Netzes dar und muss mit anderen Komponenten wie Basisstation, Funkzugangnetzen und Endgeräten zusammenarbeiten, um ein vollständiges 5G-Netzwerk abzubilden. (vgl. Free5GC o.D.)

2.1.6 UERANSIM

Die Open-Source-Software emuliert einen virtuellen *5G UE (User Equipment)* und wird häufig in Kombination mit *Free5GC* zu Test- und Entwicklungszwecken eingesetzt. Es ermöglicht Entwicklern und Forschern, virtuelle 5G-Benutzergeräte zu erstellen und mit einem 5G-Netzwerk zu verbinden. Dadurch ist es möglich verschiedene Anwendungsfälle, Netzwerkparameter und Protokolle zu testen, ohne auf physische Geräte angewiesen zu sein. Die Nutzung der Software in Verbindung mit *Free5GC* bietet besonders in Anwendungsfällen, in denen eine hohe Anzahl an Endgeräten benötigt wird, viele Möglichkeiten, da eine hohe Skalierbarkeit durch Virtualisierung möglich ist. (vgl. Aligungr 2021)

2.2 5G-System

Das hier vorgestellte 5G-System besteht aus mehreren Komponenten, die zusammenarbeiten und untereinander vernetzt sind. Dazu zählt das *User-Equipment (UE)*, die *Basisstationen (gNodeB)* und das *Kernnetzwerk*, welches das zentrale Steuerungssystem des 5G-Netzwerks darstellt. Darüber hinaus können weitere Komponenten wie eine Cloud-Infrastruktur und Netzwerkfunktionen wie *VNFs* eingebunden werden. (vgl. Trick, U. o.D.)

2.2.1 Netzwerkfunktionen

Die Netzwerkfunktionen beziehen sich auf virtualisierte Funktionen, die im 5G-Netzwerk implementiert werden und laufen in der Regel auf virtualisierten Servern oder in Cloud-Infrastrukturen.

Dazu zählen Netzwerkfunktionen wie:

- Access Mobility Management Function (AMF)
- Session Management Function (SMF)
- User Plane Function (UPF)
- Authentication Server Function (AUSF)
- Unified Data Management (UDM)
- Policy Control Function (PCF)
- Network Slice Selection Function (NSSF)
- Application Function (AF)

Die *AMF* ist für die Verwaltung des Zugriffs und der Mobilität der Endgeräte im 5G-Netzwerk zuständig, dazu zählen Authentifizierung im Netzwerk, Autorisierung des Zugriffs auf die Dienste. Außerdem verwaltet die *AMF* den Zustand und die Position der Endgeräte, einschließlich des *Handovers* zwischen verschiedenen Basisstationen, um eine kontinuierliche Konnektivität während der Bewegung sicherzustellen. Die *AMF* ist auch für die Einrichtung, Aktualisierung und Beendigung von Verbindungen zwischen den Endgeräten und dem Netzwerk verantwortlich. (vgl. Dryanski, M. 2018)

Die *SMF* ist hingegen für die Verwaltung der Kommunikationssitzungen im 5G-Netzwerk zuständig. Darunter fällt auch die Verwaltung der *QoS-Parameter* (*Quality of Service*) und legt fest, welche Priorität, Bandbreite und andere Parameter für verschiedene Kommunikationssitzungen zugewiesen werden sollen. Außerdem ist die *SMF* auch für die Zuweisung und Verwaltung von IP-Adressen verantwortlich. (vgl. Dryanski, M. 2018)

Die *UPF* ist für die Verarbeitung des Datenverkehrs in der Benutzerebene (*User Plane*) zuständig. Dazu gehören die Paketbearbeitung und Weiterleitung der Pakete. Außerdem kann die *UPF* Daten puffern, um sicherzustellen, dass sie in der richtigen Reihenfolge und mit den richtigen Prioritäten verarbeitet werden. Zudem implementiert die *UPF* die festgelegten *QoS-Parameter* und stellt damit sicher, dass der Datenverkehr die vereinbarten Dienstgüteanforderungen erfüllt. (vgl. Dryanski, M. 2018)

Die *AUSF* stellt Sicherheitsfunktionen wie Verschlüsselung und Integritätsschutz zur Verfügung. Außerdem werden Schlüssel für die Authentifizierung von der Funktion verwaltet. (vgl. Dryanski, M. 2018)

Die *UDM* speichert und verwaltet die Benutzerprofile, Abonnementinformationen, Authentifizierungsdaten und relevante Benutzerinformationen. Zudem können auch Abrechnungs- und Zahlungsinformationen im Zusammenhang mit den 5G-Diensten verwaltet werden. Die *UDM* stellt sicher, dass die Benutzerdaten konsistent, korrekt und sicher gespeichert und verwaltet werden. (vgl. Dryanski, M. 2018)

Die *PCF* ist für die Steuerung und Verwaltung der Richtlinien im 5G-Netzwerk zuständig. Dazu zählen die Steuerung der *QoS-Parameter*, Zugriffskontrollen für Netzwerk und Dienste und *Networkslicing*, um dedizierte virtuelle Netzwerke für spezifische Dienste oder Anwendungen bereitzustellen. (vgl. Dryanski, M. 2018)

Die *Networkslices* werden von der *NSSF* verwaltet und zugewiesen, dazu zählen die benötigten Ressourcen, Eigenschaften und *QoS-Anforderungen*. (vgl. Dryanski, M. 2018)

Die *AF* repräsentiert spezifische Anwendungen oder Dienste im 5G-Netzwerk. Außerdem interagiert die Netzwerkfunktion mit anderen Netzwerkfunktionen, um erforderliche Netzwerkressourcen und Dienste für die Anwendung bereitzustellen. (vgl. Dryanski, M. 2018)

2.2.2 Zugangsnetz

Das Zugangsnetz, auch als *RAN (Radio Access Network)* bezeichnet, ist ein wesentlicher Bestandteil eines Mobilfunknetzes und stellt die Verbindung zwischen den Endgeräten und dem Kernnetzwerk her. Es ist für die drahtlose Übertragung von Daten, Sprache und anderen Kommunikationsdiensten verantwortlich. Es ermöglicht den Aufbau einer drahtlosen Verbindung zwischen dem Endgerät und der *Basisstation (gNB)*. Die Basisstation ist Teil des Zugangsnetzes und dient als Schnittstelle zwischen dem Endgerät und dem Kernnetzwerk. (vgl. Sultan, A. 2022) Um eine effiziente Übertragung zwischen dem Endgerät und der Basisstation zu gewährleisten, kodiert, moduliert und demoduliert es Signale. Außerdem werden hierüber die Kapazitäten des Funkkanals verwaltet und die Ressourcenallokation optimiert. Dazu zählt die Zuweisung von Frequenzbändern, Verwaltung von Übertragungszeitplänen und Interferenzkontrolle. Das *Handover* zwischen verschiedenen Basisstationen, zum Beispiel bei Bewegung des Endgeräts von einer in eine andere Funkzelle wird ebenfalls vom Zugangsnetz geregelt. Schließlich implementiert das Zugangsnetz Sicherheitsmechanismen wie Authentifizierung, Verschlüsselung und Integritätsschutz, um die Vertraulichkeit und Integrität der übertragenen Daten zu gewährleisten. (vgl. Sultan, A. 2022)

2.2.3 Benutzerequipment

Das *User Equipment (UE)* bezieht sich auf die Endgeräte in einem Mobilfunknetzwerk. Es ist die Hardware, die von Benutzern verwendet wird, um eine drahtlose Kommunikation herzustellen und auf Netzwerkdienste zuzugreifen. Das *UE* ist für den Zugriff auf das Mobilfunknetzwerk verantwortlich, indem es Anmeldeinformationen sendet, um sich im Netzwerk zu authentifizieren und bekommt als Antwort eine Netzwerkadresse zugewiesen, um Dienste nutzen zu können. Das *UE* kann verschiedene Formen wie z.B. Smartphones, Tablets, Laptops, *IoT*-Geräte und andere drahtlose Endgeräte annehmen. (vgl. 3GPP 2021)

2.3 Protokolle

Im 5G-Netzwerk werden mehrere Protokolle eingesetzt, um die Kommunikation und den Austausch von Daten und Steuerungsinformationen zwischen den Netzwerkkomponenten und -funktion zu ermöglichen.

2.3.1 Packet Forwarding Control Protocol (PFCP)

Das *PFCP-Protokoll* basiert auf dem IP-Protokoll und wird über *UDP (User Datagram Protocol)* transportiert. Es verwendet eine client-serverbasierte Architektur, bei der die *SMF* als Client und die *UPF* als Server fungiert.

Das Protokoll wird dazu verwendet, die Steuerung des Paketflusses und die Verwaltung der Ressourcen zwischen der *SMF (Session Management Function)* und der *UPF (User Plane Function)* zu ermöglichen. Das Protokoll ermöglicht unter anderem die Einrichtung, Änderung und Beendigung von Sitzungen zwischen der *SMF* und der *UPF*, sowie die Verwaltung von Ressourcen für den Paketfluss, einschließlich Bandbreite und *QoS-Parameter*. Außerdem kann das *PFCP-Protokoll* dazu verwendet werden, Fehler oder Abweichungen im Paketfluss zu erkennen. (vgl. Devopedia 2023)

Die Abbildung 1 zeigt eine Beispielregistrierung initiiert von der *SMF* an die *UPF*. Dabei wird zuerst wie in Nachricht (1) eine Assoziation eingeleitet, welche dann von der *UPF* bestätigt wird (2). Die

Assoziation ermöglicht die Steuerung und den Austausch von Informationen über den Paketfluss und die Ressourcenverwaltung zwischen den beiden Funktionen. Als nächstes wird eine Nachricht (3) von der *SMF* an die *UPF* gesendet, um eine neue Sitzung zu etablieren, welche im Anschluss bestätigt wird (4). Schließlich wird eine Nachricht gesendet, welche die Modifizierung der Sitzung ermöglicht (5). Die Modifizierung wird von der *UPF* entsprechend bestätigt (6). Die hier beschriebenen Nachrichtentypen stellen nur ein Teil der Gesamtmenge an Typen dar.

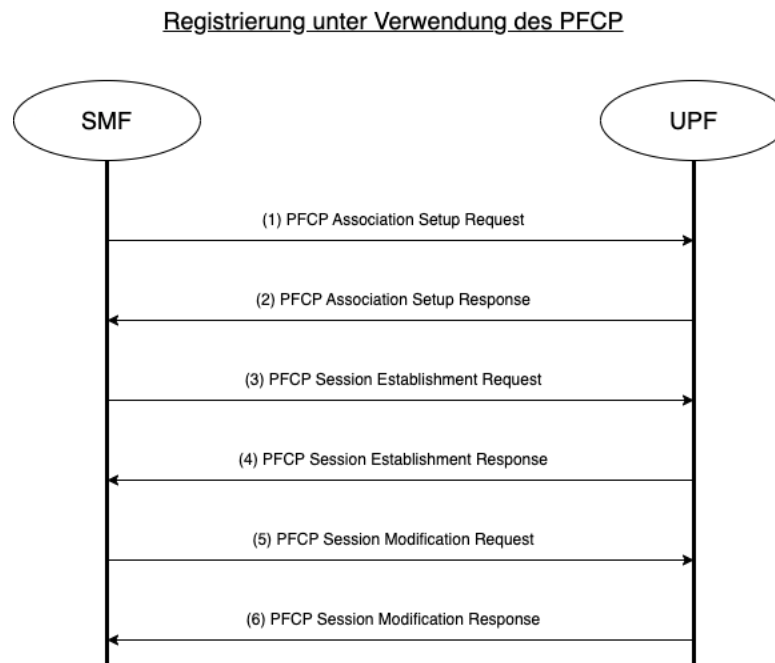


Abbildung 1: Registrierungsvorgang SMF

2.3.2 GPRS Tunneling Protocol (GTP)

Dieses Protokoll wird eingesetzt, um in Mobilfunknetzen und in diesem Fall dem 5G-Netzwerk ein Tunneling von Nutzdatenpaketen zwischen dem Mobilfunkkernnetzwerk und dem Mobilfunkzugangszugangnetzwerk zu ermöglichen. Es ermöglicht die Übertragung von IP-Datenpaketen zwischen verschiedenen Netzwerkelementen und wird sowohl für die *Uplink*- (vom Zugangszugangnetzwerk zum Kernnetzwerk) als auch für die *Downlink-Richtung* (vom Kernnetzwerk zum Zugangszugangnetzwerk) verwendet. (vgl. Devopedia 2023)

3 Anforderungsanalyse

Die Anforderungsanalyse wird die allgemeine Zielsetzung, die speziellen Anforderungen, das angestrebte Ziel und die potenziellen Einsatzmöglichkeiten der Anwendung beleuchten.

3.1 Generelle Zielsetzung

Im Rahmen dieser Arbeit werden zum einen Voraussetzungen, die zur Implementierung der *Free5GC*-Software notwendig sind und zum anderen die Open-Source-Software selbst näher beschrieben. Außerdem wird im weiteren Verlauf das *RAN* und die Registrierung eines *UE* mittels *UERANSIM* simuliert. Dadurch sollen die Basisfunktionen des Core-Netzwerks getestet und simuliert werden.

Die *Free5GC*-Implementierung soll Benutzern ermöglichen, die Funktionalität des 5G-Core-Netzwerks zu testen und später zur Entwicklung verwenden zu können. Deshalb soll zuerst eine Umgebung geschaffen werden, in der die *Free5GC*-Software implementiert werden kann. Im Anschluss sollen die Kernfunktionen des 5G-Core-Netzwerks getestet werden. Dazu zählt die *Registrierung*, *GUTI-Registrierung*, *Service-Request*, *Xn-Handover*, *Deregistration*, *PDU-Session-Release-Request*, *Paging*, *N2Handover*, *Non3GPP*, *ReSynchronization* und *ULCL*. Unter der Verwendung des Open-Source-Simulators wird dann ein *5G-User-Equipment (UE)* emuliert. Durch die Kombination zwischen *Free5GC* und *UERANSIM* soll eine umfassende Test- und Entwicklungsplattform für 5G-Netzwerke entstehen, in der Entwickler die Interaktionen und das Verhalten von *5G User Equipment* und dem 5G-Core-Netzwerk in einer simulierten Umgebung untersuchen können.

3.2 Klärung der Anforderungen

Die Entwicklungsumgebung soll mittels virtueller Maschinen in *VirtualBox* aufgesetzt werden. Dazu sollen *Ubuntu Server* erstellt werden, auf denen die Software *Free5GC* installiert und konfiguriert wird. Um die erfolgreiche Installation zu verifizieren wird der Server gestartet und mögliche Fehler behoben. Im Anschluss sollen die integrierten Tests der Software ausgeführt werden, um einen Überblick über den Funktionsumfang der Open-Source-Software zu bekommen und um die erfolgreiche Installation der Software zu gewährleisten.

Im Anschluss soll über einen Klon des Ursprungsservers die Software *UERANSIM* auf einer zweiten virtuellen Maschine installiert und so konfiguriert werden, dass sie in Verbindung mit *Free5GC* genutzt werden kann. Dafür werden die Konfigurations-Dateien beider Seiten entsprechen angepasst und eine Kommunikation zu ermöglichen. Im nächsten Schritt soll das *RAN* gestartet werden und ein *UE* auf der Weboberfläche von *Free5GC* registriert werden.

3.3 Angestrebtes Ziel

Durch die Installation von *Free5GC* in Kombination mit der Software *UERANSIM* soll eine Plattform für Entwickler und Forscher entstehen, die es ermöglicht den Funktionsumfang des 5G-Core-Netzwerks und des *Radio Access Networks* zu simulieren und testen zu können. Durch die Simulation des virtuellen

5G-Netzwerks sollen neue Funktion entwickelt und getestet werden können, wodurch eine Bewertung bezüglich der Leistung des Netzwerks durch die Implementation der neuen Funktionen vorgenommen werden soll. Außerdem soll die Entwicklungsumgebung dazu verwendet werden, um verschiedene Konfigurationen zu testen, um die optimale Leistung des Netzwerks zu erreichen.

3.4 Anwendungsfälle

Durch das Aufsetzen und die Simulation des *Free5GC* ergeben sich eine Vielzahl von Anwendungsfällen.

Entwickler und Forscher können mit *Free5GC* und *UERANSIM* neue Netzwerkfunktionen und -protokolle entwerfen, testen und optimieren. Sie können verschiedene Szenarien simulieren, um die Leistung des 5G-Netzwerks zu bewerten und Verbesserungen vorzunehmen. Es können neue Algorithmen zur Verbindungsoptimierung entwickelt und das Netzwerk unter hoher Last analysiert werden.

Durch die Verwendung dieser Entwicklungsumgebung können auch Unternehmen ihre 5G-Anwendungen in einer simulierten Umgebung testen, ehe sie in einer Produktionsumgebung eingesetzt werden. Dadurch können sie sicherstellen, dass ihre Anwendungen reibungslos mit dem 5G-Netzwerk interagieren und ihre Leistung erbringen.

Außerdem bietet die Kombination eine ideale Umgebung für Schulungs- und Ausbildungszwecke im Bereich 5G. Es bietet die Möglichkeiten, dass Studenten, Forscher und Fachleute praktische Erfahrung sammeln können, indem sie verschiedene Szenarien simulieren können.

Integrationstests und Kompatibilitätstests bieten zudem die Möglichkeit, dass vor der Einführung neuer 5G-Komponenten oder -Geräte über *Free5GC* und *UERANSIM* sichergestellt werden kann, dass die Geräte nahtlos mit dem 5G-Netzwerk arbeiten und die erforderlichen Standards erfüllt werden.

Die Anwendungsfälle sind nur einige Beispiele für das Potenzial der Nutzung von *Free5GC* in Verbindung mit *UERANSIM*. Durch diese Kombination wird eine flexible und kostengünstige Lösung für die Entwicklung, das Testen und die Erforschung von 5G-Netzwerken geboten.

4 Realisierung

In diesem Abschnitt wird die Implementierung der in den vorherigen Kapiteln vorgestellten Technologien durchgeführt. Dazu wird ein Rechner benötigt, auf dem *VirtualBox* installiert werden muss. Für die Installation von *VirtualBox* wird der offiziellen Anleitung folge geleistet. Außerdem wird die entsprechende ISO-Datei mit der *Ubuntu 20.04 (Server)* Distribution von der offiziellen Internetseite heruntergeladen. (vgl. Free5GC o.D.)

Nach jedem Unterkapitel werden kleinere Tests durchgeführt, die die erfolgreiche Installation und Implementierung der jeweiligen Technologie gewährleisten. Dazu wird zusätzlich das Programm *Wireshark* auf dem Rechner installiert, sodass die jeweiligen Mitschnitte eingesehen und analysiert werden können.

4.1 Aufsetzen der virtuellen Maschinen in VirtualBox

Zur Vorbereitung für die Implementierung von *Free5GC* wird eine virtuelle Maschine gemäß der Anleitung von *Free5GC* konfiguriert und gestartet. Zu beachten ist, dass eine zweite Netzwerkschnittstelle hinzugefügt wird, die als ‚*host-only*‘ neben dem bereits bestehenden ‚*NAT*‘ konfiguriert wird. Außerdem wird bei der Installation des Betriebssystems der SSH-Server mitinstalliert. Das ermöglicht später die Verbindung via *SSH*. (vgl. Free5GC o.D.)

Netzwerkschnittstelle (Typ)	IP-Adresse
enp0s3 (NAT)	10.0.2.15
enp0s8 (host-only)	192.168.56.101

Um die Netzwerkverbindungen zu testen, wird über ein Powershell-Terminal eine *SSH*-Verbindung über die ‚*host-only*‘-Schnittstelle aufgebaut und nach erfolgreichem Anmelden ein *Ping* an ‚*google.com*‘ abgesetzt, um die Verbindung der ‚*NAT*‘-Schnittstelle zum Internet zu überprüfen. (siehe Abbildung 2)

```
ssh 192.168.56.101 -l ubuntu
ping google.com
```

```

ubuntu@free5gc:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::a00:27ff:fe01:2ac3 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:e1:2a:c3 txqueuelen 1000 (Ethernet)
    RX packets 625865 bytes 896510756 (896.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 257754 bytes 16246541 (16.2 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe01:a05f prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:a1:a0:5f txqueuelen 1000 (Ethernet)
    RX packets 26686 bytes 1921336 (1.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 38948 bytes 9131701 (9.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 99716 bytes 13685912 (13.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 99716 bytes 13685912 (13.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu@free5gc:~$ ping google.com
PING google.com (142.251.36.238) 56(84) bytes of data:
64 bytes from muc11s22-in-f14.1e100.net (142.251.36.238): icmp_seq=1 ttl=118 time=15.2 ms
64 bytes from muc11s22-in-f14.1e100.net (142.251.36.238): icmp_seq=2 ttl=118 time=15.0 ms
64 bytes from muc11s22-in-f14.1e100.net (142.251.36.238): icmp_seq=3 ttl=118 time=15.5 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2022ms
rtt min/avg/max/mdev = 14.952/15.199/15.455/0.205 ms

```

Abbildung 2: Netzwerkkonfiguration Free5GC

Danach wird außerdem das Programm *t-shark* installiert, um auf Terminalebene Verkehr über die Netzwerkschnittstellen mitzuschneiden. Diese können dann später unter Verwendung von *SCP* auf den Host-Rechner kopiert und mit *Wireshark* analysiert werden. (vgl. Ross, J. 2019)

Der aktuelle Stand der virtuellen Maschine wird nun als Snapshot gespeichert und dient nun als Ausgangspunkt für die Klonen für *Free5GC* und *UERANSIM*.

4.2 Installation von Free5GC

Bevor *Free5GC* installiert wird, wird auf der Basis der angelegten VM ein Klon erstellt und gestartet. (siehe Abbildung 3)

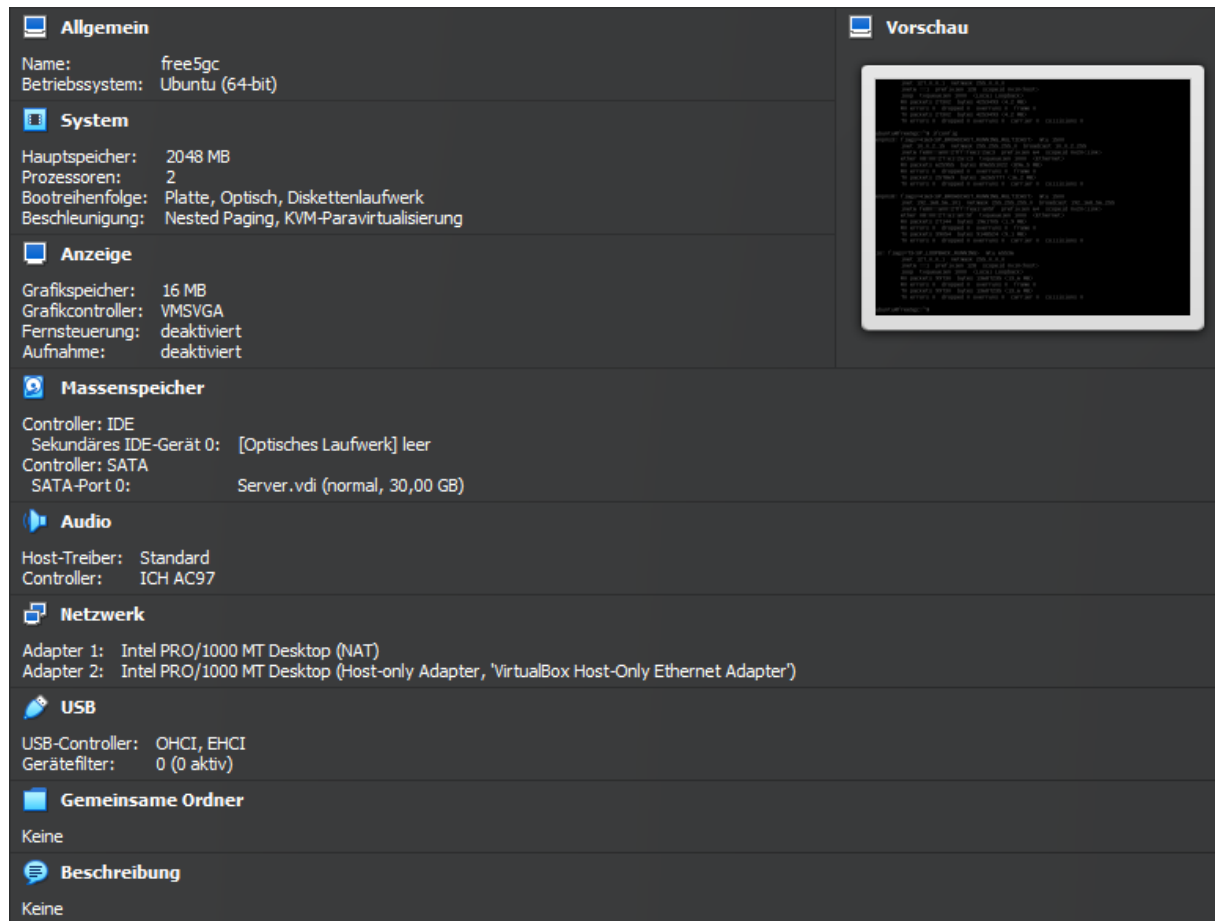


Abbildung 3: Konfiguration VM für Free5GC

Für die Installation wird den Schritten der offiziellen Dokumentation von *Free5GC* gefolgt. Dazu werden vorab alle Abhängigkeiten wie *Go*, *MongoDB*, *Git* und *wget* installiert. Im Anschluss wird *MongoDB* als Service gestartet. Im nächsten Schritt werden alle Control-Plane-Elemente installiert und alle von *Go* benötigten Pakete bereitgestellt.

```
go mod init
go mod tidy
```

Danach können die Control-Plane-Elemente gestartet werden und die User-Plane-Function installiert werden. Zu beachten ist, dass die richtige Kernelversion (v5.0.0-23-generic oder v5.4.x) vorhanden ist. Zum Schluss wird die WebConsole (siehe Abbildung 4) installiert und mit folgendem Befehl gestartet. (vgl. *Free5GC* o.D.)

```
go run server.go
```

Diese kann dann vom Host-Rechner über folgende Adresse erreicht werden:

<http://192.168.56.101:5000>

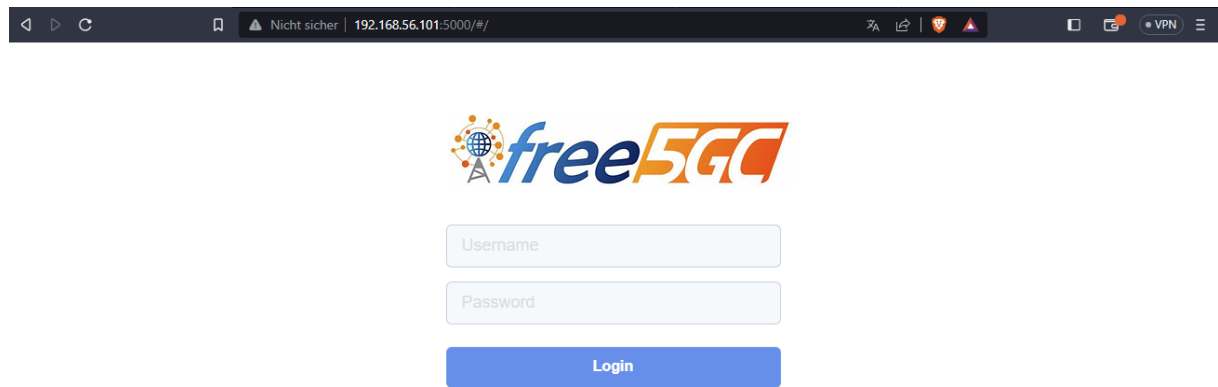


Abbildung 4: WebConsole

Nun kann mit den Tests, die *Free5GC* in der Pfadstruktur bereitstellt gestartet werden, um alle Funktionen zu überprüfen.

4.3 Testen der Kernnetzfunktionen

Zum Testen der Funktionen werden die im *Free5GC*-Verzeichnis vorgegeben Tests aus dem Testscript verwendet, wie in der offiziellen Dokumentation von *Free5GC* beschrieben. Zuerst wird jedoch die *UPF* installiert. (vgl. *Free5GC* o.D.) Die entsprechenden Wireshark-Mittschnitte befinden sich im Anhang.

```
make upf
```

Test	Ergebnis
Registration (siehe Anhang 1)	<pre>--- PASS: TestRegistration (9.43s) PASS ok test 11.097s</pre> <p>Abbildung 5: Test Registration</p>
GUTI-Registration (siehe Anhang 2)	<pre>--- PASS: TestGUTIRegistration (9.53s) PASS ok test 11.177s</pre> <p>Abbildung 6: Test GUTI-Registration</p>
Service-Request (siehe Anhang 3)	<pre>--- PASS: TestServiceRequest (9.67s) PASS ok test 11.320s</pre> <p>Abbildung 7: Test Service-Request</p>
Xn-Handover (siehe Anhang 4)	<pre>--- PASS: TestXnHandover (9.49s) PASS ok test 11.156s</pre> <p>Abbildung 8: Test Xn-Handover</p>
Deregistration (siehe Anhang 5)	<pre>--- PASS: TestDeregistration (8.28s) PASS ok test 9.965s</pre> <p>Abbildung 9: Test Deregistration</p>

PDU-Session-Release-Request
(siehe Anhang 6)

```
--- PASS: TestPDUSessionReleaseRequest (10.44s)
PASS
ok      test    12.092s
```

Abbildung 10: Test PDU-Session-Release-Request

Paging
(siehe Anhang 7)

```
--- PASS: TestPaging (11.71s)
PASS
ok      test    13.398s
```

Abbildung 11: Test Paging

N2-Handover
(siehe Anhang 8)

```
--- PASS: TestPaging (11.71s)
PASS
ok      test    13.398s
```

Abbildung 12: Test N2-Handover

Non-3GPP
(siehe Anhang 9)

```
--- PASS: TestNon3GPPUE (9.68s)
PASS
ok      test    9.713s
```

Abbildung 13: Test Non-3GPP

Resynchronization
(siehe Anhang 10)

```
--- PASS: TestReSynchronization (9.55s)
PASS
ok      test    11.230s
```

Abbildung 14: Test Resynchronization

Request-Two-PDU-Sessions
(siehe Anhang 11)

```
--- PASS: TestRequestTwoPDUSessions (11.63s)
PASS
ok      test    13.337s
```

Abbildung 15: Test Request-Two-PDU-Sessions

Um einen tieferen Einblick in die Abläufe der Tests zu bekommen, werden diese jeweils mit *tshark* an der *veth0*-Netzwerkschnittstelle mitgeschnitten. Danach sollen die *PCAP*-Dateien per *SCP* auf den Host-Rechner kopiert und zur Analyse mit *Wireshark* geöffnet werden.

4.4 Vorbereitung der virtuellen Maschine für UERANSIM

Bevor *UERANSIM* installiert werden kann, wird eine neue virtuelle Maschine auf Basis der Anfangsmaschine geklont und nach der Dokumentation von *Free5GC* konfiguriert. (vgl. *Free5GC* o.D.)

Dazu wird die VM und der *hostname* nach ‚*ueransim*‘ umbenannt, neue *MAC*-Adressen für alle Schnittstellen generiert und eine statische IP-Adresse vergeben. (siehe Abbildung 16)

Netzwerkschnittstelle (Typ)	IP-Adresse
enp0s8 (host-only)	192.168.56.102

Um die Netzwerkkonfiguration zu testen, wird ein *Ping* an die *Free5GC*-Maschine abgesetzt. (siehe Abbildung 16)

```
ping 192.168.56.101
```



```

ubuntu@ueransim:~$ ifconfig
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.102 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::a00:27ff:fe58:b1eb prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:58:b1:eb txqueuelen 1000 (Ethernet)
    RX packets 5532 bytes 488289 (488.2 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4254 bytes 510213 (510.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 36333 bytes 1991612 (1.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 36333 bytes 1991612 (1.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ubuntu@ueransim:~$ ping 192.168.56.101
PING 192.168.56.101 (192.168.56.101) 56(84) bytes of data.
64 bytes from 192.168.56.101: icmp_seq=1 ttl=64 time=0.689 ms
64 bytes from 192.168.56.101: icmp_seq=2 ttl=64 time=0.324 ms
64 bytes from 192.168.56.101: icmp_seq=3 ttl=64 time=0.926 ms
^C
--- 192.168.56.101 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2440ms
rtt min/avg/max/mdev = 0.324/0.646/0.926/0.247 ms

```

Abbildung 16: Netzwerkkonfiguration UERANSIM-VM

4.5 Installation von UERANSIM

Zur Installation von *UERANSIM* wird der offiziellen Dokumentation des Entwicklers gefolgt. Im Anschluss wird die empfohlene Überprüfung durchgeführt. (siehe Abbildung 17) Da die zu erwartenden Dateien wie in der Anleitung beschrieben vorhanden sind, wurde die Installation erfolgreich durchgeführt. (vgl. aligungr o.D.)

```

ubuntu@ueransim:~/UERANSIM/build$ ls -al
total 10876
drwxrwxr-x 2 ubuntu ubuntu 4096 Jun 20 10:24 .
drwxrwxr-x 9 ubuntu ubuntu 4096 Jun 20 10:18 ..
-rwxrwxr-x 1 ubuntu ubuntu 15560 Jun 20 10:24 libdevbnd.so
-rw-rw-r-- 1 ubuntu ubuntu 377 Jun 20 10:24 nr-binder
-rwxrwxr-x 1 ubuntu ubuntu 350568 Jun 20 10:24 nr-cli
-rwxrwxr-x 1 ubuntu ubuntu 6185128 Jun 20 10:24 nr-gnb
-rwxrwxr-x 1 ubuntu ubuntu 4563680 Jun 20 10:24 nr-ue

```

Abbildung 17: Verzeichnisstruktur /UERANSIM/build

4.6 Registrierung des Benutzerequipments

Bevor das Benutzerequipment registriert werden kann, müssen zuvor einige Konfigurationen in der *Free5GC*-VM vorgenommen werden. Dazu werden insgesamt drei Dateien geöffnet und entsprechend der Dokumentation angepasst. (vgl. *Free5GC* o.D.)

Als erstes wird der *ngapIpList*-Parameter in der AMF-Konfiguration auf die Schnittstelle gesetzt, über die dann das *gNB* erreicht werden kann. In diesem Fall wird die IP-Adresse von *127.0.0.1* auf *192.168.56.101* (IP-Adresse der *enp0s8*-Schnittstelle an der *Free5GC*-VM) geändert. (vgl. *Free5GC* o.D.)

Um die *SMF* zu konfigurieren wird die entsprechende Konfigurationsdatei geöffnet und die IP-Adresse der *N3*-Schnittstelle von *127.0.0.1* auf *192.168.56.101* geändert. Dadurch wird die Kommunikation zwischen der *SMF* und der *UPF* über das entsprechende Netzwerksegment gewährleistet. (vgl. *Free5GC* o.D.)

Im letzten Schritt wird der *UPF* konfiguriert. Dafür wird die *N3*-Schnittstelle in der *UPF*-Konfigurationsdatei von *127.0.0.1* auf die IP-Adresse der *enp0s8*-Schnittstelle (*192.168.56.101*) gesetzt. Dadurch gibt die *UPF* an, dass sie über diese Schnittstelle erreichbar ist und Verbindungen von der *SMF* akzeptiert. (vgl. *Free5GC* o.D.)

Nun ist die Konfiguration der *Free5GC*-VM abgeschlossen und somit für die Verbindung mit dem *gNB* und dem *UE* vorbereitet. Nun wird mit der Konfiguration der *UERANSIM*-VM fortgefahren. Dazu wird die *free5gc-gnb.yaml*-Datei geöffnet, um die entsprechenden Schnittstellen zu konfigurieren. Die *N2*-Schnittstelle wird auf die IP-Adresse der *enp0s8*-Netzwerkschnittstelle (*192.168.56.102*) der *UERANSIM*-VM gesetzt, um die Kommunikation mit der *AMF* zu gewährleisten. Die entsprechende *N3*-Schnittstelle, die für die Kommunikation mit der *UPF* zuständig ist, wird ebenfalls auf die IP-Adresse (*192.168.56.102*) gesetzt. Schließlich wird die noch die IP-Adresse der *AMF* angegeben, sodass das *gNB* weiß über welche Adresse das *AMF* erreichbar ist. (vgl. *Free5GC* o.D.)

Die Vorbereitung ist abgeschlossen, sobald die entsprechenden Parameter in der *free5gc-ue.yaml*-Datei mit den Parameter (siehe Abbildung 19), die in der WebConsole bei Neuanlegen des *UE* gesetzt wurden (siehe Abbildung 18), übereinstimmen. Dazu wird die Datei geöffnet und die Parameter miteinander verglichen. Entscheidend ist auch, dass der *OP-Type* übereinstimmt, da dieser dazu verwendet wird, um die Authentifizierung und Autorisierung von Benutzern sowie die Zuordnung zu bestimmten Mobilfunknetzen zu ermöglichen. (vgl. *Free5GC* o.D.)

Edit Subscriber ✕

Subscriber data number (auto-increased with SUPI)*

PLMN ID*

SUPI (IMSI)*

Authentication Method*

K*

Operator Code Type*

Operator Code Value*

SQN*

Abbildung 18: Subscriber-Info WebConsole

```

GNU nano 4.8                                free5gc-ue.yaml
# IMSI number of the UE. IMSI = [MCC|MNC|MSISDN] (In total 15 digits)
supi: 'imsi-208930000000003'
# Mobile Country Code value of HPLMN
mcc: '208'
# Mobile Network Code value of HPLMN (2 or 3 digits)
mnc: '93'
# SUCI Protection Scheme : 0 for Null-scheme, 1 for Profile A and 2 for Profile B
protectionScheme: 0
# Home Network Public Key for protecting with SUCI Profile A
homeNetworkPublicKey: '5a8d38864820197c3394b92613b20b91633cbd897119273bf8e4a6f4eec0a650'
# Home Network Public Key ID for protecting with SUCI Profile A
homeNetworkPublicKeyId: 1
# Routing Indicator
routingIndicator: '0000'

# Permanent subscription key
key: '8baf473f2f8fd09487cccbd7097c6862'
# Operator code (OP or OPC) of the UE
op: '8e27b6af0e692e750f32667a3b14605d'
# This value specifies the OP type and it can be either 'OP' or 'OPC'
opType: 'OP'
# Authentication Management Field (AMF) value
amf: '8000'
# IMEI number of the device. It is used if no SUPI is provided
imei: '356938035643803'
# IMEISV number of the device. It is used if no SUPI and IMEI is provided
imeiSv: '4370816125816151'

# List of gNB IP addresses for Radio Link Simulation
gnbSearchList:
  - 127.0.0.1

# UAC Access Identities Configuration
uacAic:
  mps: false
  mcs: false

# UAC Access Control Class
uacAcc:
  normalClass: 0
  class11: false
  class12: false
  class13: false
  class14: false
  class15: false

```

Abbildung 19: Subscriber-Info UERANSIM

4.7 Testen der Funktionalität

Um die Funktionalität zu testen wird den entsprechenden Schritten der Dokumentation gefolgt und die empfohlene Anzahl an *SSH*-Verbindungen hergestellt, sodass alle Services parallel laufen und analysiert werden können. Dazu wird eine Verbindung zur *Free5GC*-VM und drei Verbindungen zu *UERANSIM* aufgebaut. Im *Free5GC*-Terminal wird *Free5GC* gestartet und in den drei *UERANSIM*-Terminals werden zum einen das *gNB* (siehe Abbildung 20) und zum anderen das *UE* (siehe Abbildung 21) gestartet. die dritte Verbindung zur VM wird dazu genutzt die Netzwerkschnittstellen zu überprüfen und entsprechende Befehle abzusetzen. (vgl. *Free5GC* o.D.)

```
ubuntu@ueransim:~/UERANSIM$ build/nr-gnb -c config/free5gc-gnb.yaml
UERANSIM v3.2.6
[2023-06-20 16:27:21.536] [sctp] [info] Trying to establish SCTP connection... (192.168.56.101:38412)
[2023-06-20 16:27:21.553] [sctp] [info] SCTP connection established (192.168.56.101:38412)
[2023-06-20 16:27:21.553] [sctp] [debug] SCTP association setup ascId[3]
[2023-06-20 16:27:21.553] [ngap] [debug] Sending NG Setup Request
[2023-06-20 16:27:21.560] [ngap] [debug] NG Setup Response received
[2023-06-20 16:27:21.560] [ngap] [info] NG Setup procedure is successful
[2023-06-20 16:27:40.687] [rrc] [debug] UE[1] new signal detected
[2023-06-20 16:27:43.197] [rrc] [info] RRC Setup for UE[1]
[2023-06-20 16:27:43.197] [ngap] [debug] Initial NAS message received from UE[1]
[2023-06-20 16:27:43.273] [ngap] [debug] Initial Context Setup Request received
[2023-06-20 16:27:43.553] [ngap] [info] PDU session resource(s) setup for UE[1] count[1]
```

Abbildung 20: UERANSIM gNB

```
ubuntu@ueransim:~/UERANSIM$ sudo build/nr-ue -c config/free5gc-ue.yaml
[sudo] password for ubuntu:
UERANSIM v3.2.6
[2023-06-20 16:27:40.687] [nas] [info] UE switches to state [MM-DEREGISTERED/PLMN-SEARCH]
[2023-06-20 16:27:40.687] [rrc] [debug] New signal detected for cell[1], total [1] cells in coverage
[2023-06-20 16:27:40.688] [nas] [info] Selected plmn[208/93]
[2023-06-20 16:27:43.187] [rrc] [info] Selected cell plmn[208/93] tac[1] category[SUITABLE]
[2023-06-20 16:27:43.187] [nas] [info] UE switches to state [MM-DEREGISTERED/PS]
[2023-06-20 16:27:43.187] [nas] [info] UE switches to state [MM-DEREGISTERED/NORMAL-SERVICE]
[2023-06-20 16:27:43.187] [nas] [debug] Initial registration required due to [MM-DEREG-NORMAL-SERVICE]
[2023-06-20 16:27:43.187] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-06-20 16:27:43.187] [nas] [debug] Sending Initial Registration
[2023-06-20 16:27:43.197] [nas] [info] UE switches to state [MM-REGISTER-INITIATED]
[2023-06-20 16:27:43.197] [rrc] [debug] Sending RRC Setup Request
[2023-06-20 16:27:43.197] [rrc] [info] RRC connection established
[2023-06-20 16:27:43.197] [rrc] [info] UE switches to state [RRC-CONNECTED]
[2023-06-20 16:27:43.197] [nas] [info] UE switches to state [CM-CONNECTED]
[2023-06-20 16:27:43.220] [nas] [debug] Authentication Request received
[2023-06-20 16:27:43.220] [nas] [debug] Sending Authentication Failure due to SQN out of range
[2023-06-20 16:27:43.232] [nas] [debug] Authentication Request received
[2023-06-20 16:27:43.239] [nas] [debug] Security Mode Command received
[2023-06-20 16:27:43.239] [nas] [debug] Selected integrity[2] ciphering[0]
[2023-06-20 16:27:43.274] [nas] [debug] Registration accept received
[2023-06-20 16:27:43.274] [nas] [info] UE switches to state [MM-REGISTERED/NORMAL-SERVICE]
[2023-06-20 16:27:43.274] [nas] [debug] Sending Registration Complete
[2023-06-20 16:27:43.274] [nas] [info] Initial Registration is successful
[2023-06-20 16:27:43.274] [nas] [debug] Sending PDU Session Establishment Request
[2023-06-20 16:27:43.274] [nas] [debug] UAC access attempt is allowed for identity[0], category[MO_sig]
[2023-06-20 16:27:43.553] [nas] [debug] PDU Session Establishment Accept received
[2023-06-20 16:27:43.553] [nas] [info] PDU Session establishment is successful PSI[1]
[2023-06-20 16:27:43.565] [app] [info] Connection setup for PDU session[1] is successful, TUN interface[uesimtun0, 10.60.0.1] is up.
```

Abbildung 21: UERANSIM UE

Über den Befehl *ifconfig* wird überprüft, ob ein entsprechender Tunnel erstellt wurde. (siehe Abbildung 22)

```
uesimtun0: flags=369<UP,POINTOPOINT,NOTRAILERS,RUNNING,PROMISC> mtu 1400
inet 10.60.0.1 netmask 255.255.255.255 destination 10.60.0.1
inet6 fe80::deef:237:ff3f:1295 prefixlen 64 scopeid 0x20<link>
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500 (UNSPEC)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 7 bytes 448 (448.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Abbildung 22: Schnittstelle uesimtun0

Über das Absetzen eines *Pings* wird die Funktionalität von *Free5GC* bestätigt. (siehe Abbildung 23)

```
ping -I uesimtun0 google.com
```

```
ubuntu@ueransim:~$ ping -I uesimtun0 google.com
PING google.com (142.250.185.110) from 10.60.0.1 uesimtun0: 56(84) bytes of data.
64 bytes from fra16s49-in-f14.1e100.net (142.250.185.110): icmp_seq=1 ttl=58 time=10.4 ms
64 bytes from fra16s49-in-f14.1e100.net (142.250.185.110): icmp_seq=2 ttl=58 time=12.2 ms
64 bytes from fra16s49-in-f14.1e100.net (142.250.185.110): icmp_seq=3 ttl=58 time=12.7 ms
64 bytes from fra16s49-in-f14.1e100.net (142.250.185.110): icmp_seq=4 ttl=58 time=11.9 ms
64 bytes from fra16s49-in-f14.1e100.net (142.250.185.110): icmp_seq=5 ttl=58 time=11.7 ms
64 bytes from fra16s49-in-f14.1e100.net (142.250.185.110): icmp_seq=6 ttl=58 time=12.8 ms
^X64 bytes from fra16s49-in-f14.1e100.net (142.250.185.110): icmp_seq=7 ttl=58 time=12.1 ms
64 bytes from fra16s49-in-f14.1e100.net (142.250.185.110): icmp_seq=8 ttl=58 time=11.1 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7046ms
rtt min/avg/max/mdev = 10.408/11.849/12.774/0.744 ms
```

Abbildung 23: uesimtun0 ping an google.com

Außerdem lässt sich über die WebConsole die Registrierung und Verbindung des *UE* einsehen. (siehe Abbildung 24 und Abbildung 25)

Registered UEs

[Refresh](#)

SUPI ▼▲	Status ▼▲	Details
imsi-208930000000003	CONNECTED	Show Info

Abbildung 24: Erfolgreich registrierte UE auf der WebConsole

AMF Information [SUPI:imsi-208930000000003]

Information Entity	Value
AccessType	3GPP_ACCESS
CmState	CONNECTED
Guti	20893cafe00000000001
Mcc	208
Mnc	93
Supi	imsi-2089300000000003
Tac	000001
Dnn	internet
PduSessionId	1
Sd	010203
SmContextRef	urn:uuid:bbc2e78a-c3f1-4043-b2f6-2353705d2039
Sst	1

SMF Information [SUPI:imsi-208930000000003]

Information Entity	Value
AnType	3GPP_ACCESS
Dnn	internet
LocalSEID	
PDUAddress	10.60.0.1
PDUSessionID	1
RemoteSEID	
Sd	010203
Sst	1

Abbildung 25: Registrierte UE Details

5 Zusammenfassung und Ausblick

Die Implementierung von *Free5GC* in Kombination mit *UERANSIM* bietet eine großartige Möglichkeit, um erste praktische Erfahrung im Umgang mit der 5G-Technologie zu sammeln und theoretisches Wissen in die Praxis umzusetzen. Angefangen von dem Aufsetzen und Konfigurieren der virtuellen Maschinen bis hin zum Testen der Netzwerkfunktionen lässt sich das Wissen in vielen Bereichen vertiefen. Dazu wurden Tests durchgeführt und Netzwerkverkehr überwacht und analysiert, um einen ersten Eindruck über die Abläufe bei der Registrierung eines UE in einem 5G-System zu bekommen. Außerdem wird ein besseres Verständnis über die Gesamtarchitektur in einem 5G-System gewonnen. Die Kombination bietet viele Wege, um Wissen in verschiedenen Bereichen zu vertiefen und aufzubauen und besitzt somit ein großes Potenzial. Außerdem hat das Projekt einen guten Einblick in die 5G-Technologie ermöglicht, auf den in der Zukunft aufgebaut werden kann.

Das Projekt kann in Zukunft in viele Richtungen erweitert werden. So kann unter anderem eine *Non-3GPP Interworking Function (N3IWF)* implementiert werden, die eine Interoperabilität zwischen dem 5G-Netzwerk und Nicht-3GPP-Netzwerken (wie z.B. *WiFi*, *Ethernet* und *DSL*) ermöglicht. Außerdem kann die Testumgebung auch dahingehend erweitert werden, dass eine Kommunikation zwischen dem *Free5GC*-Kernnetzwerk und anderen Kernnetzwerken stattfinden kann. Schließlich haben die *AMF* und *SMF* in *Free5GC* bisher keine Schnittstelle zu *NEF*, welche mit *Go* umgesetzt werden kann.

Free5GC in Kombination mit *UERANSIM* besitzen viele Möglichkeiten der Weiterentwicklung und können sehr gut für Test- und Lernzwecke verwendet werden.

6 Abkürzungsverzeichnis

0...9

3GPP	3rd Generation Partnership Project
5G	5th Generation

A

AF	Application Function
AMF	Access and Mobility Management Function
API	Application Programming Interface
AUSF	Authentication Server Function

D

DSL	Digital Subscriber Line
------------	-------------------------

G

gNB	gNodeB
GTP	GPRS Tunneling Protocol
GUTI	Global Unique Temporary Identity

I

IoT	Internet of Things
IP	Internet Protocol
ISO	International Organization for Standardization

J

JSON	JavaScript Object Notation
-------------	----------------------------

M

MAC	Media Access Control
------------	----------------------

N

N3IWF	Non-3GPP Interworking Function
NAT	Network Address Translation
NEF	Network Exposure Function
NSSF	Network Slice Selection Function

O

OP-Type	Operation-Type
----------------	----------------

P

PCAP	Packet Capture
PCF	Policy Control Function
PDU	Protocol Data Unit
PFCP	Packet Forwarding Control Protocol

Q

QoS	Quality of Service
------------	--------------------

R

RAN	Radio Access Network
S	
SCP	Secure Copy
SMF	Session Management Function
SQL	Structured Query Language
SSH	Secure Shell
U	
UDM	Unified Data Management
UDP	User Datagram Protocol
UE	User Equipment
UPF	User Plane Function
V	
VM	Virtual Machine
W	
WiFi	Wireless Fidelity

7 Abbildungsverzeichnis

Abbildung 1: Registrierungsvorgang SMF.....	9
Abbildung 2: Netzwerkkonfiguration Free5GC	13
Abbildung 3: Konfiguration VM für Free5GC.....	14
Abbildung 4: WebConsole	15
Abbildung 5: Test Registration.....	15
Abbildung 6: Test GUTI-Registration.....	15
Abbildung 7: Test Service-Request	15
Abbildung 8: Test Xn-Handover	15
Abbildung 9: Test Deregistration	15
Abbildung 10: Test PDU-Session-Release-Request.....	16
Abbildung 11: Test Paging	16
Abbildung 12: Test N2-Handover	16
Abbildung 13: Test Non-3GPP	16
Abbildung 14: Test Resynchronization	16
Abbildung 15: Test Request-Two-PDU-Sessions	16
Abbildung 16: Netzwerkkonfiguration UERANSIM-VM	17
Abbildung 17: Verzeichnisstruktur /UERANSIM/build	17
Abbildung 18: Subscriber-Info WebConsole	19
Abbildung 19: Subscriber-Info UERANSIM	20
Abbildung 20: UERANSIM gNB.....	21
Abbildung 21: UERANSIM UE.....	21
Abbildung 22: Schnittstelle uesimtun0.....	21
Abbildung 23: uesimtun0 ping an google.com.....	22
Abbildung 24: Erfolgreich registrierte UE auf der WebConsole	22
Abbildung 25: Registrierte UE Details.....	23

8 Referenzen

1. Oracle (o.D.): Welcome to VirtualBox.org!, <https://www.virtualbox.org> [abgerufen am 07.05.2023]
2. Oracle (o.D.): Why is virtualization useful?, <https://www.virtualbox.org/manual/ch01.html#virt-why-useful> [abgerufen am 07.05.2023]
3. Kurek, T (24.04.2020): Ubuntu Server 20.04 LTS: stability, security and more, <https://ubuntu.com/blog/ubuntu-server-20-04> [abgerufen am 07.05.2023]
4. Google (o.D.): Documentation, <https://go.dev/doc/> [abgerufen am 12.05.2023]
5. Google (o.D.): Download and Install, <https://go.dev/doc/install> [abgerufen am 12.05.2023]
6. MongoDB (o.D.), <https://www.mongodb.com/de-de> [abgerufen am 18.06.2023]
7. Ross, J. (07.07.2019): Installing tshark only, <https://tshark.dev/setup/install/> [abgerufen am 14.05.2023]
8. Free5GC (o.D.): Create a Ubuntu Server, <https://free5gc.org/guide/1-vm-en/#3-create-a-ubuntu-server-vm> [abgerufen am 14.05.2023]
9. Free5GC (o.D.): Install Free5GC, <https://free5gc.org/guide/3-install-free5gc/> [abgerufen am 14.05.2023]
10. Free5GC (o.D.): Test Free5GC, <https://free5gc.org/guide/4-test-free5gc/> [abgerufen am 27.05.2023]
11. Free5GC (o.D.): Install UERANSIM VM, <https://free5gc.org/guide/5-install-ueransim/#1-install-ueramsim-vm> [abgerufen am 10.06.2023]
12. Aligungr (24.01.2021): Installation, <https://github.com/aligungr/UERANSIM/wiki/Installation> [abgerufen am 14.06.2023]
13. Trick, U. (o.D.): Mobilfunknetze der 5. Generation, https://e-technik.org/frame_vorlesungen.htm [abgerufen am 20.06.2023]
14. Sultan, A. (11.10.2022): 5G System Overview, <https://www.3gpp.org/technologies/5g-system-overview> [abgerufen am 09.06.2023]
15. Dryjanski, M. (02.03.2018): 5G Core Network Functions, <https://www.grandmetric.com/5g-core-network-functions/> [abgerufen am 09.06.2023]
16. Köpsell, Stefan/Ruzhanskiy, Andrey/Hecker, Andreas/Stachorra, Dirk (21.02.2022): Open-RAN Risikoanalyse, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Studien/5G/5GRAN-Risikoanalyse.pdf?__blob=publicationFile&v=5 [abgerufen am 21.06.2023]
17. 3GPP (19.02.2021): NG-RAN Architecture, <https://www.3gpp.org/news-events/3gpp-news/ng-ran-architecture> [abgerufen am 19.06.2023]

-
18. Huo, Yiming/Dong, Xiaodai/Xu, Wei (o.D.): 5G Cellular User Equipment: From Theory to Practical Hardware Design, <https://ieeexplore.ieee.org/document/7983335> [abgerufen am 18.06.2023]
 19. Devopedia (02.05.2023): Packet Forwarding Control Protocol, <https://devopedia.org/packet-forwarding-control-protocol> [abgerufen am 22.06.2023]
 20. ETSI (2020-11): ETSI TS 129 244, https://www.etsi.org/deliver/etsi_ts/129200_129299/129244/16.05.00_60/ts_129244v160500p.pdf [abgerufen am 26.06.2023]

9 Anhang

Anhang 1: Wireshark-Mittschnitt Test-Registration

pfcp gtp icmp						
No.	Time	Source	Destination	Protocol	Length	Info
9	2.132363159	10.200.200.1	10.200.200.1	PFCP	72	PFCP Association Setup Request
10	2.133013911	10.200.200.101	10.200.200.1	PFCP	72	PFCP Association Setup Response
11	3.644490526	10.200.200.1	10.200.200.1	PFCP	329	PFCP Session Establishment Request
12	3.646483420	10.200.200.101	10.200.200.1	PFCP	89	PFCP Session Establishment Response
13	3.650838458	10.200.200.1	10.200.200.1	PFCP	190	PFCP Session Modification Request
14	3.651249091	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response
18	4.650434869	10.60.0.1	10.60.0.101	GTP <ICMP>	130	Echo (ping) request id=0x306a, seq=1/256, ttl=64 (reply in 19)
19	4.650523477	10.60.0.101	10.60.0.1	GTP <ICMP>	106	Echo (ping) reply id=0x306a, seq=1/256, ttl=64 (request in 18)

> Frame 9: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0
> Ethernet II, Src: 6a:54:a6:6f:c4:52 (6a:54:a6:6f:c4:52), Dst: ba:01:ba:51:55:0f (ba:01:ba:51:55:0f)
> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805
▼ Packet Forwarding Control Protocol
 > Flags: 0x20
 Message Type: PFCP Association Setup Request (5)
 Length: 26
 Sequence Number: 1
 Spare: 0
 > Node ID : IPv4 address: 10.200.200.1
 > Recovery Time Stamp : May 23, 2023 10:32:27.000000000 UTC
 > CP Function Features :
 [Response In: 10]

0000 ba 01
0010 00 3a
0020 c8 65
0030 01 00
0040 17 14

Anhang 2: Wireshark-Mittschnitt Test-GUTI-Registration

pfcp gtp						
No.	Time	Source	Destination	Protocol	Length	Info
9	2.543319538	10.200.200.1	10.200.200.101	PFCP	72	PFCP Association Setup Request
10	2.543590413	10.200.200.101	10.200.200.1	PFCP	72	PFCP Association Setup Response

> Frame 9: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0
> Ethernet II, Src: fe:f2:10:cd:89:17 (fe:f2:10:cd:89:17), Dst: 06:74:c3:7b:cf:8f (06:74:c3:7b:cf:8f)
> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805
▼ Packet Forwarding Control Protocol
 > Flags: 0x20
 Message Type: PFCP Association Setup Request (5)
 Length: 26
 Sequence Number: 1
 Spare: 0
 > Node ID : IPv4 address: 10.200.200.1
 > Recovery Time Stamp : May 23, 2023 10:42:17.000000000 UTC
 ▼ CP Function Features :
 IE Type: CP Function Features (89)
 IE Length: 1
 0... .. = UIAUR: Not supported
 .0.. .. = ARDR: Not supported
 ..0. = MPAS: Not supported
 ...0 = BUNDL: Not supported
 0... = SSET: Not supported
 0.. = EPFAR: Not supported
 0. = OVRLL: Not supported
 0 = LOAD: Not supported
 [Response In: 10]

0000 06 74
0010 00 3a
0020 c8 65
0030 01 00
0040 17 17

Anhang 3: Wireshark-Mittschnitt Test-Service-Request

pfcip gtp icmp						
No.	Time	Source	Destination	Protocol	Length	Info
10	3.040284753	10.200.200.1	10.200.200.101	PFCP	72	PFCP Association Setup Request
11	3.041059575	10.200.200.101	10.200.200.1	PFCP	72	PFCP Association Setup Response
13	4.562283390	10.200.200.1	10.200.200.101	PFCP	329	PFCP Session Establishment Request
14	4.564042359	10.200.200.101	10.200.200.1	PFCP	89	PFCP Session Establishment Response
15	4.570884217	10.200.200.1	10.200.200.101	PFCP	190	PFCP Session Modification Request
16	4.571624403	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response
17	4.777469672	10.200.200.1	10.200.200.101	PFCP	133	PFCP Session Modification Request
18	4.779328524	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response
20	5.798109720	10.200.200.1	10.200.200.101	PFCP	190	PFCP Session Modification Request
21	5.798723735	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response

> Frame 10: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0

> Ethernet II, Src: 4e:5a:fc:b6:ae:04 (4e:5a:fc:b6:ae:04), Dst: 52:02:03:5e:e5:e4 (52:02:03:5e:e5:e4)

> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101

> User Datagram Protocol, Src Port: 8805, Dst Port: 8805

> Packet Forwarding Control Protocol

> Flags: 0x20

Message Type: PFCP Association Setup Request (5)

Length: 26

Sequence Number: 1

Spare: 0

> Node ID : IPv4 address: 10.200.200.1

> Recovery Time Stamp : May 23, 2023 10:43:45.000000000 UTC

> CP Function Features :

[Response In: 11]

0000 52 02
0010 00 3a
0020 c8 65
0030 01 00
0040 17 17

Anhang 4: Wireshark-Mittschnitt Test-XnHandover

pfcip gtp icmp						
No.	Time	Source	Destination	Protocol	Length	Info
10	2.990301922	10.200.200.1	10.200.200.101	PFCP	72	PFCP Association Setup Request
11	2.990687849	10.200.200.101	10.200.200.1	PFCP	72	PFCP Association Setup Response
13	4.535762441	10.200.200.1	10.200.200.101	PFCP	329	PFCP Session Establishment Request
14	4.536387401	10.200.200.101	10.200.200.1	PFCP	89	PFCP Session Establishment Response
15	4.540698308	10.200.200.1	10.200.200.101	PFCP	190	PFCP Session Modification Request
16	4.541313518	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response
18	6.545810749	10.200.200.1	10.200.200.101	PFCP	133	PFCP Session Modification Request
19	6.546142428	10.200.200.102	10.200.200.1	GTP	50	End Marker
22	6.546163596	10.200.200.1	10.200.200.102	ICMP	78	Destination unreachable (Port unreachable)
23	6.551203540	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response

> Frame 10: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0

> Ethernet II, Src: e2:3b:f9:cf:86:c2 (e2:3b:f9:cf:86:c2), Dst: d2:8f:c0:eb:ed:e9 (d2:8f:c0:eb:ed:e9)

> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101

> User Datagram Protocol, Src Port: 8805, Dst Port: 8805

> Packet Forwarding Control Protocol

> Flags: 0x20

Message Type: PFCP Association Setup Request (5)

Length: 26

Sequence Number: 1

Spare: 0

> Node ID : IPv4 address: 10.200.200.1

> Recovery Time Stamp : May 23, 2023 10:45:20.000000000 UTC

> CP Function Features :

[Response In: 11]

0000 d2 8f
0010 00 3a
0020 c8 65
0030 01 00
0040 17 17

Anhang 5: Wireshark-Mittschnitt Test-Deregistration

pfcip gtp icmp						
No.	Time	Source	Destination	Protocol	Length	Info
11	2.949271715	10.200.200.1	10.200.200.101	PFCP	72	PFCP Association Setup Request
12	2.949597328	10.200.200.101	10.200.200.1	PFCP	72	PFCP Association Setup Response

> Frame 11: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0

> Ethernet II, Src: 0e:85:4f:6b:54:71 (0e:85:4f:6b:54:71), Dst: f6:4e:70:04:e0:de (f6:4e:70:04:e0:de)

> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101

> User Datagram Protocol, Src Port: 8805, Dst Port: 8805

> Packet Forwarding Control Protocol

> Flags: 0x20

Message Type: PFCP Association Setup Request (5)

Length: 26

Sequence Number: 1

Spare: 0

> Node ID : IPv4 address: 10.200.200.1

> Recovery Time Stamp : May 23, 2023 10:46:45.000000000 UTC

> CP Function Features :

[Response In: 12]

0000 f6 4e
0010 00 3a
0020 c8 65
0030 01 00
0040 17 18

Anhang 6: Wireshark-Mittschnitt Test-PDU-Session-Release-Request

pfcp igmp icmp						
No.	Time	Source	Destination	Protocol	Length	Info
11	2.998816211	10.200.200.1	10.200.200.101	PFCP	72	PFCP Association Setup Request
12	3.003375742	10.200.200.101	10.200.200.1	PFCP	72	PFCP Association Setup Response
13	4.493480747	10.200.200.1	10.200.200.101	PFCP	329	PFCP Session Establishment Request
14	4.494012312	10.200.200.101	10.200.200.1	PFCP	89	PFCP Session Establishment Response
15	4.504754819	10.200.200.1	10.200.200.101	PFCP	190	PFCP Session Modification Request
16	4.505139465	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response
17	4.711527127	10.200.200.1	10.200.200.101	PFCP	58	PFCP Session Deletion Request
18	4.712454095	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Deletion Response

> Frame 11: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0	0000	9a 3b
> Ethernet II, Src: c6:b4:ab:f5:81:8e (c6:b4:ab:f5:81:8e), Dst: 9a:3b:0e:56:8a:91 (9a:3b:0e:56:8a:91)	0010	00 3a
> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101	0020	c8 65
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805	0030	01 00
> Packet Forwarding Control Protocol	0040	17 18
> Flags: 0x20		
Message Type: PFCP Association Setup Request (5)		
Length: 26		
Sequence Number: 1		
Spare: 0		
> Node ID : IPv4 address: 10.200.200.1		
> Recovery Time Stamp : May 23, 2023 10:48:32.000000000 UTC		
> CP Function Features :		
[Response In: 12]		

Anhang 7: Wireshark-Mittschnitt Test-Paging

pfcp igmp icmp						
No.	Time	Source	Destination	Protocol	Length	Info
11	3.115489938	10.200.200.1	10.200.200.101	PFCP	72	PFCP Association Setup Request
12	3.115864223	10.200.200.101	10.200.200.1	PFCP	72	PFCP Association Setup Response
14	4.661767322	10.200.200.1	10.200.200.101	PFCP	329	PFCP Session Establishment Request
15	4.662344204	10.200.200.101	10.200.200.1	PFCP	89	PFCP Session Establishment Response
16	4.669999331	10.200.200.1	10.200.200.101	PFCP	190	PFCP Session Modification Request
17	4.670455563	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response
18	4.871102750	10.200.200.1	10.200.200.101	PFCP	133	PFCP Session Modification Request
19	4.871654798	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response
21	5.896329263	10.200.200.101	10.200.200.1	PFCP	73	PFCP Session Report Request
22	5.898553967	10.200.200.1	10.200.200.101	PFCP	63	PFCP Session Report Response
23	7.923584442	10.200.200.1	10.200.200.101	PFCP	190	PFCP Session Modification Request
24	7.923982820	10.60.0.101	10.60.0.1	GTP <UDP>	91	46236 → 8080 Len=5
25	7.924156823	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response

> Frame 11: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0	0000	a6 83
> Ethernet II, Src: 86:30:0b:14:fc:37 (86:30:0b:14:fc:37), Dst: a6:83:25:8a:96:1f (a6:83:25:8a:96:1f)	0010	00 3a
> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101	0020	c8 65
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805	0030	01 00
> Packet Forwarding Control Protocol	0040	17 18
> Flags: 0x20		
Message Type: PFCP Association Setup Request (5)		
Length: 26		
Sequence Number: 1		
Spare: 0		
> Node ID : IPv4 address: 10.200.200.1		
> Recovery Time Stamp : May 23, 2023 10:49:39.000000000 UTC		
> CP Function Features :		
[Response In: 12]		

Anhang 8: Wireshark-Mittschnitt Test-N2-Handover

pfcp gtp icmp					
No.	Time	Source	Destination	Protocol	Length Info
11	3.339334708	10.200.200.1	10.200.200.101	PFCP	72 PFCP Association Setup Request
12	3.339719915	10.200.200.101	10.200.200.1	PFCP	72 PFCP Association Setup Response
14	4.907739228	10.200.200.1	10.200.200.101	PFCP	329 PFCP Session Establishment Request
15	4.908700474	10.200.200.101	10.200.200.1	PFCP	89 PFCP Session Establishment Response
16	4.914055218	10.200.200.1	10.200.200.101	PFCP	190 PFCP Session Modification Request
17	4.914796391	10.200.200.101	10.200.200.1	PFCP	63 PFCP Session Modification Response
21	5.913819594	10.60.0.1	10.60.0.101	GTP <ICMP>	130 Echo (ping) request id=0x306a, seq=1/256, ttl=64 (reply in 22)
22	5.913974425	10.60.0.101	10.60.0.1	GTP <ICMP>	106 Echo (ping) reply id=0x306a, seq=1/256, ttl=64 (request in 21)
23	7.354920250	10.200.200.1	10.200.200.101	PFCP	190 PFCP Session Modification Request
24	7.355502230	10.200.200.101	10.200.200.1	PFCP	63 PFCP Session Modification Response
25	8.351720380	10.60.0.1	10.60.0.101	GTP <ICMP>	130 Echo (ping) request id=0x306a, seq=1/256, ttl=64 (reply in 28)
28	8.351757037	10.60.0.101	10.60.0.1	GTP <ICMP>	106 Echo (ping) reply id=0x306a, seq=1/256, ttl=64 (request in 25)
> Frame 11: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0					
> Ethernet II, Src: 2e:ef:53:3d:5d:01 (2e:ef:53:3d:5d:01), Dst: 3e:bd:7b:fa:78:85 (3e:bd:7b:fa:78:85)					
> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101					
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805					
▼ Packet Forwarding Control Protocol					
> Flags: 0x20					
Message Type: PFCP Association Setup Request (5)					
Length: 26					
Sequence Number: 1					
Spare: 0					
> Node ID : IPv4 address: 10.200.200.1					
> Recovery Time Stamp : May 23, 2023 10:51:19.000000000 UTC					
> CP Function Features :					
[Response In: 12]					
					0000 3e bd
					0010 00 3a
					0020 c8 65
					0030 01 00
					0040 17 19

Anhang 9: Wireshark-Mittschnitt Test-Non-3GPP

pfcp gtp icmp					
No.	Time	Source	Destination	Protocol	Length Info
9	2.641097150	10.200.200.1	10.200.200.101	PFCP	72 PFCP Association Setup Request
10	2.641433069	10.200.200.101	10.200.200.1	PFCP	72 PFCP Association Setup Response
17	17.947933126	10.200.200.1	10.200.200.101	PFCP	329 PFCP Session Establishment Request
18	17.948534480	10.200.200.101	10.200.200.1	PFCP	89 PFCP Session Establishment Response
21	17.955996721	10.200.200.2	10.200.200.102	GTP	56 Echo request
22	17.956010560	10.200.200.102	10.200.200.2	GTP	56 Echo response
23	17.967960449	10.200.200.1	10.200.200.101	PFCP	190 PFCP Session Modification Request
24	17.968357100	10.200.200.101	10.200.200.1	PFCP	63 PFCP Session Modification Response
25	17.991120277	fe80::5efe:10.0.0.1...	ff02::2	GTP <ICMPv6>	114 Router Solicitation from 0a:00:00:80:00:00
26	18.196072902	10.200.200.1	10.200.200.101	PFCP	293 PFCP Session Establishment Request
27	18.196547733	10.200.200.101	10.200.200.1	PFCP	89 PFCP Session Establishment Response
28	18.207031779	10.200.200.1	10.200.200.101	PFCP	190 PFCP Session Modification Request
29	18.210655189	10.200.200.101	10.200.200.1	PFCP	63 PFCP Session Modification Response
30	18.217444553	fe80::5efe:10.0.0.1...	ff02::2	GTP <ICMPv6>	114 Router Solicitation from 0a:00:00:80:00:00
31	18.457180945	10.200.200.1	10.200.200.101	PFCP	293 PFCP Session Establishment Request
32	18.458348880	10.200.200.101	10.200.200.1	PFCP	89 PFCP Session Establishment Response
33	18.477940454	fe80::5efe:10.0.0.1...	ff02::2	GTP <ICMPv6>	114 Router Solicitation from 0a:00:00:80:00:00
34	18.479988121	10.200.200.1	10.200.200.101	PFCP	190 PFCP Session Modification Request
35	18.480558198	10.200.200.101	10.200.200.1	PFCP	63 PFCP Session Modification Response
36	18.702921857	10.200.200.1	10.200.200.101	PFCP	293 PFCP Session Establishment Request
37	18.703530060	10.200.200.101	10.200.200.1	PFCP	89 PFCP Session Establishment Response
38	18.732621363	fe80::5efe:10.0.0.1...	ff02::2	GTP <ICMPv6>	114 Router Solicitation from 0a:00:00:80:00:00
39	18.735213638	10.200.200.1	10.200.200.101	PFCP	190 PFCP Session Modification Request
40	18.740155070	10.200.200.101	10.200.200.1	PFCP	63 PFCP Session Modification Response
41	21.751054836	10.60.0.1	10.60.0.101	GTP <ICMP>	102 Echo (ping) request id=0xbb86, seq=0/0, ttl=64 (reply in 42)
42	21.751076918	10.60.0.101	10.60.0.1	GTP <ICMP>	102 Echo (ping) reply id=0xbb86, seq=0/0, ttl=64 (request in 41)
43	22.337586793	fe80::5efe:10.0.0.1...	ff02::2	GTP <ICMPv6>	114 Router Solicitation from 0a:00:00:80:00:00
44	22.337757229	fe80::5efe:10.0.0.1...	ff02::2	GTP <ICMPv6>	114 Router Solicitation from 0a:00:00:80:00:00
45	22.751766665	10.60.0.1	10.60.0.101	GTP <ICMP>	102 Echo (ping) request id=0xbb86, seq=1/256, ttl=64 (reply in 46)
46	22.751785020	10.60.0.101	10.60.0.1	GTP <ICMP>	102 Echo (ping) reply id=0xbb86, seq=1/256, ttl=64 (request in 45)
47	22.849029773	fe80::5efe:10.0.0.1...	ff02::2	GTP <ICMPv6>	114 Router Solicitation from 0a:00:00:80:00:00
48	22.849227454	fe80::5efe:10.0.0.1...	ff02::2	GTP <ICMPv6>	114 Router Solicitation from 0a:00:00:80:00:00
51	23.751354062	10.60.0.1	10.60.0.101	GTP <ICMP>	102 Echo (ping) request id=0xbb86, seq=2/512, ttl=64 (reply in 52)
52	23.751368090	10.60.0.101	10.60.0.1	GTP <ICMP>	102 Echo (ping) reply id=0xbb86, seq=2/512, ttl=64 (request in 51)
53	24.752959688	10.60.0.1	10.60.0.101	GTP <ICMP>	102 Echo (ping) request id=0xbb86, seq=3/768, ttl=64 (reply in 54)
54	24.752972706	10.60.0.101	10.60.0.1	GTP <ICMP>	102 Echo (ping) reply id=0xbb86, seq=3/768, ttl=64 (request in 53)
55	25.751506602	10.60.0.1	10.60.0.101	GTP <ICMP>	102 Echo (ping) request id=0xbb86, seq=4/1024, ttl=64 (reply in 56)
56	25.751549085	10.60.0.101	10.60.0.1	GTP <ICMP>	102 Echo (ping) reply id=0xbb86, seq=4/1024, ttl=64 (request in 55)
> Frame 9: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0					
> Ethernet II, Src: 06:26:26:dd:05:8e (06:26:26:dd:05:8e), Dst: f2:4e:bb:31:87:40 (f2:4e:bb:31:87:40)					
> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101					
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805					
▼ Packet Forwarding Control Protocol					
> Flags: 0x20					
Message Type: PFCP Association Setup Request (5)					
Length: 26					
Sequence Number: 1					
Spare: 0					
> Node ID : IPv4 address: 10.200.200.1					
> Recovery Time Stamp : May 23, 2023 10:52:41.000000000 UTC					
> CP Function Features :					
[Response In: 10]					
					0000 f2 4e
					0010 00 3a
					0020 c8 65
					0030 01 00
					0040 17 19

Anhang 10: Wireshark-Mittschnitt Test-Re-Synchronisation

No.	Time	Source	Destination	Protocol	Length	Info
11	3.240664564	10.200.200.1	10.200.200.101	PFCP	72	PFCP Association Setup Request
12	3.242140371	10.200.200.101	10.200.200.1	PFCP	72	PFCP Association Setup Response
13	4.790534180	10.200.200.1	10.200.200.101	PFCP	329	PFCP Session Establishment Request
14	4.792622807	10.200.200.101	10.200.200.1	PFCP	89	PFCP Session Establishment Response
15	4.803963909	10.200.200.1	10.200.200.101	PFCP	190	PFCP Session Modification Request
16	4.804695701	10.200.200.101	10.200.200.1	PFCP	63	PFCP Session Modification Response
21	5.803671396	10.60.0.1	10.60.0.101	GTP <ICMP>	130	Echo (ping) request id=0x306a, seq=1/256, ttl=64 (reply in 22)
22	5.803753693	10.60.0.101	10.60.0.1	GTP <ICMP>	106	Echo (ping) reply id=0x306a, seq=1/256, ttl=64 (request in 21)

> Frame 11: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0	0000	4a 9a
> Ethernet II, Src: 9a:dc:48:6e:e0:7a (9a:dc:48:6e:e0:7a), Dst: 4a:9a:0c:7e:84:b0 (4a:9a:0c:7e:84:b0)	0010	00 3a
> Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101	0020	c8 65
> User Datagram Protocol, Src Port: 8805, Dst Port: 8805	0030	01 00
> Packet Forwarding Control Protocol	0040	17 1a
> Flags: 0x20		
Message Type: PFCP Association Setup Request (5)		
Length: 26		
Sequence Number: 1		
Spare: 0		
> Node ID : IPv4 address: 10.200.200.1		
> Recovery Time Stamp : May 23, 2023 10:58:54.000000000 UTC		
> CP Function Features :		
[Response In: 12]		

Anhang 11: Wireshark-Mittschnitt Test-Request-Two-PDU-Sessions

No.	Time	Source	Destination	Protocol	Length	Info
28	3.251701491	10.200.200.1	10.200.200.101	PFPCP	72	PFPCP Association Setup Request
29	3.252340953	10.200.200.101	10.200.200.1	PFPCP	72	PFPCP Association Setup Response
32	3.253249368	10.200.200.1	10.200.200.102	PFPCP	72	PFPCP Association Setup Request
33	3.253805707	10.200.200.102	10.200.200.1	PFPCP	72	PFPCP Association Setup Response
36	4.814024145	10.200.200.1	10.200.200.101	PFPCP	329	PFPCP Session Establishment Request
37	4.815473146	10.200.200.101	10.200.200.1	PFPCP	89	PFPCP Session Establishment Response
38	4.825708370	10.200.200.1	10.200.200.101	PFPCP	190	PFPCP Session Modification Request
39	4.826342380	10.200.200.101	10.200.200.1	PFPCP	63	PFPCP Session Modification Response
43	5.844039135	10.200.200.1	10.200.200.102	PFPCP	332	PFPCP Session Establishment Request
44	5.845999276	10.200.200.102	10.200.200.1	PFPCP	89	PFPCP Session Establishment Response
45	5.854017882	10.200.200.1	10.200.200.102	PFPCP	192	PFPCP Session Modification Request
46	5.854673417	10.200.200.102	10.200.200.1	PFPCP	63	PFPCP Session Modification Response
47	6.853740167	10.60.0.1	10.60.0.101	GTP <ICMP>	130	Echo (ping) request id=0x306a, seq=1/256, ttl=64 (reply in 48)
48	6.853843890	10.60.0.101	10.60.0.1	GTP <ICMP>	106	Echo (ping) reply id=0x306a, seq=1/256, ttl=64 (request in 47)
49	7.854371458	10.62.0.1	10.60.0.102	GTP <ICMP>	130	Echo (ping) request id=0x306a, seq=1/256, ttl=64 (no response f
> Frame 28: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface veth0, id 0 > Ethernet II, Src: ce:7d:ba:6b:6b:95 (ce:7d:ba:6b:6b:95), Dst: 4a:03:35:16:d4:f9 (4a:03:35:16:d4:f9) > Internet Protocol Version 4, Src: 10.200.200.1, Dst: 10.200.200.101 > User Datagram Protocol, Src Port: 8805, Dst Port: 8805 > Packet Forwarding Control Protocol > Flags: 0x20 Message Type: PFPCP Association Setup Request (5) Length: 26 Sequence Number: 1 Spare: 0 > Node ID : IPv4 address: 10.200.200.1 > Recovery Time Stamp : May 23, 2023 11:00:24.000000000 UTC > CP Function Features : [Response In: 29]						
	0000	4a	03			
	0010	00	3a			
	0020	c8	65			
	0030	01	00			
	0040	17	1b			