

Masterarbeit

TELEOPERATION ÜBER 5G

ausgeführt am



FACHHOCHSCHULE DER WIRTSCHAFT

Fachhochschul-Masterstudiengang
Automatisierungstechnik-Wirtschaft

von

Florian Seidl

51841212

betreut und begutachtet von

FH-Prof. Dipl.-Ing. Dieter Lutzmayr

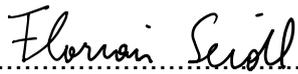
Graz, im November 2022

Florian Seidl

.....
Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.


.....

Unterschrift

DANKSAGUNG

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Masterarbeit unterstützt und motiviert haben.

Hervorheben möchte ich meine Familie, die mich durch ihre stets aufmunternden Worte selbst die herausforderndsten Phasen meistern ließ. Eine besondere Rolle kam hier meinem Bruder Matthias Seidl zu, der mich sowohl auf persönlicher als auch auf technischer Ebene durch seine berufliche Tätigkeit bei der Citycom Telekommunikation GmbH unterstützte. Weiters möchte ich mich bei Dr. Thomas Klemenschits und dem restlichen 5G-Team des Unternehmens bedanken.

Ein Dankeschön möchte ich ebenfalls den Verantwortlichen der Studienrichtung Automatisierungstechnik der Fachhochschule CAMPUS 02 für die Bereitstellung der Infrastruktur aussprechen, ohne die diese Masterarbeit nicht möglich gewesen wäre. Besonderer Dank gilt auch meinem Betreuer FH-Prof. Dipl.-Ing. Dieter Lutzmayr, der mich bei der Ausführung organisatorisch sowie durch konstruktive Gespräche und seine pragmatische Sichtweise unterstützte.

KURZFASSUNG

Die Mobilfunktechnologie beeinflusst die Kommunikation und den Informationszugang der Menschheit seit den 1980er Jahren. Der 2017 definierte New-Radio-Standard der fünften Generation, bezeichnet als ‚5G‘, bietet fundamental neue Funktionalitäten, die mit keiner vorhergegangenen Mobilfunktechnologie vergleichbar sind. Aus diesem Grund wurde an der Fachhochschule CAMPUS 02 eine 5G-Infrastruktur errichtet, um Forschungsarbeit in den neuen Anwendungsfeldern und Einsatzszenarien zu betreiben.

Das Ziel dieser Arbeit ist die Realisierung eines ersten Anwendungsfalls in der Infrastruktur der Fachhochschule CAMPUS 02, um die Datenübertragung mit niedriger Latenz und Latenzzeitmessungen in einem 5G-Netzwerk zu untersuchen.

Zu Beginn erfolgte eine Sichtung der verfügbaren Soft- und Hardware für die Entwicklung eines Versuchsträgers, der die Anforderungen erfüllt und fernbedienbar ausgeführt werden kann. Der Aufbau bestand aus zwei USB-Webcams, montiert auf einem motorisierten Stativ. Die Steuerung erfolgte mit einem Servo-Controller-Board und einem Raspberry Pi 4, der mit einem 5G-Modem/Router verbunden war. Als Wiedergabegerät der Videoübertragung diente eine VR-Brille, die mit einem Kabel an einem Rechner angeschlossen war. Die Anforderungen an die Datenübertragung konnten durch die Nutzung des bidirektionalen WebRTC-Protokolls erfüllt werden. Dieses ermöglichte die Video- und Audioübermittlung sowie einen Datenkanal für die Übermittlung der Kopfbewegung.

Als wichtigstes Ergebnis sind die Entwicklung und Inbetriebnahme der Testanwendung zur Durchführung von Netzwerkmessungen mittels perfSONAR, einer Network-Performance-Monitoring-Software, zu nennen. Aufgrund von Limitationen technischer, infrastruktureller und zeitlicher Natur konnten nicht alle Aspekte vollumfänglich untersucht und umgesetzt werden. Basierend auf den erzielten Resultaten wurde eine Roadmap für die Rekonfiguration der 5G-Infrastruktur erstellt und weitere Entwicklungsmöglichkeiten für den Versuchsaufbau aufgezeigt.

ABSTRACT

Radio Access Networks have influenced the communication and information access of humanity since the 1980s. The recently deployed New Radio Standard of the fifth generation 5G features completely new functionalities, which are not comparable to any past radio network technology. Therefore, a 5G infrastructure was built at the University of Applied Sciences CAMPUS 02 to enable research in the new application fields and deployment scenarios.

The aim of the master's thesis is to establish a first use case within the CAMPUS 02 5G infrastructure to investigate the execution of low latency data transmissions and latency measurements in a 5G network.

Initially, a screening of available hardware and software solutions had to be carried out to enable the procurement and development of a prototype that meets the requirements of the test application and can be operated remotely. The prototype hardware consists of two USB-Webcams mounted on a motorized tripod, which are controlled by a servo-controller-board and Raspberry Pi 4 connected to a 5G modem/router. As playback device a VR-Headset wired to a computer was utilized to display the 5G video transmission. The application requirements for the data transmission could be fulfilled by the deployment of the bidirectional WebRTC-Protocol, which enables video, audio and a data channel for the head movement.

The major outcome was the development and initial operation of a functional test application in order to carry out measurements by the implementation of the perfSONAR network performance monitoring software. Due to technical, infrastructural and timeframe limitations, not every aspect could be investigated in full detail. Based on the findings a roadmap was established for the reconfiguration of the 5G infrastructure and the implementation of further enhancements to the prototype test application.

INHALTSVERZEICHNIS

1	Einleitung.....	1
2.3	Entwicklung und Hauptanwendungsszenarien von 5G	4
2.4	NSA- und SA-Architektur	6
2.5	Enhanced Mobile Broadband	7
2.6	Ultra Reliable and Low Latency Communications	9
2.7	Massive Machine Type Communication	10
2.8	Network Slicing	10
2.9	Multi-Access Edge Computing.....	12
2.10	High Data Rate and Low Latency AR/VR.....	17
3	Anbindung der Hardware-Komponenten.....	18
3.1	Inter Integrated Circuit Bus I ² C	18
3.2	Universal Serial Bus USB	18
4	Videoverbindung	20
4.1	Streaming-Protokolle	20
5	Optimierung von 5G-Kommunikation	23
5.1	Enhanced Mobile Broadband	23
5.2	Ultra Reliable and Low Latency Communications	24
5.3	Massive Machine Type Communication.....	26
6	Latenzzeitmessung	27
6.1	Latenzzeit, Download- und Upload-Geschwindigkeit in Österreich.....	27
6.2	Verfahren zur Latenzzeitmessung	29
6.3	Performance-Testing für industrielle Automatisierung	32
7	Definition des Versuchsaufbaus.....	35
8	Komponentenauswahl.....	36
8.1	Raspberry Pi Model B 5G HAT.....	36
8.2	Schwenk- / Neigesystem	37
8.3	Kamera	38
8.4	Motorsteuerung.....	40
8.5	VR-Headset	42
9	Praktische Umsetzung	43
9.1	Raspberry Pi	43
9.2	WebRTC Videoübertragung	43
9.2.1	UV4L Basic Setup.....	44
9.2.2	UFW Uncomplicated Firewall	50
9.3	Dynamic DNS Service	52
9.4	Port Forwarding	54
9.5	VR-Headset Programmierung	55
9.5.1	Datenkanal.....	59
9.5.2	Servo-Motor-Steuerung	60

9.6	Stereoskopisches Kamerasetup	61
9.7	Netzwerkmessung	64
9.7.1	Ausgangslage	64
9.7.2	Installation CentOS 7	66
9.7.3	perfSONAR	67
10	Ergebnisse und Ausblick	75
10.1	Ergebnisse Videoübertragung	75
10.2	Ergebnisse VR-Headset	77
10.3	Ergebnisse Netzwerkmessung	78
10.4	Ausblick 5G Infrastruktur	80
10.5	Fazit	82
	Literaturverzeichnis	83
	Abbildungsverzeichnis	88
	Tabellenverzeichnis	94
	Quellcodeverzeichnis	95

1 EINLEITUNG

Durch das weltweit steigende Datenübertragungsvolumen mit immer mehr Teilnehmern und den zunehmenden Ausbau der 5G-Infrastruktur wurde eine Kooperation zwischen Citycom, Nokia und der Fachhochschule (FH) CAMPUS02 gebildet. Im Zuge dieser Zusammenarbeit wurden Ende 2021 eine 5G-Outdoor- und drei Indoor-Antennen sowie das dazugehörige Equipment im Laborkomplex der Fachhochschule installiert. Dadurch sollte ein Kompetenzzentrum für 5G und das Internet-of-Things (IoT) etabliert werden.

Das Ziel dieser Arbeit ist die Realisierung eines ersten Anwendungsfalls in der Infrastruktur der FH CAMPUS 02, um Datenübertragung mit niedriger Latenz und Latenzzeitmessungen in einem 5G-Netzwerk zu untersuchen.

Der Versuchsaufbau besteht aus einem stereoskopischen Kamerasystem montiert auf einem dreh- und schwenkbar gelagerten Stativ sowie eine Virtual-Reality(VR)-Brille als Wiedergabegerät. Mittels des 5G-Mobilfunkstandards erfolgt die Übertragung von Bilddaten und Kopfdrotation zwischen den Endgeräten. Mittels der Bewegungsdaten von der VR-Brille wird die Kameraausrichtung gesteuert.

Die Umsetzung erfolgt mit Soft- und Hardware. Sie umfasst Konzeptentwicklung, Komponentenauswahl, Aufbau und Inbetriebnahme. Die Versuchsplattform soll eine Weiterentwicklung hinsichtlich des mobilen Einsatzes durch Montage auf einem Roboterchassis und Fernsteuerung mit nachgelagerter Bildauswertung wie Objekterkennung unterstützen.

Die Arbeit ist in einen Theorie- und einen Praxisteil gegliedert. Ersterer dient der Grundlagenrecherche, die anschließend auf den Versuch angewandt wird. Der Praxisteil umfasst die Anfertigung und Inbetriebnahme des Aufbaus sowie die anschließende Durchführung definierter Messungen.

Im Theoriekapitel werden folgende Themen erläutert:

1. 5G-Mobilfunkstandard,
2. Anbindung der Hardware-Komponenten,
3. Videoverbindung,
4. Optimierung von 5G-Kommunikation,
5. Latenzzeitmessung.

Im Praxisteil werden die Konzeptentwicklung mit der Komponentenauswahl, die anschließende Montage des Versuchsaufbaus sowie die Inbetriebnahme beschrieben. Erläutert werden die Herstellung einer Datenübertragung sowie die Bedienung der Videokamera, der Servomotoren des Kamerastativs, der Steuerungseinheit und des Raspberry Pi. Ein 5G-Modem/Router ermöglicht die Übermittlung der Videoaufnahmen via 5G-Mobilfunkstandard an eine Workstation. Von dieser aus erfolgt anschließend die Wiedergabe mittels VR-Brille. Durch Rückgabe von Bewegungsdaten sollen mittels der Ansteuerung der Servomotoren am Stativ die Drehung und Neigung der Kamera realisiert werden. Basierend auf dem Versuchsaufbau soll ein Messsystem geschaffen werden, das die Durchführung von Latenzzeitmessungen und die nachfolgende Datenauswertung erlaubt. Den Abschluss bilden eine Übersicht der erreichten Ergebnisse und der Limitationen sowie ein Ausblick auf zukünftig geplante Verbesserungen der 5G-Infrastruktur an der FH CAMPUS 02.

2 5G-MOBILFUNKSTANDARD

In diesem Kapitel wird ein kurzer Einblick in die 5G-Mobilfunktechnik gegeben, beginnend mit der Historie des Mobilfunkstandards, gefolgt von der aktuellen Verbreitung und der Wachstumsprognose des Mobilfunkstandards der fünften Generation. Weiterführend wird auf die Entwicklung des Standards und die Definition der Hauptanwendungsszenarien von 5G eingegangen, die Funknetzarchitektur, *Network Slicing* und *Multi-Access Edge Computing* eingegangen. Das Kapitel wird mit einem Ausblick hinsichtlich der AR/VR-Unterstützung geschlossen.

2.1 Historie der Mobilfunkstandards

Wie in Abb. 1 chronologisch dargestellt, hat sich der Mobilfunk seit den 1980er Jahren von Generation zu Generation in Datengeschwindigkeit und Funktionsumfang rapide weiterentwickelt. Während die erste Generation (1G) für Sprachanrufe mittels Analogtechnik konzipiert war, verfügte die zweite Generation (2G) bereits über Digitaltechnik und begrenzte Daten- und Messaging-Funktionen. Zur Jahrtausendwende wurde die dritte Generation (3G, 3.5G) entwickelt, um für Millionen von mobilen Geräten (Smartphones, Laptops etc.) mobilen Breitbandzugang mit einer Downloadgeschwindigkeit von mehreren Megabit pro Sekunde (Mbit/s, Mbps) bereitzustellen. Die vierte Generation der Mobilfunktechnologie (4G, *Long Term Evolution* oder LTE) mit einer Datenübertragungsgeschwindigkeit von bis zu 110 Mbit/s wurde in Europa ab dem Jahr 2009 kommerziell vermarktet. Die Einführung von *LTE Advanced* (LTE-A, 4.5G) erfolgte 2016, mit einer gesteigerten Datenübertragungsgeschwindigkeit von bis zu 300 Mbit/s. Während der Mobilfunkstandard der dritten Generation (3G, 3.5G) in zehn Jahren weltweit 2,5 Milliarden Nutzern Zugang zu Informationen, Anwendungen und Diensten ermöglichte, erreichte LTE die gleiche Anzahl an Nutzern bereits innerhalb von fünf Jahren.¹

¹ Vgl. Brooksby/Murphy/Featherston/Thomas/Kangru/Vaez-Ghaemi/Wei Koo/Yamany (2021), S. xv-xvii.

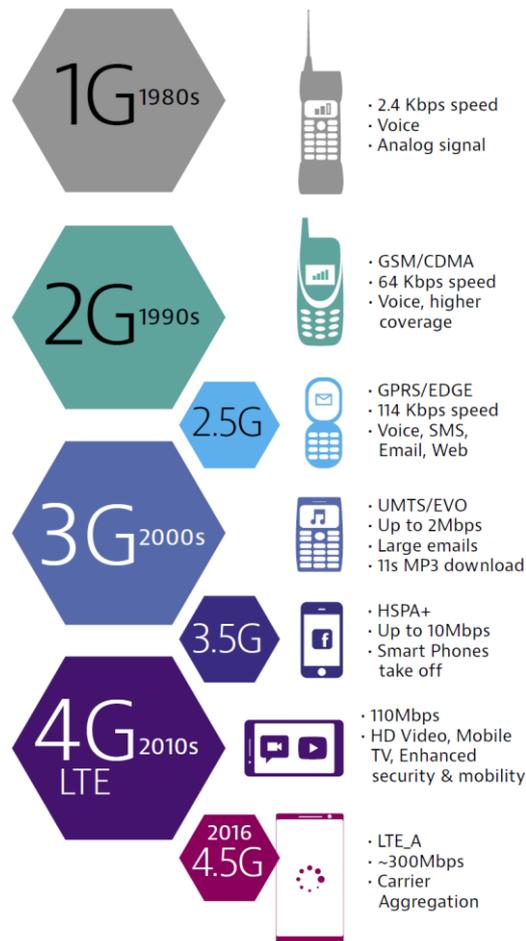


Abb. 1: Entwicklung der Mobilfunktechnologie von 1980er bis 2017,
Quelle: Understanding 5G: A Practical Guide to Deploying and Operating 5G Networks (2021), S. xvi.

2.2 Verbreitung und Wachstum der 5G-Infrastruktur

Durch weitere neue Funktionalitäten, die der neue Mobilfunkstandard der fünften Generation (5G, NR, *New Radio*) bietet, werden ein noch schnelleres Wachstum und eine noch schnellere Verbreitung erwartet als es bereits mit LTE der Fall war.² Wie in Abb. 2 dargestellt, wird weltweit in allen Wirtschaftsräumen eine Vervielfachung der 5G-Anschlüsse prognostiziert, wobei Asien eine Vorreiterrolle einnimmt. Mit Fokus auf Europa (Abb. 3) bedeutet dies, dass sich der Anteil der 5G-Mobilfunkanschlüsse bis 2025 mehr als verzehnfachen wird.³

² Vgl. Brooksby/Murphy/Featherston/Thomas/Kangru/Vaez-Ghaemi/Wei Koo/Yamany (2021), S. xvii

³ Vgl. Statista (Hrsg.) (2021), S. 10, 13.

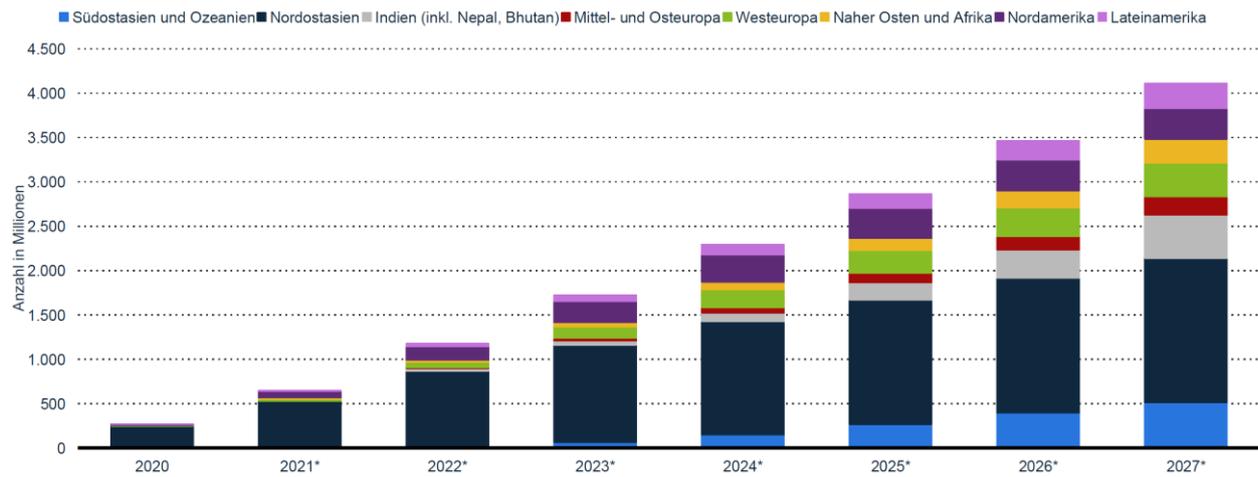


Abb. 2: Prognose zur Anzahl der 5G-Anschlüsse weltweit nach Region von 2020 bis 2027 (in Millionen), Quelle: Statista-Dossier Mobilfunkstandard-5g (2021), S. 10.

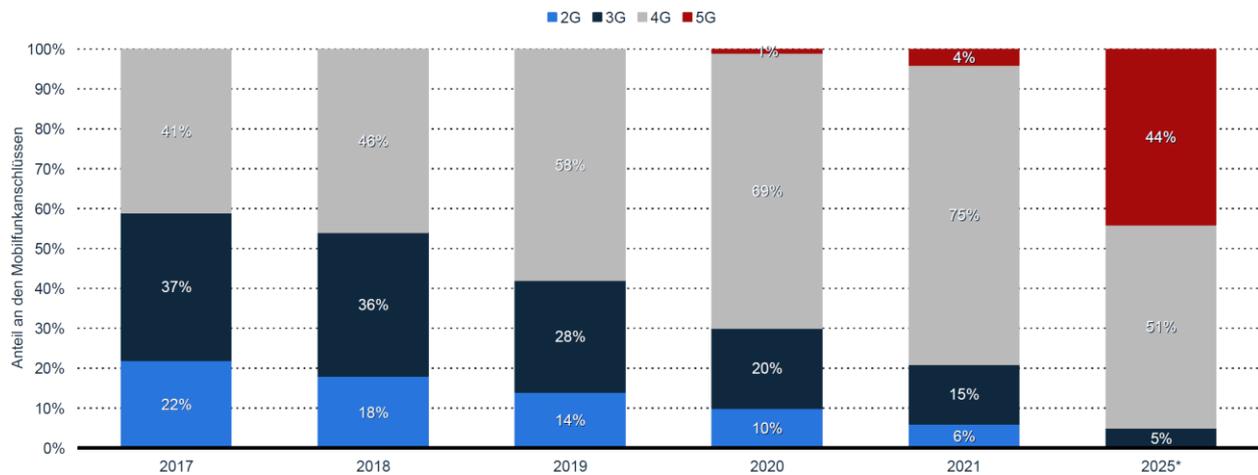


Abb. 3: Verteilung der Mobilfunkanschlüsse in Europa nach Mobilfunkstandard in den Jahren 2017 bis 2021 und Prognose 2025, Quelle: Statista-Dossier Mobilfunkstandard-5g (2021), S. 13.

2.3 Entwicklung und Hauptanwendungsszenarien von 5G

Im November 2017 wurde von der *International Telecommunication Union* (ITU) der Bericht *International Mobile Telecommunications for 2020 and beyond* (IMT-2020) vorgestellt, mit dem die ersten Anforderungen an 5G-Netzwerke definiert wurden. Basierend auf diesem Standard begann die Kooperation *3rd Generation Partnership Project* (3GPP) mit der Einführung des 5G-New-Radio-Standards. Neben 3GPP arbeiten noch andere Organisationen an der technischen Umsetzung des IMT-2020-Standards, auf die jedoch in dieser Arbeit nicht weiter eingegangen wird.⁴ Mit Release 15 begann 3GPP mit der Definition von Phase 1 des neuen Mobilfunkstandards der fünften Generation (5G).⁵ Stand Oktober 2022 wurden Release 15, Release 16 und Release 17 in Bezug auf 5G von 3GPP veröffentlicht.

⁴ Vgl. Brooksby/Murphy/Featherston/Thomas/Kangru/Vaez-Ghaemi/Wei Koo/Yamany (2021), S. xvii, Kim (2020), S. 10.

⁵ Vgl. 3GPP (Hrsg.) (2019), S. 9.

In Release 15 sind folgende Hauptanwendungsszenarien definiert:

- **Enhanced Mobile Broadband (eMBB):** Nachfolger von 4G mit erhöhter Datenübertragungsgeschwindigkeit und Abdeckung zusätzlicher Einsatzbereiche und Szenarien, wie Innen- und Außenbereich, städtische und ländliche Gebiete, Büro und Eigenheim, lokale und weiträumige Gebietsabdeckung, Hochgeschwindigkeitsfahrzeuge, *Ultra-High Definition (UHD) Video*, *Augmented Reality (AR)* und *Virtual Reality (VR)*.
- **Critical Communications (CC) and Ultra Reliable and Low Latency Communications (URLLC):** Für Szenarien, die auf sehr geringe Latenz und sehr hohe Verfügbarkeit angewiesen sind, wie industrielle Automation, öffentliche Sicherheit, autonomes Fahren und Smart-Grid. Als Beispiel wird für die Fernsteuerung von Prozessautomation eine 99,999%ige Zuverlässigkeit, eine Datenübertragungsgeschwindigkeit von bis zu 100 Mbit/s und eine End-to-End-Latenzzeit von 50 ms erwartet.
- **Massive Machine Type Communication (mMTC) / Massive Internet of Things (mIoT):** Szenarien zur Unterstützung einer massiven Zahl an 5G-Systemen, im Speziellen von Internet-of-Things(IoT)-Geräten. Kennzahlen hierfür sind eine Gerätedichte von 1 Millionen Geräte pro km², eine Verfügbarkeit von über 99 % und einer Latenzzeit unter 10 s.
- **Flexible Network Operations:** Besondere Funktionalitäten, die das 5G-System bietet, wie *Network Slicing*, Skalierbarkeit, Mobilität, Sicherheit und Bereitstellung von Inhalten.⁶

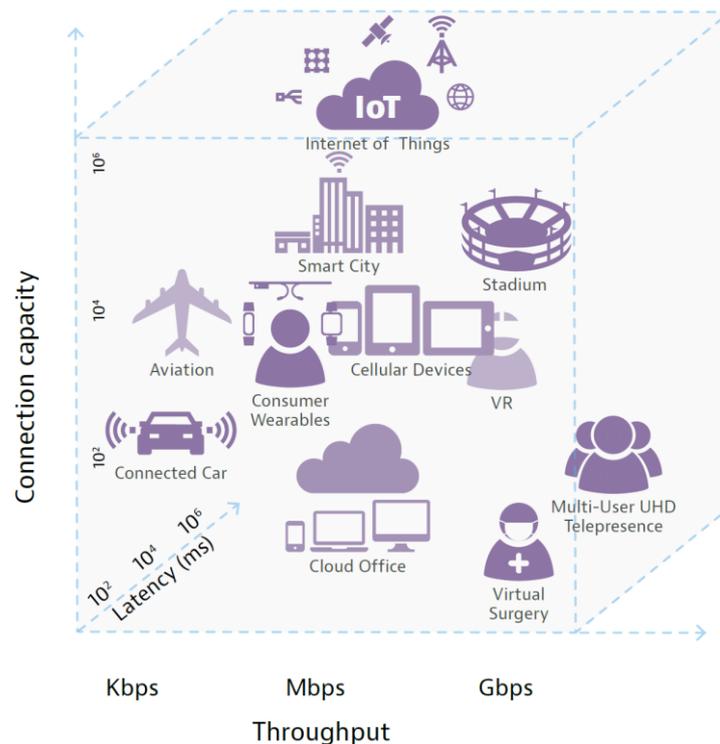


Abb. 4: 5G Hauptanwendungsszenarien nach International Telecommunication Union (ITU) International Mobile Telecommunications (IMT) Vision, Quelle: Understanding 5G: A Practical Guide to Deploying and Operating 5G Networks (2021), S. 23.

⁶ Vgl. 3GPP (Hrsg.) (2019), S. 9-10, Kim (2020), S. 19.

2.4 NSA- und SA-Architektur

Unabhängig von der Technologie setzt sich jede mobile Netzwerkarchitektur aus einem *Core Network* (CN) und einer Funkbasisstation zusammen, die wiederum aus einer *Base Station* und einer *Antenna Unit* besteht. Mittels der Funkbasisstationen wird das *Radio Access Network* (RAN) gebildet, in das sich mit einer Form von *User Equipment* (UE) eingewählt werden kann. Der Datenaustausch und die Dateninterpretation zwischen den Elementen erfolgt mittels genau definierter Protokollschichten, um einen korrekten Betrieb und Interoperabilität zwischen verschiedenen Anbietern zu gewährleisten.⁷

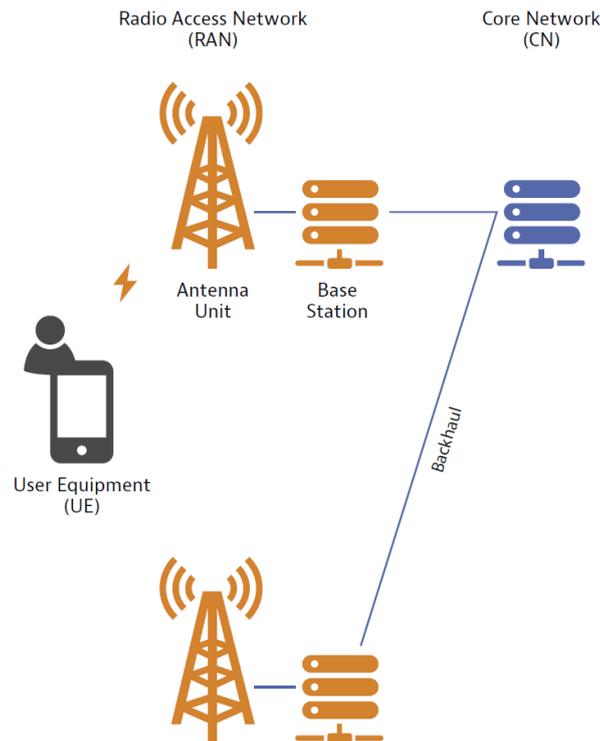


Abb. 5: Prinzipdarstellung einer mobilen Netzwerkarchitektur,
Quelle: Understanding 5G: A Practical Guide to Deploying and Operating 5G Networks (2021), S. 4.

Mit der Einführung von 5G wurden zwei Netzwerkarchitekturen definiert, die Non-Stand-Alone-Architektur (NSA) und die Stand-Alone-Architektur (SA). Die NSA-Architektur setzt auf die bestehende Infrastruktur auf. Unterstützt werden nur 4G-Dienste, jedoch mit 5G-Kapazitäten. Das Einwählen in das Netzwerk ist mittels 4G und 5G möglich. Diese Variante stellt einen temporären Schritt in Richtung ‚Full 5G Deployment‘ bzw. SA-Architektur dar. Die NSA-Architektur besteht aus dem *LTE Evolved Packet Core* (EPC), der die *Mobility Management Entity* (MME) und das *Serving Gateway* (S-GW) beinhaltet. Die RAN-Umgebung wird durch das *Evolved Universal Terrestrial Radio Access Network* (E-UTRAN) definiert, das sich aus LTE-eNB-Funkbasistationen (Evolved NodeB) und 5G-en-gNB-Funkbasisstation (Next generation NodeB) zusammensetzt.⁸

⁷ Vgl. Brooksby/Murphy/Featherston/Thomas/Kangru/Vaez-Ghaemi/Wei Koo/Yamany (2021), S. 3-4.

⁸Vgl. 3GPP (Hrsg.) (2019), S. 11.

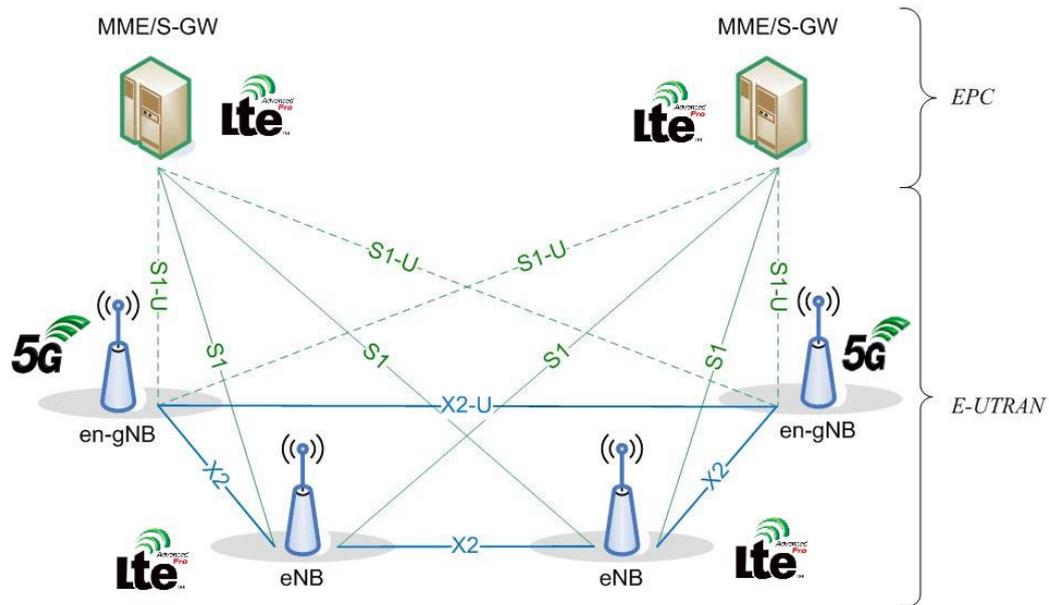


Abb. 6: Non-Stand-Alone-Architektur (NSA),
Quelle: 3GPP TR 21.915 version 15.0.0 Release 15 (2019), S 11.

Nur eine Stand-Alone (SA) Architektur, bestehend aus 5G Core (5GC) und gNB (Next generation NodeB) Funkbasisstation, unterstützt im vollem Umfang Dienste von 5G der Phase 1.⁹

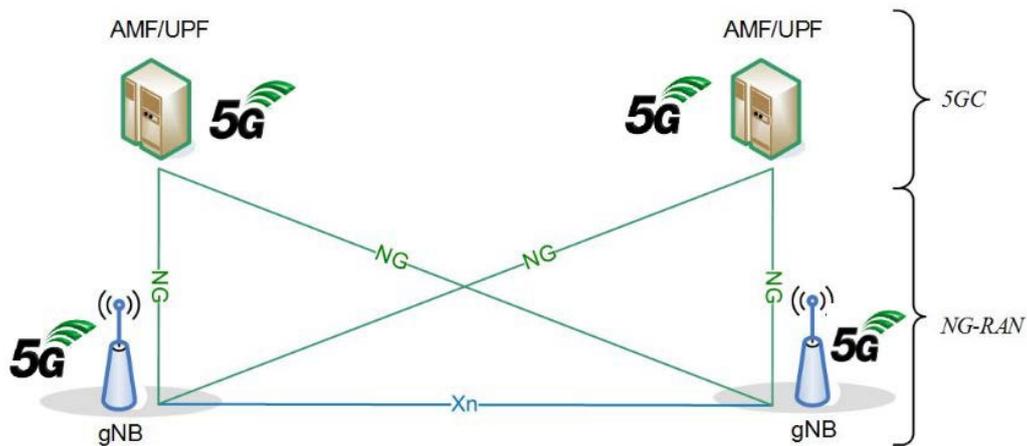


Abb. 7: Stand-Alone-Architektur (SA),
Quelle: 3GPP TR 21.915 version 15.0.0 Release 15 (2019), S 12.

2.5 Enhanced Mobile Broadband

Das *enhanced Mobile Broadband* (eMBB) stellt eine Erweiterung des 4G-Breitband dar und verfügt über höhere Datenübertragungsgeschwindigkeiten, bessere Anbindung und höhere Mobilität. Das Hauptkriterium bei eMBB ist der Durchsatz, der durch erfolgreiche Informationsübertragung über einen Kanal definiert ist. Da viele Einflussfaktoren auf den Durchsatz wirken, wie Bandbreite, Paketverlust,

⁹ Vgl. 3GPP (Hrsg.) (2019), S. 12.

Leistung, Latenz, Nutzerdichte, Kanaltyp, Netzwerkprotokoll und Netzwerktopologie, hängt die Definition im Einzelfall von Systemmodellen und Annahmen ab. Auf der Bitübertragungsschicht ist die Definition hingegen eindeutig: ‚Wie viele Daten werden pro Sekunde erfolgreich übertragen?‘, also Durchsatz (bps, bit/s) = Daten (bit) / Zeit (s). Die Bandbreite stellt im Vergleich eine theoretische Größe für die Datenübertragung über ein Medium dar. Der Durchsatz hingegen ist realitätsnaher, da Rauschen, Fehler, Interferenzen usw. berücksichtigt werden. In Abb. 8 ist das Verhalten von Bandbreite und Durchsatz über die Zeit zueinander veranschaulicht.¹⁰

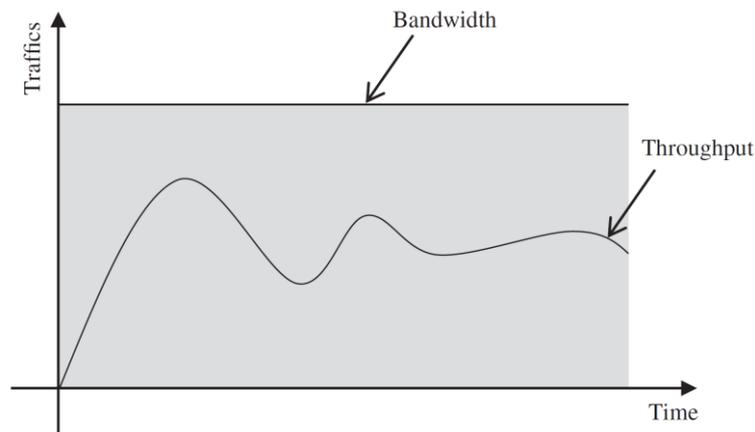


Abb. 8: Bandbreite (Bandwidth) vs Durchsatz (Throughput),
Quelle: Design and optimization for 5G wireless communications (2020), S. 241.

Die Gleichung (1.1) beschreibt eine vereinfachte Definition des Netzdurchsatzes pro Flächeneinheit. Daraus geht hervor, dass mehr verfügbare Bandbreite, eine Erhöhung der Zelldichte oder eine Verbesserung der spektralen Effizienz eine Erhöhung des Netzdurchsatzes zur Folge hat. Die spektrale Effizienz ist ein Maß zur Nutzung der Funkfrequenz und durch die Anzahl der Informationsbits definiert, die zuverlässig über den Kanal übertragen werden können¹¹

$Nt = \frac{B\rho_z}{\eta_s}$ (1.1)	$Nt/\text{bits}^{-1} \text{ km}^{-2}$	Netzdurchsatz mit Flächenbezug
	B/Hz	Bandbreite
	$\rho_z/\text{Zelle km}^{-2}$	Zelldichte
	$\eta_s/\text{bits}^{-1} \text{ Hz}^{-1} \text{ Zelle}^{-1}$	Spektrale Effizienz

Eine Vielzahl an Szenarien erfordern die Unterstützung einer sehr hohen Datenübertragungsgeschwindigkeit oder Nutzerdichte. In Abb. 9 werden einige mit ihren dazugehörigen Anforderungen aufgelistet.¹²

¹⁰ Vgl. Kim (2020), S 240.

¹¹ Vgl. Kim (2020), S. 240-241.

¹² Vgl. 3GPP (Hrsg.) (2022), S. 47.

	Scenario	Experienced data rate (DL)	Experienced data rate (UL)	Area traffic capacity (DL)	Area traffic capacity (UL)	Overall user density	Activity factor	UE speed	Coverage
1	Urban macro	50 Mbit/s	25 Mbit/s	100 Gbit/s/km ² (note 4)	50 Gbit/s/km ² (note 4)	10 000/km ²	20 %	Pedestrians and users in vehicles (up to 120 km/h)	Full network (note 1)
2	Rural macro	50 Mbit/s	25 Mbit/s	1 Gbit/s/km ² (note 4)	500 Mbit/s/km ² (note 4)	100/km ²	20 %	Pedestrians and users in vehicles (up to 120 km/h)	Full network (note 1)
3	Indoor hotspot	1 Gbit/s	500 Mbit/s	15 Tbit/s/km ²	2 Tbit/s/km ²	250 000/km ²	note 2	Pedestrians	Office and residential (note 2) (note 3)
4	Broadband access in a crowd	25 Mbit/s	50 Mbit/s	[3,75] Tbit/s/km ²	[7,5] Tbit/s/km ²	[500 000]/km ²	30 %	Pedestrians	Confined area
5	Dense urban	300 Mbit/s	50 Mbit/s	750 Gbit/s/km ² (note 4)	125 Gbit/s/km ² (note 4)	25 000/km ²	10 %	Pedestrians and users in vehicles (up to 60 km/h)	Downtown (note 1)
6	Broadcast-like services	Maximum 200 Mbit/s (per TV channel)	N/A or modest (e.g. 500 kbit/s per user)	N/A	N/A	[15] TV channels of [20 Mbit/s] on one carrier	N/A	Stationary users, pedestrians and users in vehicles (up to 500 km/h)	Full network (note 1)
7	High-speed train	50 Mbit/s	25 Mbit/s	15 Gbit/s/train	7,5 Gbit/s/train	1 000/train	30 %	Users in trains (up to 500 km/h)	Along railways (note 1)
8	High-speed vehicle	50 Mbit/s	25 Mbit/s	[100] Gbit/s/km ²	[50] Gbit/s/km ²	4 000/km ²	50 %	Users in vehicles (up to 250 km/h)	Along roads (note 1)
9	Airplanes connectivity	15 Mbit/s	7,5 Mbit/s	1,2 Gbit/s/plane	600 Mbit/s/plane	400/plane	20 %	Users in airplanes (up to 1 000 km/h)	(note 1)
<p>NOTE 1: For users in vehicles, the UE can be connected to the network directly, or via an on-board moving base station. NOTE 2: A certain traffic mix is assumed; only some users use services that require the highest data rates [2]. NOTE 3: For interactive audio and video services, for example, virtual meetings, the required two-way end-to-end latency (UL and DL) is 2-4 ms while the corresponding experienced data rate needs to be up to 8K 3D video [300 Mbit/s] in uplink and downlink. NOTE 4: These values are derived based on overall user density. Detailed information can be found in [10]. NOTE 5: All the values in this table are targeted values and not strict requirements.</p>									

Abb. 9: Leistungsanforderungen für Szenarien mit hoher Datenübertragungsgeschwindigkeit und Nutzerdichte, Quelle: 3GPP TS 22.261 version 17.10.0 Release 17 (2022), S. 48.

2.6 Ultra Reliable and Low Latency Communications

Da die aktuellen und zukünftigen Funktionalitäten von Ultra Reliable and Low Latency Communications (URLLC) die von 4G übersteigen ist eine andere Betrachtungsweise in Bezug auf zellulare Kommunikation und Netze erforderlich. URLLC-Systeme weisen zwei Schlüsselkriterien auf, eine niedrige Latenzzeit und eine hohe Zuverlässigkeit, wie zum Beispiel eine 99,999%ige erfolgreiche Empfangsrate und eine End-to-End Latenzzeit von 50 ms. In einem 5G-System wird nach 3GPP-Standard zwischen 3 Latenzzeitdefinitionen unterschieden: End-to-End (E2E), User Plane und Control Plane. Die Zeit, die ein Paket von Quelle bis zum Ziel benötigt wird als E2E-Latenzzeit bezeichnet, in ihr sind alle Verzögerungen wie Verarbeitungszeit, Warteschlangen, Neuübertragungen, Routing usw. enthalten. Die User Plane Latenzzeit beschreibt die Dauer zwischen dem Erreichen der Eingangspunkte der Service Data Unit (SDU) der OSI-Schichten 2 bzw. Data Link Layer (Sicherungsschicht) und OSI-Schicht 3 bzw. Network Layer (Vermittlungsschicht) im aktiven Zustand der Teilnehmergeräte. Die Control Plane Latenzzeit ist als die Übergangszeit vom batterieeffizientesten Zustand bis zum Beginn einer kontinuierlichen Datenübertragung

definiert. Die Mindestanforderungen in einem URLLC-System an die User Plane Latenzzeit sind 1 ms bzw. 20 ms an die Control Plane Latenzzeit. Die Zuverlässigkeit ist definiert als die Fähigkeit eines Netzes erfolgreich Daten innerhalb eines bestimmten Zeitraumes zu übertragen. Der 3GPP-Standard definiert die Zuverlässigkeit als die Fähigkeit, eine bestimmte Menge an Datenverkehr innerhalb einer vorgegebenen Zeit mit hoher Erfolgswahrscheinlichkeit zu übertragen.¹³

2.7 Massive Machine Type Communication

Massive Machine Type Communication (mMTC) bzw. *Massive Machine-to-Machine Communication* (M2M) ist eine Kommunikationsverbindung von einer Vielzahl an internetfähigen Geräten, die miteinander kommunizieren, mit oder ohne menschliches Eingreifen. Speziell ausgerichtet ist mMTC auf die Anforderungen von IoT-Diensten. Die grundlegenden Merkmale von IoT-Systemen sind Interkonnektivität, Heterogenität, dynamische Veränderungen und exorbitante Teilnehmerzahlen. Die Hauptkriterien für mMTC sind umfassende Kommunikationsverbindungen, kostengünstige Geräte, flächendeckende Netzabdeckung, eine niedrige Datenrate und ein extrem niedriger Stromverbrauch (*Ultra-low Power Consumption*). Für dieses Anwendungsgebiet wurden in der Vergangenheit schon mehr als zehn Standards entwickelt, einer davon ist das lizenzfreie *Long Range* (LoRa).¹⁴

2.8 Network Slicing

Network Slicing ist einer der wesentlichen Vorteile gegenüber den bisherigen RANs. Mit 5G wird damit ein neues Konzept von ‚*Service based Architecture*‘ eingeführt, basierend auf Netzwerkvirtualisierung. Das bedeutet, dass auf einer physikalischen Netzwerkarchitektur mehrere virtuelle Netzwerke (*Network Slices*) konfiguriert werden können. Die Anforderungen an die *Network Slices* sind stark von den unterschiedlichen Anforderungen der Industrie abhängig, wie Healthcare, Automotivindustrie, Landwirtschaft, Fabrikautomatisierung oder SmartGrid. Diese unterschiedlichen Anforderungen sind auch in den Hauptanwendungsszenarien (eMBB, URLLC, mMTC) eingebettet. Mit Abb. 10 wird ein Netz über die Schlüsselfähigkeiten von 5G gesponnen und ein Überblick über mögliche Gewichtungen gewährt.¹⁵

¹³ Vgl. Kim (2020), S. 304.

¹⁴ Vgl. Kim (2020), S. 343.

¹⁵ Vgl. Kim (2020), S. 232.

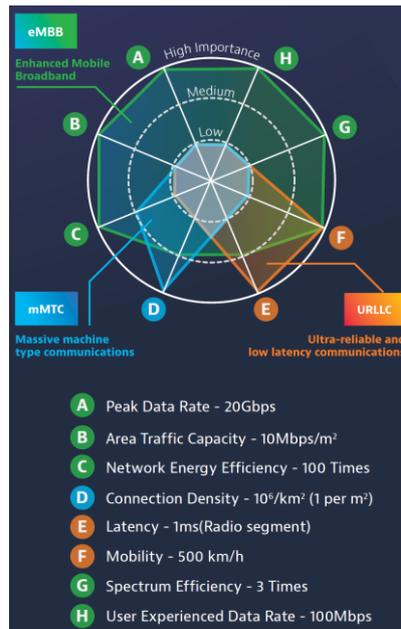


Abb. 10: Gewichtete Schlüsselfähigkeiten in verschiedenen Nutzungsszenarien, Quelle: VIAVI Solutions 5G Architecture and Specification, Primärquelle: ITU-R M.2083-0 (2015), S. 14, 15.

In Abb. 11 wird das Konzept von *Network Slicing* veranschaulicht. Es handelt sich um einen kosteneffizienten Customized-Service. Das bedeutet, dass ein Kunde ein Funknetzwerk mit spezifischen Anforderungen (z. B. Slice-Typ: eMBB, Datenrate: 1 Gbps, Zuverlässigkeit: 99,99 %, Latenz: 50 ms, Verbindungsdichte: 105 km²) bei einem Mobilfunkbetreiber anfragen kann. Der Mobilfunkbetreiber verwaltet die Netzinfrastruktur und stellt dem Kunden mittels *Network Slicing* ein logisch getrenntes virtuelles Netzwerk (*Network Slice*) als Dienst bereit. *Network Slicing* deckt zwei Arten von Diensten ab, Netzwerkverbindungsdienste (Latenz, Durchsatz, Mobilität, Zuverlässigkeit, Skalierbarkeit, Energieeffizienz usw.) und Netzressourcendienste (Datenmanagement, Big-Data-Analyse, Abrechnung, Ortung, Cloud-Speicher, verschiedene Security-Levels usw.).¹⁶

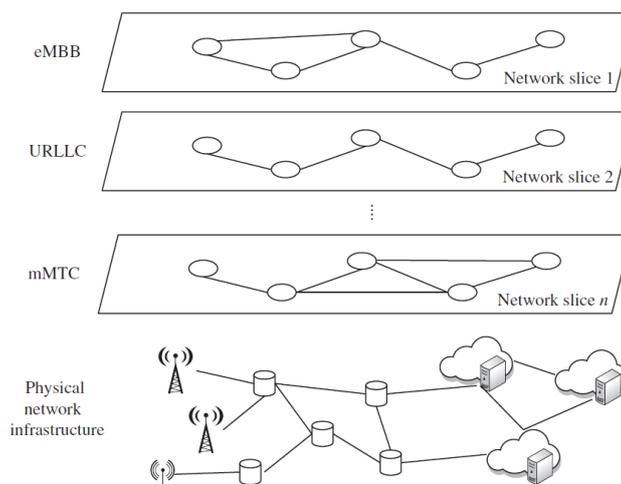


Abb. 11: Network-Slicing-Architektur, Quelle: Design and optimization for 5G wireless communications (2020), S. 204.

¹⁶ Vgl. Kim (2020), S. 203-204, 232.

2.9 Multi-Access Edge Computing

Edge-Computing in einem 5G-Netzwerk ist als Wirkverstärker für die vorhandenen 5G-Funktionalitäten zu verstehen und erweitert mit neuen bzw. optimierten Anwendungsfällen das Einsatzspektrum. Durch die Verknüpfung von Diensten und Rechenleistung, z. B. mit einer *Base Station*, lassen sich die End-to-End-Latenzzeit und die Belastung des Transportnetzes erheblich verringern, da der 5G-Core in der Kommunikationsschleife ausgeklammert wird.¹⁷

Hinsichtlich der Architekturgröße und Komplexität kann Multi-Access-Edge-Computing (MEC) in drei Kategorien unterteilt werden:

1. im Nahbereich oder lokal (3rd Generation Partnership Project, 3GPP),
2. mit Managementebene oder im Netzwerk eines Betreibers (Europäisches Institut für Telekommunikationsnormen, ETSI),
3. netzwerkbetreiberübergreifend (ETSI; Groupe Speciale Mobile Association, GSMA).

Der Verbindungsaufbau in einer 3GPP-Edge-Architektur kann in Abb. 12 nachverfolgt werden. Der *Edge-Enabler-Client* (EEC) stellt Unterstützungsfunktionen wie die *Edge-Application-Server*(EAS)-Erkennung für die *Application-Clients* bzw. dem *User-Equipment* bereit. Im *Edge-Data-Network* übernimmt diese Aufgabe der *Edge-Enabler-Server* (EES). Die Konfiguration für die Verbindung zwischen dem *Edge-Application-Server* und den *Application-Clients* wird vom *Edge-Configuration-Server* (ECS) an den *Edge-Enabler-Client* übermittelt. Nach erfolgtem Verbindungsaufbau kann der *Application-Data-Traffic* aufgenommen werden. Ein mit Hardwarekomponenten ausgestalteter Anwendungsfall der 3GPP-Edge-Architektur ist in Abb. 13 dargestellt.¹⁸

¹⁷ Vgl. Chitturi (2021), Online-Quelle [20.11.2022].

¹⁸ Vgl. Chitturi (2021), Online-Quelle [20.11.2022].

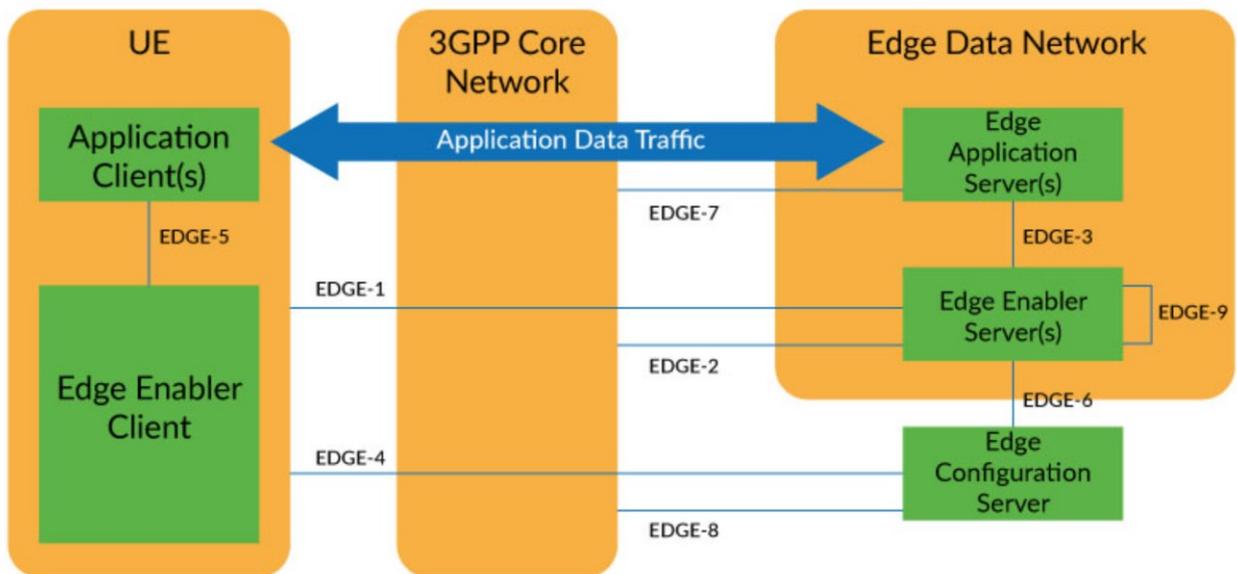


Abb. 12: 3GPP-Architektur zur Realisierung von Edge-Anwendungen, Quelle: <https://www.3gpp.org/news-events/3gpp-news/edge-sa6> [Stand: 20.11.2022].

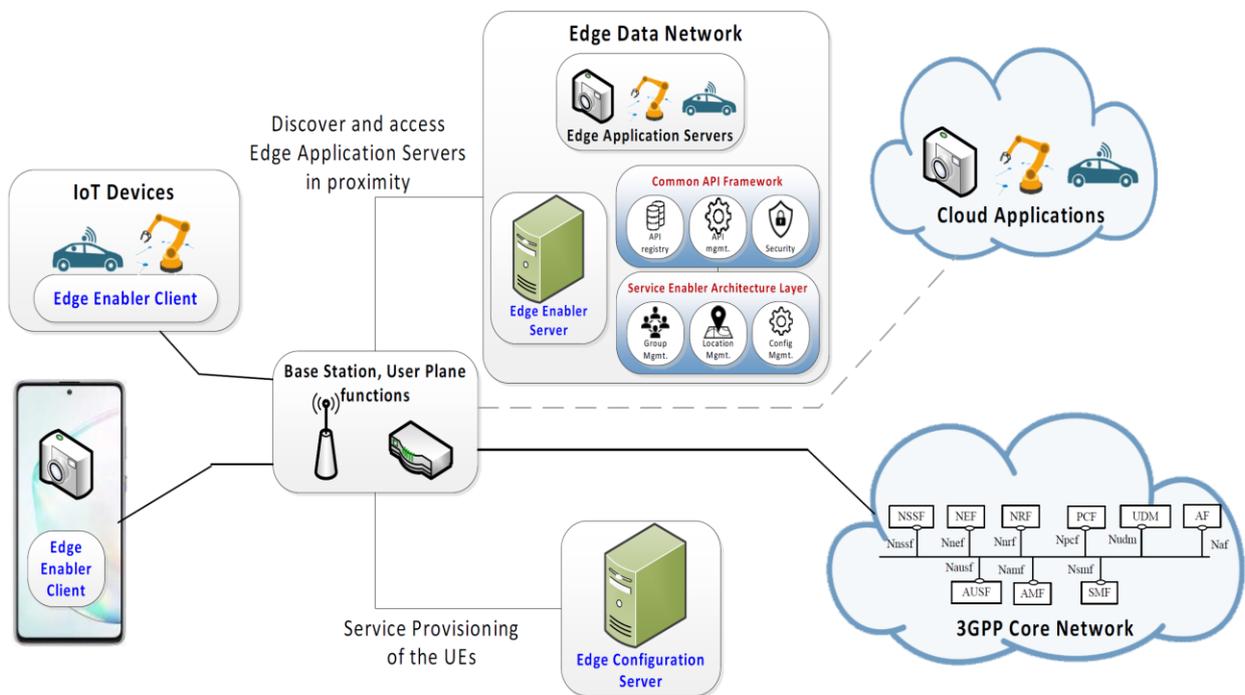


Abb. 13: 3GPP-Edge-Architektur, Quelle: 3GPP SA6 EDGEAPP Architecture for enabling Edge Applications (2021), S. 10.

In Abb. 14 sind drei vom ETSI beschriebene Bereitstellungsoptionen zur Erkennung von *Edge-Application-Servers* visualisiert.¹⁹

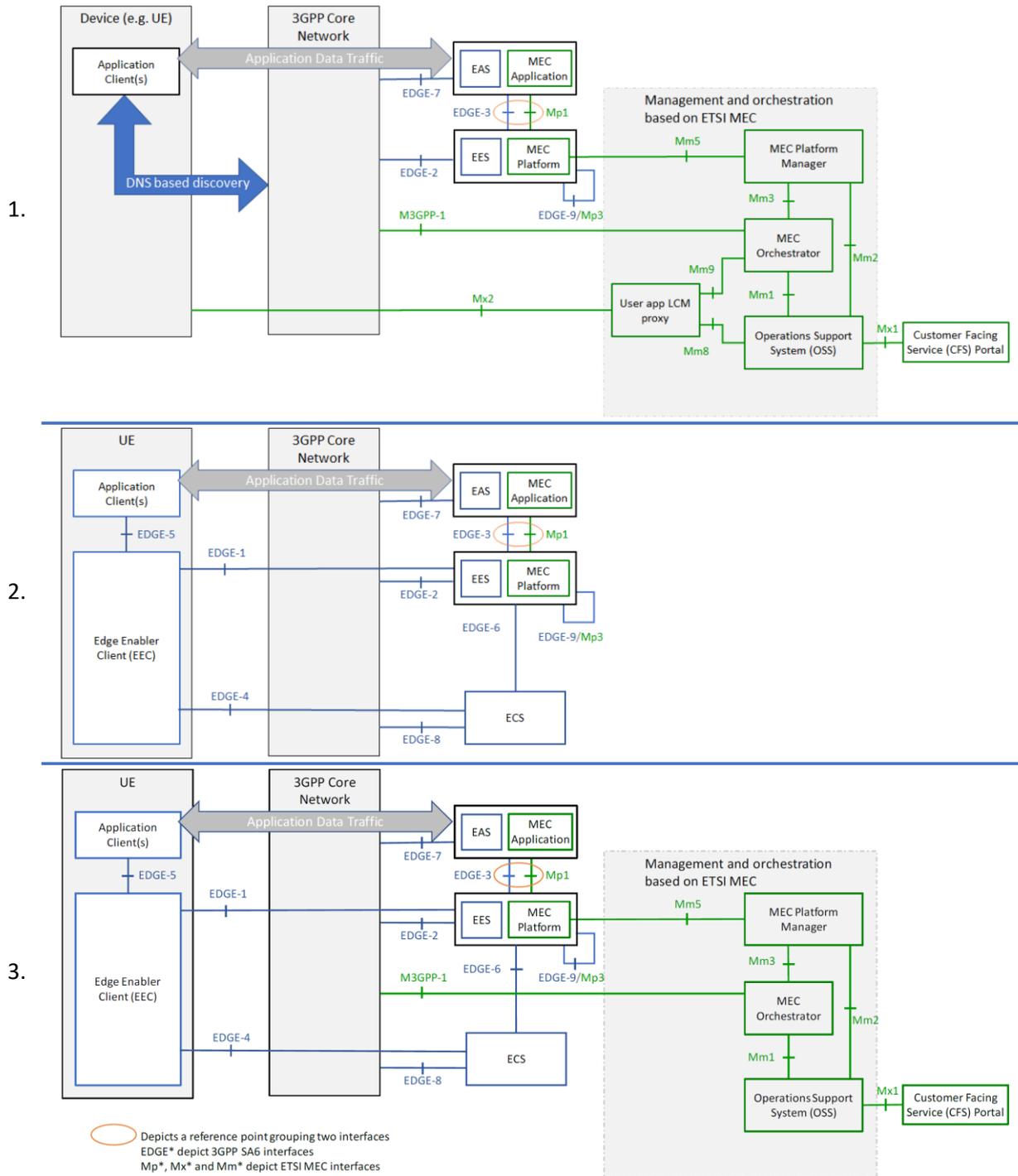


Abb. 14: 1. DNS-basierte Erkennung, 2. Edge-Enabler-Client-Erkennung, 3. Edge-Enabler-Client-Erkennung - Management und Orchestrierung auf Grundlage von ETSI ISG MEC, Quelle: ETSI White Paper #36 (2020), S. 11-12, (leicht modifiziert).

¹⁹ Vgl. Sprecher/Martin/Elloumi/Shen/Kuure/Feng/Neal/Garcia/Soloway/Yamamoto/Harmand/Lee/Chitturi/Mustapha/Reznik/Huang/Li/Rosbach/Costa/Huang/Sabella/Kekki/Moses/Mishra/Gunjal/Kleber/Hall/Featherstone/Chen/Yoon/Singh (2020), S. 11-13.

Edge-Node-Sharing

Unter Edge-Node-Sharing wird die gemeinsame Nutzung von Edge-Standorten oder Kapazitäten verstanden. Hierbei können mehrere Edge-Data-Networks in der eigenen Organisation miteinander gekoppelt werden. Alternativ kann die Dienstleistung von einem externen Netzwerkbetreiber erbracht werden. In Abb. 15 ist ersichtlich, wie ein Endnutzer im Netzwerk des Betreibers B eingeloggt ist und einen Service beanspruchen möchte. Da am Nutzerstandort keine eigenen Ressourcen verfügbar sind, werden diese bei Partner A angefordert und der Endnutzer wird mit dem *Edge-Application-Servers* des Netzwerkbetreibers verbunden.²⁰

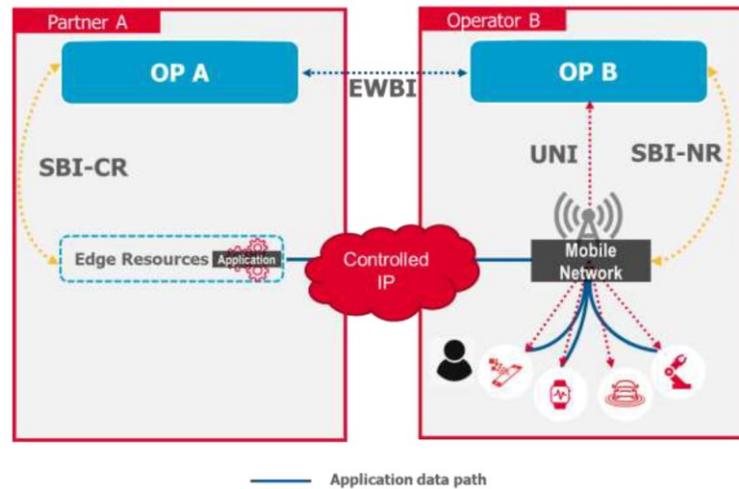


Abb. 15: Edge Node Sharing, Quelle: GSMA Operator Platform Telco Edge Requirements Version 1.0 (2021), S. 33.

Edge-Node-Sharing-Funktionalitäten sind in unterschiedlichem Umfang in den Architekturen aller drei Standard-Entwicklungs-Organisationen (Standard Development Organisations, SDO) enthalten. Bei der 3GPP-Edge-Architektur sind sie in EDGE-9 enthalten, die in Abb. 16 und Abb. 12 dargestellt ist.²¹

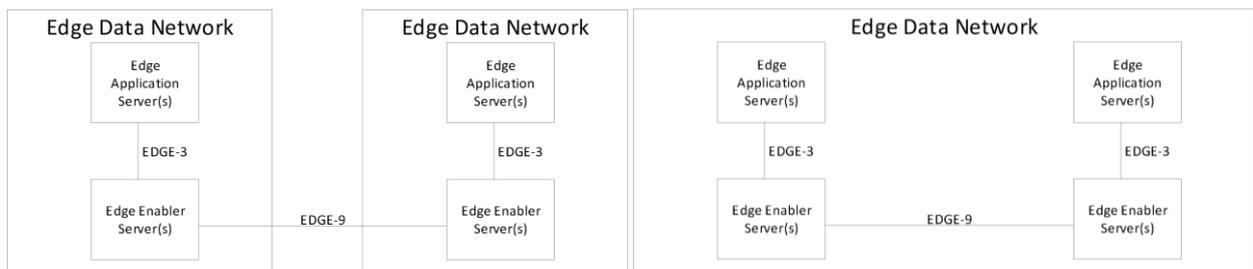


Abb. 16: 3GPP *Inter-Edge-Data-Network* EDGE-9 und *Inter-Edge-Data-Network* EDGE-9, Quelle: 3GPP TS 23.558 version 17.5.0 Release 17 (2022), S 27.

²⁰ Vgl. GSM Association (Hrsg.) (2021), S. 33-34.

²¹ Vgl. 3GPP (Hrsg.) (2022), S. 27.

Ein Gesamtbild einer MEC-Architektur im Endausbau mit Integration aller Teilaspekte der drei Organisationen ist in Abb. 17 ersichtlich.²²

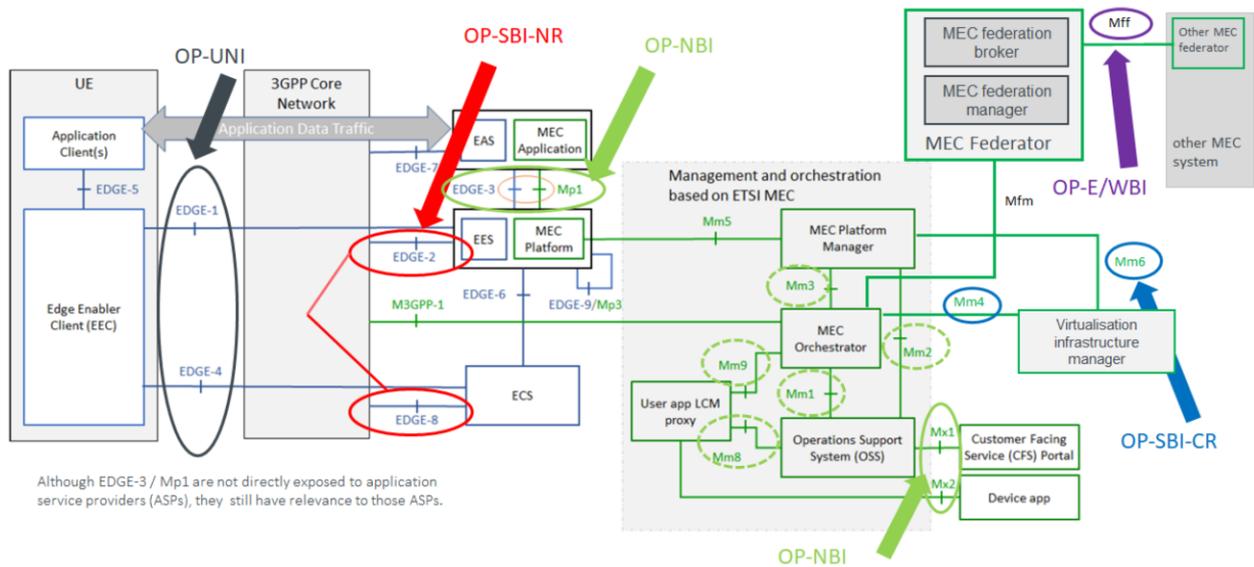


Abb. 17: Standards Development Organisation (SDO) übergreifendes Mapping der Operator-Plattform(OP) Architektur
Quelle: ETSI White Paper No. 49 (2022), S. 13.

In Abb. 18 ist ein Zukunftsszenario dargestellt, in dem multiple MEC-Systeme verschiedener Netzbetreiber miteinander vernetzt sind.²³

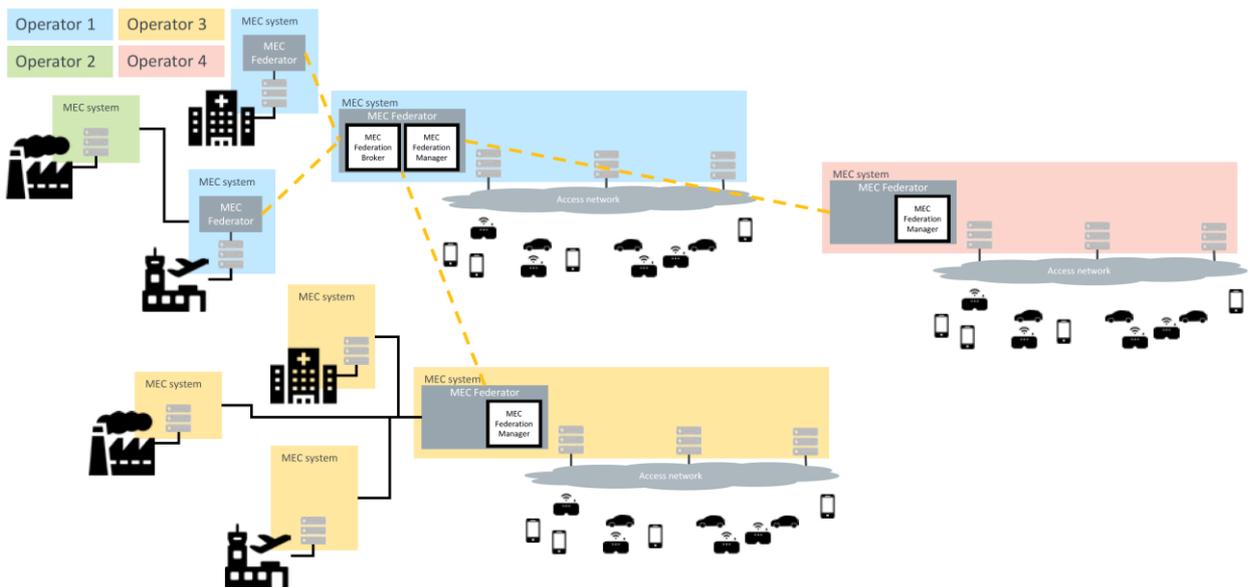


Abb. 18: Gesamtbild eines vernetzten MEC-Systems, Quelle: ETSI White Paper No. 49 (2022), S. 19.

²² Vgl. Suzuki/Joh/Lee/Featherstone/Sprecher/Sabella/Oliver/Shailendra/Graneli/Costa/Chen/Nieminen/Berzin/Naim (2022)

²³ Vgl. Suzuki/Joh/Lee/Featherstone/Sprecher/Sabella/Oliver/Shailendra/Graneli/Costa/Chen/Nieminen/Berzin/Naim (2022), S. 19.

2.10 High Data Rate and Low Latency AR/VR

Die Kombination von hoher Datenübertragungsgeschwindigkeit und niedriger Latenzzeit, wie sie für AR und VR benötigt wird, stellt sehr hohe Anforderungen an ein 5G-Netzwerk. In diesem Unterkapitel wird ein Ausblick darauf gezeichnet, in welcher Weise 5G solche Anwendungen in Zukunft unterstützen soll. Von 3GPP wurden dazu zwei Latenzzeiten definiert, *Motion-to-Photon* und *Motion-to-Audio*. Erstgenanntes beschreibt die Latenz zwischen Kopfbewegung und Bildupdate am Bildschirm des VR-Headsets. Zweitgenanntes bemisst die Zeit von Kopfbewegung bis zum Zeitpunkt, in dem die Schallwellen der Lautsprecher die Ohren erreichen. Die Motion-to-Photon-Latenzzeit soll sich im Bereich von 7 bis 15 ms befinden, mit einer Datenübertragungsgeschwindigkeit von bis zu 1 Gbit/s, um eine 8k-Auflösung zu unterstützen. Die Motion-to-Audio-Latenzzeit soll maximal 20 ms betragen. Zur Unterstützung von Kommunikation während interaktiver Aufgaben bzw. Tätigkeiten soll die Audioverzögerung (*One-way Mouth-to-Ear*) 100 ms betragen. Die angeführten Spezifikationen sind für den kabelgebundenen Betrieb eines VR-Headsets gedacht, um Berechnungen und Rendering auf ein lokales Gerät (UE) auszuführen. In Abb. 19 ist eine Übersicht über Dienste und dazugehörige Kennzahlen dargestellt, für die hohe Datenübertragungsgeschwindigkeiten und niedrige Latenzzeiten benötigt werden.²⁴

Use Cases	Characteristic parameter (KPI)			Influence quantity		
	Max allowed end-to-end latency	Service bit rate: user-experienced data rate	Reliability	# of UEs	UE Speed	Service Area (note 2)
Cloud/Edge/Split Rendering (note 1)	5 ms (i.e. UL+DL between UE and the interface to data network) (note 4)	0,1 to [1] Gbit/s supporting visual content (e.g. VR based or high definition video) with 4K, 8K resolution and up to 120 frames per second content.	99,99 % in uplink and 99,9 % in downlink (note 4)	-	Stationary or Pedestrian	Countrywide
Gaming or Interactive Data Exchanging (note 3)	10ms (note 4)	0,1 to [1] Gbit/s supporting visual content (e.g. VR based or high definition video) with 4K, 8K resolution and up to 120 frames per second content.	99,99 % (note 4)	≤ [10]	Stationary or Pedestrian	20 m x 10 m; in one vehicle (up to 120 km/h) and in one train (up to 500 km/h)
Consumption of VR content via tethered VR headset (note 6)	[5 to 10] ms (note 5)	0,1 to [10] Gbit/s (note 5)	[99,99 %]	-	Stationary or Pedestrian	-
<p>NOTE 1: Unless otherwise specified, all communication via wireless link is between UEs and network node (UE to network node and/or network node to UE) rather than direct wireless links (UE to UE).</p> <p>NOTE 2: Length x width (x height).</p> <p>NOTE 3: Communication includes direct wireless links (UE to UE).</p> <p>NOTE 4: Latency and reliability KPIs can vary based on specific use case/architecture, e.g. for cloud/edge/split rendering, and may be represented by a range of values.</p> <p>NOTE 5: The decoding capability in the VR headset and the encoding/decoding complexity/time of the stream will set the required bit rate and latency over the direct wireless link between the tethered VR headset and its connected UE, bit rate from 100 Mbit/s to [10] Gbit/s and latency from 5 ms to 10 ms.</p> <p>NOTE 6: The performance requirement is valid for the direct wireless link between the tethered VR headset and its connected UE.</p>						

Abb. 19: Kennzahlen für Dienste mit hoher Datenübertragungsgeschwindigkeit und niedriger Latenzzeit, Quelle: 3GPP TS 22.261 version 17.10.0 Release 17 (2022), S. 58.

²⁴ Vgl. 3GPP (Hrsg.) (2022), S. 57.

3 ANBINDUNG DER HARDWARE-KOMPONENTEN

In Kapitel 3 wird erläutert, wie die Hardware-Komponenten des Versuchsaufbaus miteinander auf Kommunikationsebene verbunden sind. Die IP-Kameras sind mittels *Universal Serial Bus* (USB) 2.0 mit dem Einplatinenrechner Raspberry Pi 4 verbunden und die Ansteuerung der Motorsteuerung erfolgt mittels *Inter Integrated Bus* (I2C). Auf die Pulsweitenmodulation (PWM) für die Positionssteuerung der analogen Servo-Motoren wird nicht näher eingegangen.

3.1 Inter Integrated Circuit Bus I²C

Der I²C ist der gebräuchlichste Peripheriebus und in einer Vielzahl von Geräten implementiert. Die Verbindung erfolgt durch vier Leitungen: Masse, Spannungsversorgung, Datenleitung (SDA, *Serial Data-Line*) und Taktleitung (SCL, *Serial Clock Line*). Ein I²C-Bussystem ist mit einer räumlichen Ausdehnung von wenigen Metern begrenzt. Jedes Element im Bussystem verfügt über eine eigene Adresse und kann entweder Sender (Transmitter) oder Empfänger (Receiver) sein. Die Organisation der Teilnehmer erfolgt nach dem Master-Slave-Prinzip, wobei auch ein Multimasterbus realisierbar ist. Aufgrund der historischen Entwicklung sind unterschiedliche Betriebsarten entstanden, deren Unterstützung vom verbauten I²C-Chip abhängig ist. Am verbreitetsten sind hierbei der Standard- und der Fast-Mode, die auch vom Raspberry Pi unterstützt werden.²⁵

Modus	Datenrate
Standard Mode	100 kBit/s
Fast Mode	400 kBit/s
Fast Mode Plus	1 MBit/s
High Speed Mode	3,4 MBit/s
Ultra Fast Mode	5 MBit/s

Abb. 20: Betriebsarten des I²C-Bussystems, Quelle: Raspberry Pi - das technische Handbuch (2019), S. 267.

3.2 Universal Serial Bus USB

Mit der Einführung des USB wurde eine universelle, externe Schnittstelle für alle Peripheriegeräte geschaffen. Der Bus lässt sich über USB-Hubs fast beliebig erweitern, die Geräteidentifikation wird automatisiert durchgeführt und Verbindungen lassen sich während des laufenden Betriebs hinzufügen. Mit jeder neuen Spezifikation hat sich die Datenübertragungsgeschwindigkeit erhöht und die Eigenschaften haben sich um neue Funktionalitäten erweitert. Dazu zählen unter anderem die Audio- und Videoübertragung und eine Stromversorgung (USB PD, *Power Delivery*) mit bis zu 100 W.²⁶

²⁵ Vgl. Dembowski (2019), S. 266-268.

²⁶ Vgl. Schnabel (2020), S. 290-314.

USB-Version	USB 1.0/1.1		USB 2.0	USB 3.0	USB 3.1
	Low-Speed	Full-Speed	High-Speed	Super-Speed	Super-Speed-Plus
Symbolrate	1,875 MBit/s	15 MBit/s	600 MBit/s	5 GBit/s	10 GBit/s
Datenrate (brutto)	1,5 MBit/s	12 MBit/s	480 MBit/s	4 GBit/s	-
Datenrate (theoretisch)	188 kByte/s	1,5 MByte/s	60 MByte/s	600 MByte/s	1.200 MByte/s
Datenrate (netto)	ca. 150 kByte/s	ca. 1 MByte/s	ca. 35 MByte/s	ca. 450 MByte/s	ca 800 MByte/s
Interface	UHCI/OHCI	UHCI/OHCI	EHCI	xHCI	-
Leitungslänge	5 m	5m	5m	3 m	1 m
Anwendungen	Maus, Tastatur	Audio	Video, Speichermedien		

Abb. 21: Übersicht: USB - Universal Serial Bus, Quelle: Computertechnik-Fibel (2020), S. 290-291, (leicht modifiziert).

4 VIDEOVERBINDUNG

In diesem Kapitel wird der Frage nachgegangen, wie eine Videoübertragung über ein Netzwerk mit möglichst niedriger Latenzzeit umgesetzt werden kann und welche technischen Möglichkeiten hierfür verfügbar sind.

4.1 Streaming-Protokolle

Grundsätzlich gibt es eine Vielzahl an Streaming-Protokollen, die sich in Eigenschaften und Anwendungsgebiet unterscheiden. Für die Aufgabenstellung in dieser Arbeit ist eine möglichst niedrige Latenzzeit das primäre Kriterium. Aus diesem Grund sind Streaming-Protokolle, die auf dem *Transmission Control Protocol* (TCP) aufbauen, ungeeignet. Die in TCP implementierten Sicherungselemente *Three-Way-Handshake* und Lösung von Paketverlust und Packetreihenfolge erhöhen zwar die Zuverlässigkeit, jedoch zulasten der Geschwindigkeit. Für Echtzeitanwendungen kommt daher das *User Datagram Protocol* (UDP) zum Einsatz, das durch den Verzicht auf diese Elemente das schnellere und dementsprechend unzuverlässigere Protokoll ist.²⁷

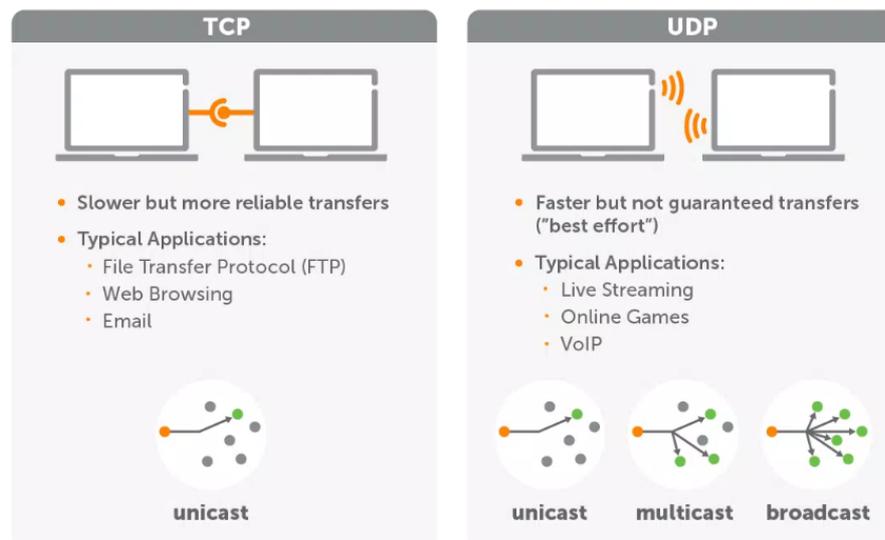


Abb. 22: Transmission Control Protocol (TCP) im Vergleich zum User Datagram Protocol (UDP),
Quelle: <https://www.wowza.com/blog/streaming-protocols>, [Stand: 17.09.2022].

Wie in Abb. 23 illustriert, streuen Streaming-Protokolle sehr weit in der Latenzzeit wie auch in den dafür vorgesehenen Anwendungsgebieten. Basierend auf dieser Zusammenstellung kommen nur die Protokolle RTSP (*Real-Time Streaming Protocol*), SRT (*Secure Reliable Transport*) und WebRTC (*Web Real-Time Communication*) für eine nähere Betrachtung in Frage.

²⁷ Vgl. Ruether (2022), Online-Quelle [17.09.2022], Vgl. Grigorik (2013), S. 315-316.

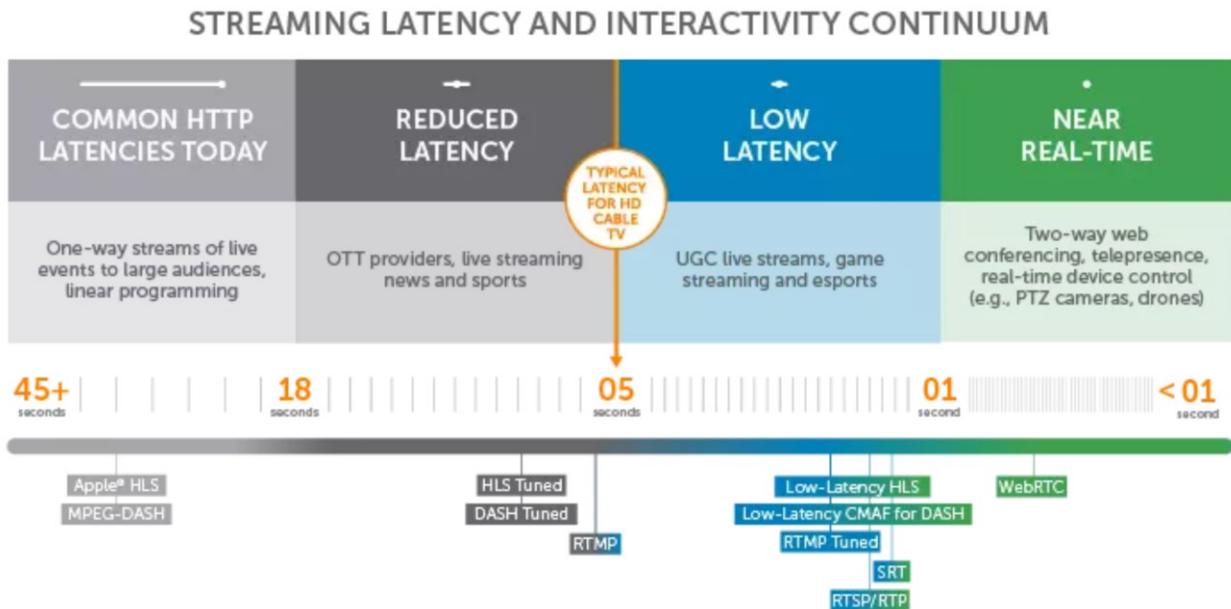


Abb. 23: Übersicht von Streaming-Protokollen, Quelle: <https://www.wowza.com/blog/streaming-protocols>, [Stand: 17.09.2022].

RTSP (Real-Time Streaming Protocol)

Im Jahr 1998 wurde RTSP veröffentlicht. Es ist somit eines der älteren Streaming-Protokolle für die Steuerung einer End-to-End-Internetübertragung von Video und Audio. In der Vergangenheit waren RTSP und *Real-Time Messaging Protocol* (RTMP) die dominierenden Protokolle für Streaminganwendungen über das Internet. Heutzutage werden sie zunehmend von anderen Protokollen abgelöst. Das Hauptanwendungsgebiet liegt heute in der Übertragung von Überwachungsvideos, da IP-Kameras das Protokoll nach wie vor uneingeschränkt unterstützen. Bei der Videoübertragung zu Endnutzern kommt dieses Protokoll in der Regel nicht mehr zum Einsatz. Üblicherweise wird TCP als Transportprotokoll verwendet, kann jedoch auch mit UDP als Transportprotokoll ausgeführt werden und weist eine Latenzzeit von zwei Sekunden auf.²⁸

SRT (Secure Reliable Transport)

SRT wurde von Haivision als ein Open-Source-, Low-Latency-Streaming-Protokoll entwickelt und 2013 das erste Mal öffentlich präsentiert. Die *SRT Alliance* wurde 2016 gegründet, um die Unterstützung des Protokolls in Anwendungen zu erhöhen. Das UDP-basierte Protokoll verfügt über einen Fehlerkorrekturmechanismus namens *Automatic Repeat ReQuest* (ARQ). Sobald der Empfänger eine Lücke im Datenstrom feststellt, wird ein *Negative Acknowledgement* (NACK) an den Sender übermittelt. Durch dieses Vorgehen müssen nur die fehlenden Pakete erneut gesendet werden. Die Latenzzeit liegt unter drei Sekunden, abhängig von den Einstellungen und der Inkaufnahme von Paketverlusten bis zu ‚Near Real-Time‘, jedoch wird das Protokoll zum aktuellen Zeitpunkt noch nicht weitgehend unterstützt z. B. Webbrowser.²⁹

²⁸ Vgl. Ruether (2022), Online-Quelle [17.09.2022], Vgl. Ruether (2022), Online-Quelle [17.09.2022].

²⁹ Vgl. Ruether (2022), Online-Quelle [17.09.2022], Vgl. Ruether (2021), Online-Quelle [17.09.2022].

WebRTC (Web Real-Time Communication)

Die Entwicklung von WebRTC startete als Google-Open-Source-Projekt und wird von der *Internet Engineering Task Force* (IETF) und von *World Wide Web Consortium* (W3C) als Standard geführt. Damit einhergehend ist die weitgehende Plattform- und Geräteunabhängigkeit sowie die Unterstützung aller großen Webbrowser gewährleistet. WebRTC gilt aktuell als das schnellste verfügbare Protokoll für Audio- und Videoübertragung und als die bevorzugte Wahl für End-to-End-Übertragung unter der Nutzung eines Protokolls. Zusätzlich lässt sich mit WebRTC eine Zwei-Wege-Kommunikation herstellen und eine *One-Way-Latency* (OWL) von weniger als 500 ms erreichen. Um eine sichere und zuverlässige Verbindung zu gewährleisten, verfügt WebRTC über einige unterstützende Protokolle, wie aus Abb. 24 hervorgeht. *Interactive Connectivity Establishment* (ISU) mit *Session Traversal Utilities for NAT* (STUN) und *Traversal Using Relays around NAT* (TURN) sind für den Aufbau und die Aufrechterhaltung der Peer-to-Peer-Verbindung mittels UDP erforderlich. Die Datenübertragungen von WebRTC werden standardmäßig mittels *Datagram Transport Layer Security* (DTLS) verschlüsselt und gesichert. Stau- und Flusskontrolle sowie das Multiplexen der Datenströme sorgen für eine teilweise zuverlässige Zustellung der Datenpakete. Diese Aufgaben werden von *Stream Control Transport Protocol* (SCTP) und *Secure Real-Time Transport Protocol* (SRTP) übernommen.³⁰

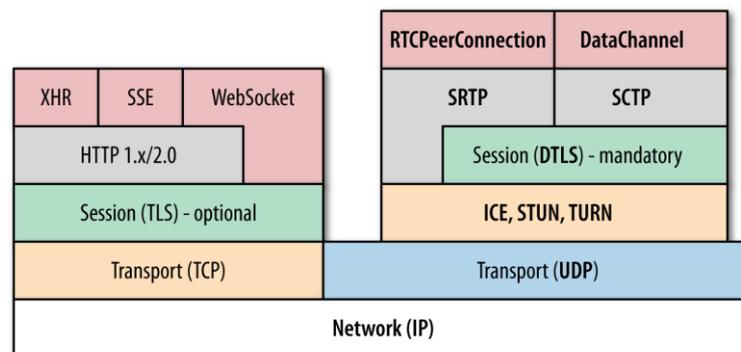


Abb. 24: WebRTC(Web Real Time Communication)-Protokoll-Stack, Quelle: High-performance browser networking (2013), S. 317.

³⁰ Vgl. Ruether (2022), Online-Quelle [17.09.2022], Vgl. Ruether (2021), Online-Quelle [17.09.2022], Vgl. Ruether (2022), Online-Quelle [17.09.2022], Vgl. Grigorik (2013), S. 3016-317.

5 OPTIMIERUNG VON 5G-KOMMUNIKATION

In diesem Kapitel werden die Optimierungsmöglichkeiten der 5G-Kommunikation behandelt. Die Möglichkeiten der mathematischen Optimierung sowie der Optimierung mittels *Artificial Intelligence* (Künstlicher Intelligenz), *Machine Learning* und *Deep Learning* werden nur der Vollständigkeit halber angeführt, in dieser Arbeit jedoch nicht weiter betrachtet. Der Fokus liegt auf grundlegenden Optimierungsoptionen einer 5G-Infrastruktur, bezogen auf die Hauptanwendungsszenarien *Enhanced Mobile Broadband* (eMBB), *Ultra Reliable and Low Latency Communications* (URLLC) und *Massive Machine Type Communication* (mMTC).

5.1 Enhanced Mobile Broadband

In einem 5G-System werden drei Ansätze verfolgt, um den Netzwerkdurchsatz und die Verfügbarkeit zu erhöhen: höhere und weitere Frequenzbänder, flexible Zeitmultiplexverfahren (*Flexible Time Division Duplexing*, *Flexible TDD*) und Vollduplexverfahren (*Full Duplex Techniques*). Zur Erhöhung der Zelldichte wird die Zellgröße reduziert und heterogene Netze werden eingeführt. Darüber hinaus werden zur Verbesserung der spektralen Effizienz neue Verfahren wie *Massive Multiple Input Output* (MIMO) und *Low Density Parity-Check Codes* (LDPC-Codes) eingeführt. In Abb. 25 ist eine Übersicht von Ansätzen dargestellt, mit deren Hilfe über einen oder mehrere Parameter (Bandbreite, Zelldichte, spektrale Effizienz) der Netzwerkdurchsatz gesteigert werden kann.³¹

5G techniques	Approaches			Remarks
	Bandwidth increase	Cell density increase	Spectral efficiency increase	
Small cell		v		Network planning
Heterogeneous networks		v		Network planning
mmWAVE	v			Physical layer
Massive MIMO			v	Physical layer
Flexible TDD			v	Physical layer
Full duplex	v			Physical layer
OFDM based waveforms			v	Physical layer
LDPC and polar codes			v	Physical layer
D2D link		v		Physical layer and Layer 2
Resource allocation and scheduling			v	Layer 2: MAC, RLC, PDCP
Carrier aggregation	v			Physical layer and Layer 2

Abb. 25: Herangehensweisen, um den Netzwerkdurchsatz zu erhöhen, Quelle: Design and optimization for 5G wireless communications (2020), S. 241.

³¹ Vgl. Kim (2020), S. 241-242.

5.2 Ultra Reliable and Low Latency Communications

Die Latenzzeit eines 5G-Systems setzt sich aus deterministischen und stochastischen Komponenten zusammen. Deterministische Komponenten im System sind das Übertragungszeitintervall (*Transmission Time Interval*, TTI), die Rahmengröße (*Frame Size*), die Dekodierungs- und Kodierungszeit usw. Der stochastische Anteil setzt sich aus erneuter Übertragung (*Retransmission Time*), Wartezeit (*Queuing Delay*), Synchronisation usw. zusammen. Zur Reduzierung der Latenzzeit verfügt 5G über einige neue Funktionen wie Rahmenstruktur bzw. *Frame Structure* (Mini-Slot, *Scalable TTI*, etc.), *Multi-Access Edge Computing* (MEC), *Network Slicing*, *Hybrid Automatic Repeat Request* (HARQ) und *Uplink (UL) Grant-Free Transmission*. Die Rahmenstruktur ist ein Schlüsselement zur Reduzierung der Bearbeitungs- und Übertragungszeit. Mini-Slot ermöglicht ein schnelles Umschalten der Übertragung zwischen Uplink und Downlink. Kombiniert mit einer skalierbaren Rahmenstruktur reduziert sich die Latenzzeit erheblich. *Hybrid Automatic Repeat Request* dient einerseits zur Erhöhung der Zuverlässigkeit und andererseits zur Reduzierung von erneuter Übertragung (*Retransmission*). Die Datenübertragung wird hierbei vom Empfänger mit Negative Acknowledge (NACK), zur Anfrage einer erneuten Übertragung, oder Reception Acknowledge (ACK) für Bestätigung der Übermittlung quittiert. Die minimale Verarbeitungszeit in 4G für ACK/NACK-Rückmeldung beträgt 3 ms, im Vergleich zu 5G mit 0,2 bis 1 ms für Downlink und 0,3 bis 0,8 ms für Uplink. *Multi-Access Edge Computing* (MEC) verringert die Latenz signifikant durch das Zuteilen von Rechenressourcen an den Netzgrenzen, zum Beispiel am Funkzugangnetz (Radio Access Network, RAN). *Network Slicing* ist eine Form der dezidierten Ressourcenzuweisung für gewisse Dienste. *Uplink (UL) Grant-Free Transmission* ermöglicht einen schnellen Uplink-Zugang, jedoch zu Kosten von Datenkollisionen und geringer Zuverlässigkeit. Die wesentlichen Schlüsselfunktionen für die Zuverlässigkeit sind *Massive Multiple Input Output* (MIMO) und *Low Density Parity Check-Codes* (LDPC-Codes). 5G-Netzwerke können die Zuverlässigkeit und Verfügbarkeit durch redundanten Netzzugang erhöhen und damit äußerst zuverlässige Netzwerke bereitstellen. URLLC leidet an zwei wesentlichen Kompromissen. Sowohl eine hohe Zuverlässigkeit als auch Energieeffizienz erfordern eine lange Latenzzeit. In Abb. 26 ist eine Übersicht von Techniken abgebildet, um Übertragungs- und Verarbeitungszeit zu verringern und in Summe eine niedrigere Latenzzeit zu erreichen.³²

³² Vgl. Kim (2020), S. 304-305.

5G techniques	Reducing transmission processing time	Reducing transmission time	Reducing reception processing time	Remarks
Short packet transmission (mini-slot, scalable TTI, etc.)	v	v	v	Deterministic components
MEC	v		v	
High-speed device design	v		v	Deterministic components
Network topology	v		v	
Network traffic management	v		v	Stochastic components
Network slicing	v		v	
HARQ		v		Stochastic components
UL grant-free transmission		v		Stochastic components
Scheduling policy	v	v	v	Stochastic components
Resource allocation	v	v	v	Stochastic components
Synchronization			v	Stochastic components

Abb. 26: Herangehensweisen, um Latenz zu verringern,
 Quelle: Design and optimization for 5G wireless communications (2020), S. 305.

5.3 Massive Machine Type Communication

In Bezug auf mMTC gibt es einige zentrale Entwicklungsherausforderungen, die in Zukunft für ein flächendeckendes Ausrollen bewältigt werden müssen:

- Umgang mit unvorhersehbaren und sporadischen Aktivitäten von mMTC-Geräten ('Grant-Free-Access-Mechanismus?').
- Einführung von Mechanismen des Ressourcenmanagements für eine enorme Anzahl an Netzwerkeilnehmern, um Durchsatzleistung und Latenzzeit zu gewährleisten.
- Schaffung von Leistungsmanagement-Mechanismen für IoT-Geräte, wie das *Sleep/Idlemode*-Verfahren.
- Entwicklung von Sendeleistungssteuerung und energiesparendes Funkfrequenzbanddesign, um eine hohe Batterielebensdauer zu ermöglichen.
- Etablierung von Funknetzzugangsverfahren für Massenzugriffe, um hohen Kollisionsraten, erhöhter Latenz und verringerter Energieeffizienz entgegenzuwirken.³³

In Abb. 27 werden grundsätzliche Designansätze für ein mMTC-Netzwerk beleuchtet. Hierbei werden sowohl die physikalischen Ebene (PHY) als auch die Medienzugriffskontrolle (*Medium Access Control*, MAC) betrachtet.

Approaches 5G techniques	Improving energy efficiency	Improving connection density	Improving radio resource utilization	Remarks
Power control schemes	v		v	MAC
Low power RF and BB design	v			PHY
Resource allocation and scheduling	v	v	v	MAC
Grant-free access mechanism		v	v	MAC
Radio access schemes		v	v	PHY and MAC

Abb. 27: mMTC Designansätze,
Quelle: Design and optimization for 5G wireless communications (2020), S. 350.

³³ Vgl. Kim (2020), S. 348-350.

6 LATENZZEITMESSUNG

In diesem Kapitel werden Methoden mit Vor- und Nachteilen zur Messung der Latenzzeit von 5G-Netzwerken erörtert. Zusätzlich wird ein Überblick mit ausgewählten Leistungsparametern der aktuellen österreichischen 5G-Infrastruktur gegeben. Das Kapitel wird mit einer Einführung in *Performance-Testing* für industrielle Automatisierung geschlossen.

6.1 Latenzzeit, Download- und Upload-Geschwindigkeit in Österreich

Dieses einführende Unterkapitel dient dazu, eine Basislinie zu ziehen und einen Ausblick zu ermöglichen, welche Erwartungen zum aktuellen Zeitpunkt an die 5G-Infrastruktur in Österreich bezüglich Latenzzeit, Down- und Upload-Geschwindigkeit gesetzt werden können. Wie aus Abb. 28, Abb. 29 und Abb. 30 hervorgeht, bietet 5G bezogen auf die Latenzzeit noch keine Vorteile gegenüber 4G. Im Gegensatz dazu zeigt sich bei den Download- und Upload-Geschwindigkeiten ein klarer Vorteil von 5G.

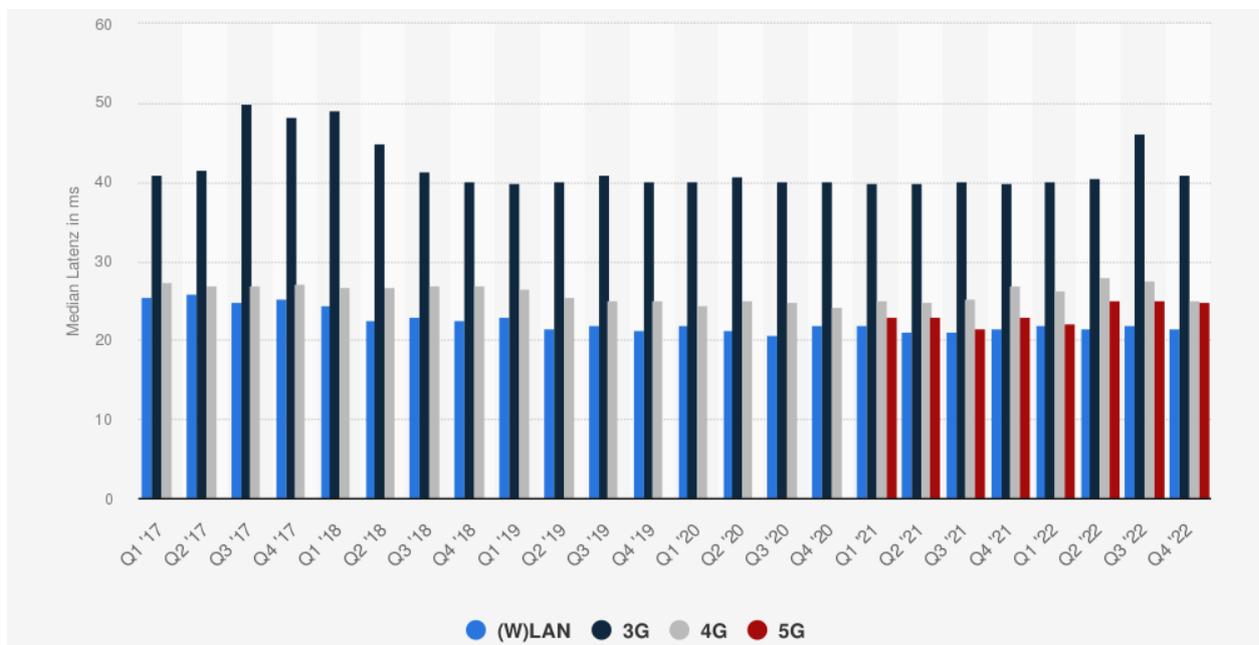


Abb. 28: Durchschnittliche Latenz der Mobilfunkstandards in Österreich vom 1. Quartal 2017 bis zum 4. Quartal 2022, Quelle: <https://de-statista-com.elibrary.campus02.at/statistik/daten/studie/1301773/umfrage/latenz-der-mobilfunkstandards-in-oesterreich/> [Stand: 30.01.2023].

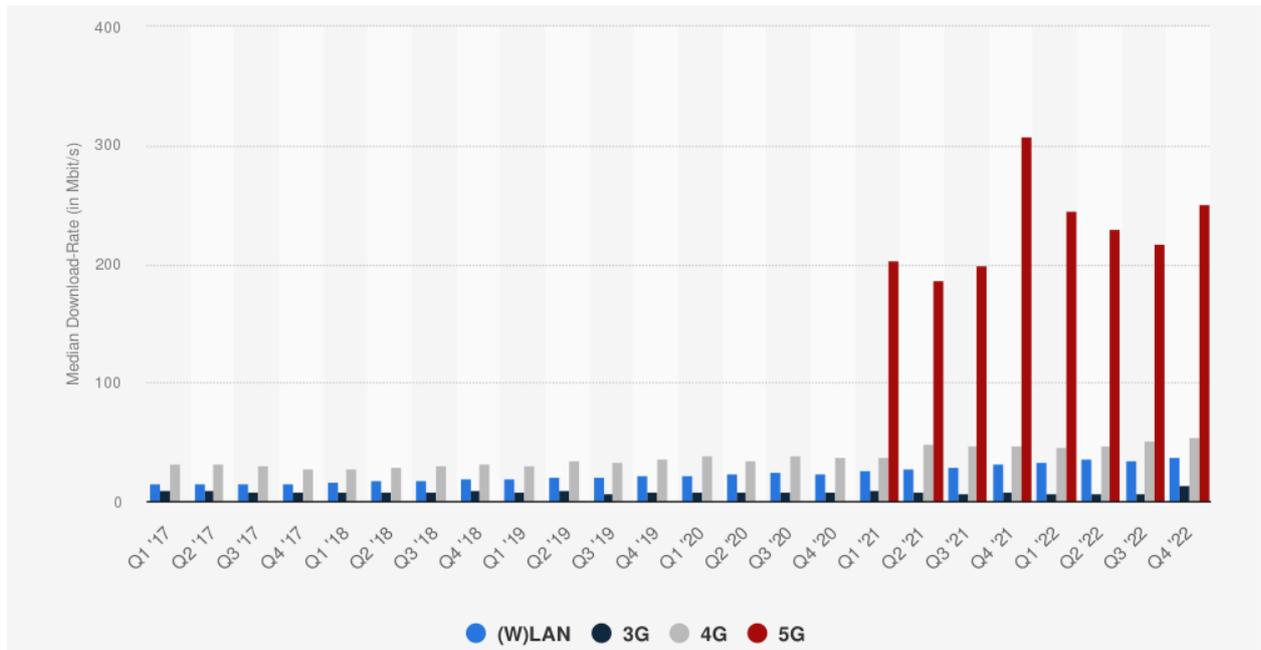


Abb. 29: Durchschnittliche Download-Geschwindigkeiten der Mobilfunkstandards in Österreich vom 1. Quartal 2017 bis zum 4. Quartal 2022,
 Quelle: <https://de-statista-com.elibrary.campus02.at/statistik/daten/studie/1301770/umfrage/download-rate-nach-mobilfunkstandard-in-oesterreich/> [Stand: 30.01.2023].

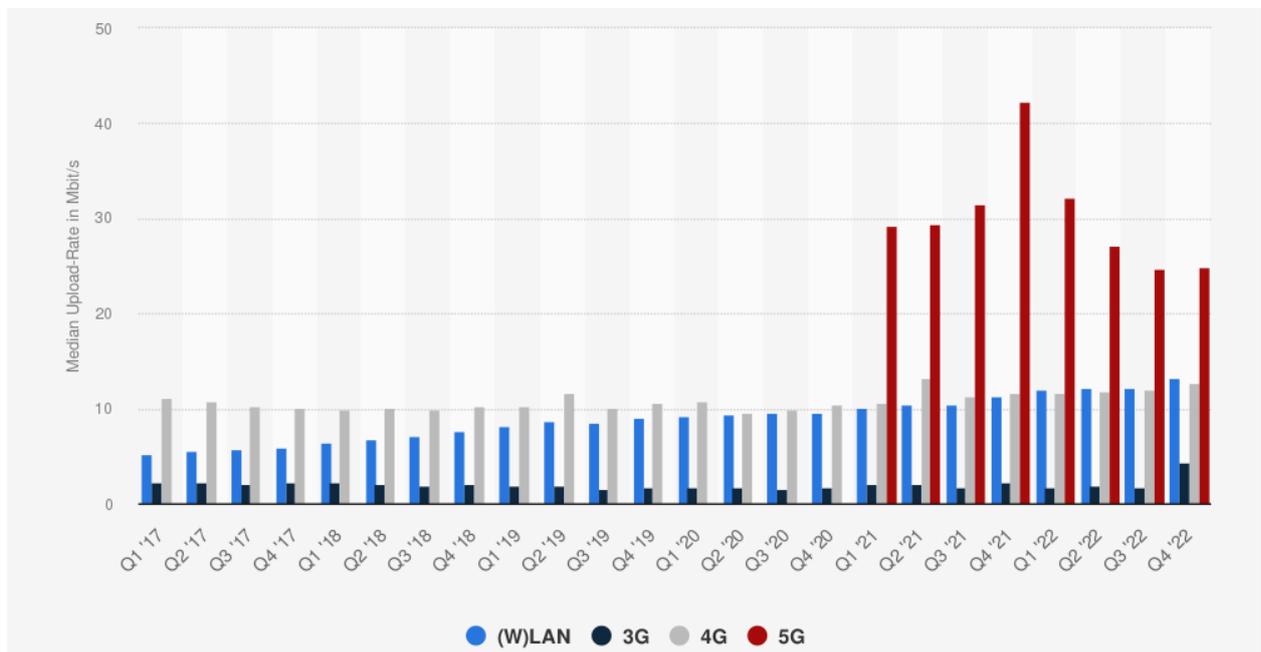


Abb. 30: Durchschnittliche Upload-Geschwindigkeiten der Mobilfunkstandards in Österreich vom 1. Quartal 2017 bis zum 4. Quartal 2022,
 Quelle: <https://de-statista-com.elibrary.campus02.at/statistik/daten/studie/1301762/umfrage/upload-rate-nach-mobilfunkstandard-in-oesterreich/> [Stand: 30.01.2023].

6.2 Verfahren zur Latenzzeitmessung

Bei der Latenzzeitmessung wird zwischen *Round-Trip Time* (RTT, Paketumlaufzeit) und *One-Way Latency* (OWL) unterschieden. Gängige Methoden, die für die Latenzmessung von 5G-Netzwerken in Frage kommen, sind Ping, OWAMP (*One-Way Active Measurement Protocol*), TWAMP (*Two-Way Active Measurement Protocol*) und Netzwerkmessgeräte, wie das *Network Master Pro MT1000A* von Anritsu.

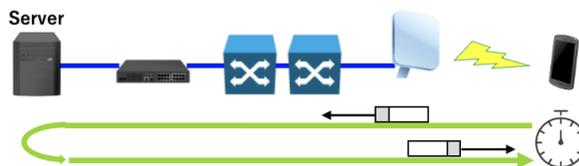


Abb. 31: *Round-Trip Time* (RTT, Paketumlaufzeit), Quelle: <https://dl.cdn-anritsu.com/en-en/test-measurement/files/Application-Notes/Application-Note/mt1000a-ef7101.pdf> [Stand: 16.09.2022].

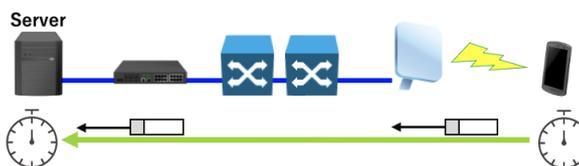


Abb. 32: *One-Way Latency* (OWL), Quelle: <https://dl.cdn-anritsu.com/en-en/test-measurement/files/Application-Notes/Application-Note/mt1000a-ef7101.pdf> [Stand: 16.09.2022].

Messung der Latenzzeit mit Ping:

Mit der PC-Anwendung Ping kann mit fast allen Geräten, die IP-Kommunikation unterstützen, die Umlaufzeit gemessen werden, ohne dass zusätzliche Software erforderlich ist. Durch Eingabe des Ping-Befehls in der Konsole wird eine Echo-Anfrage gesendet, die eine Echo-Antwort auslöst. Die Differenz zwischen Sende- und Empfangszeit wird als Latenzzeit (*Round-Trip Time*, RTT) in der Konsole ausgegeben, wie in Abb. 33 abgebildet. Der minimale Sendeintervall einer Echo-Anfrage beträgt an einer PC-Plattform 100 ms. Der Ping-Test dient zur Überprüfung der Verfügbarkeit von IP-Geräten, das bedeutet es wird fast keine Last am System erzeugt.³⁴

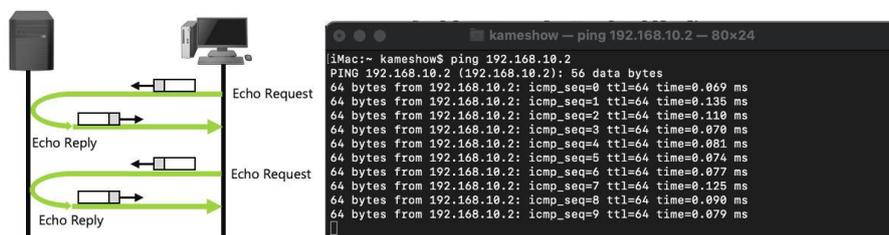


Abb. 33: Ping-Schema und Konsolenfenster mit Ping-Befehl und Ausgabe, Quelle: <https://dl.cdn-anritsu.com/en-en/test-measurement/files/Application-Notes/Application-Note/mt1000a-ef7101.pdf> [Stand: 16.09.2022].

³⁴ Vgl. Anritsu Corporation (2021), Online-Quelle [16.09.2022], Sochos (2020), Online-Quelle [17.09.2022].

Messung der Latenzzeit mit OWAMP/TWAMP:

Bei OWAMP und TWAMP handelt es sich um Protokolle, die von der *Internet Engineering Task Force* (IETF) spezifiziert wurden. OWAMP zielt darauf ab, Einwegcharakteristika wie *One-Way Latency* und *One-Way Loss* zu messen und ist in der IETF RFC 4656 definiert. TWAMP setzt auf der IETF RFC 4656 auf, ergänzt diese um Round-Trip-Messungen und ist in der IETF RFC 5357 spezifiziert. Das Unternehmen Rohde & Schwarz hat die Round-Trip-Latenzmessmethode übereinstimmend mit der IETF RFC 5357 (TWAMP) in seinen Mobile-Network-Testing-Produkten (MNT) implementiert und vertreibt diese auch zur 5G-Netzwerkmesung. Zwei Artikel mit unterschiedlichen Testfällen wurden hierzu von Rohde & Schwarz auf der Unternehmenswebseite bzw. auf *5gtechnologyworld* veröffentlicht. Der wesentliche Vorteil eines solchen Protokolls liegt darin, dass Lastszenarien durch Parameter wie die Paketrage, Paketanzahl und Testdurchlaufzeit definiert werden können.³⁵

Traffic pattern	Packets per second	Packet size	Target bandwidth
Constant Low	125	100 bytes	0.1 mb/sec (0.47%)
Constant Medium	200	650 bytes	1 Mb/sec
Constant High	1300	1450 bytes	15 Mb/sec

Abb. 34: Liste unterschiedlicher Lastparameter, Quelle: <https://www.5gtechnologyworld.com/measurements-show-5g-improves-latency-in-public-networks/> [Stand: 17.09.2022].

³⁵ Vgl. Sochos (2020), Online-Quelle [17.09.2022], Berger (2022), Online-Quelle [17.09.2022], Kottkamp/Rohde & Schwarz (2022), Online-Quelle [17.09.2022], Zekauskas/Karp/Shalunov/Boot/Teitelbaum (2015), Online-Quelle [17.09.2022], Babiarez/Krzanowski/Hedayat/Yum/Morton (2020), Online-Quelle [17.09.2022]

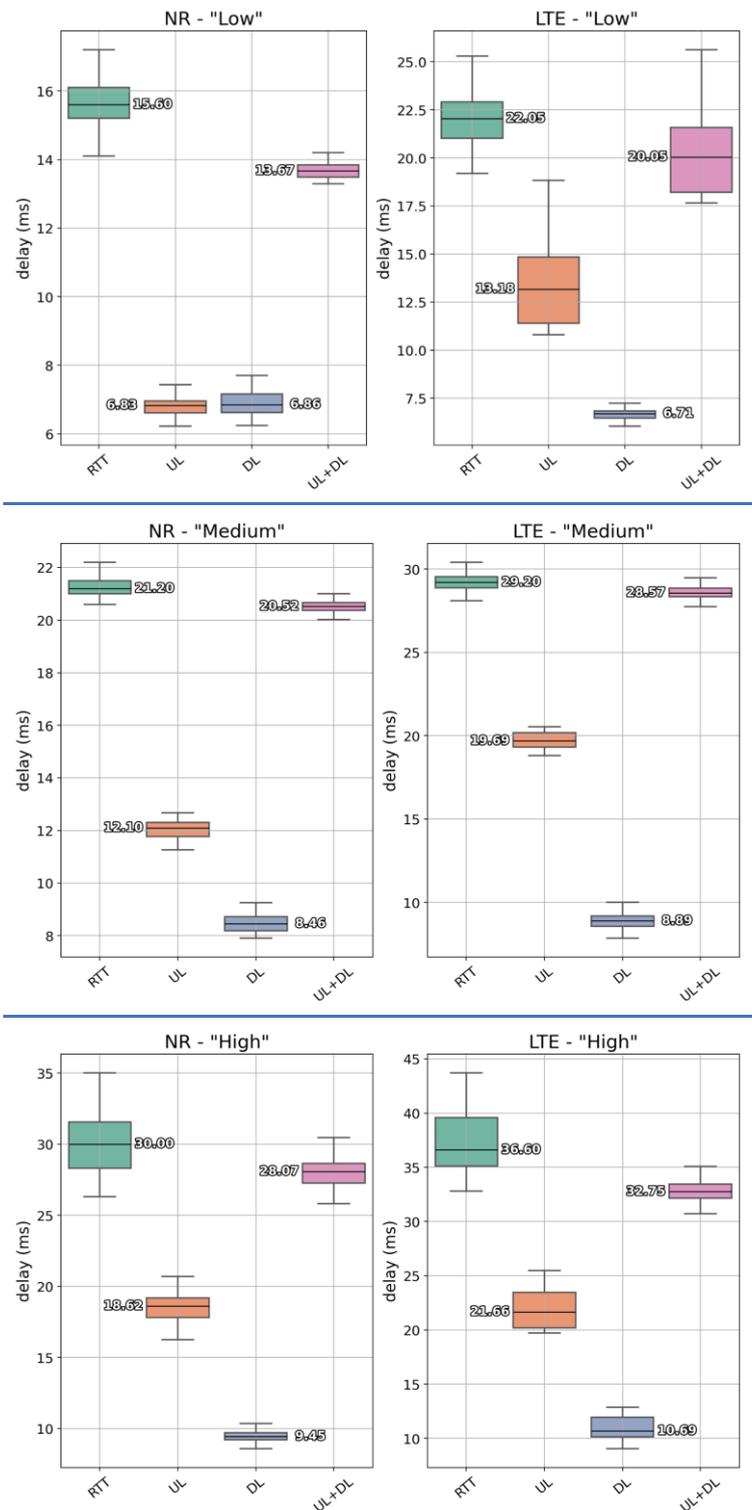


Abb. 35: Latenz-Box-Plots pro Technologie für den Vergleich von RTT mit OWL-Ergebnissen für ‚Low‘-, ‚Medium‘- und ‚High‘-Lastparameter, Quelle: <https://www.5gtechnologyworld.com/measurements-show-5g-improves-latency-in-public-networks/>, [Stand: 17.09.2022].

Da es sich bei TWAMP um einen offenen Standard handelt, kann er auch als Open-Source-Anwendung bezogen werden. Eine mögliche Variante ist die Open-Source-Software perfSONAR (**performance Service-Oriented Network monitoring ARchitecture**).

Eine nähere Betrachtung kostspieliger Netzwerkmessgeräte für 5G wird nicht weiterverfolgt, da diese für die Durchführung der vorliegenden Arbeit keine Relevanz haben.

6.3 Performance-Testing für industrielle Automatisierung

Beim Performance-Testing für industrielle Automatisierung werden zwei Arten von Parametern zur Testbeschreibung herangezogen. Erstere sind die Leistungsparameter, die gemessen bzw. berechnet werden können. Zweitere sind die Basisparameter, die sich auf die Testbedingungen beziehen und die Leistungsparameter beeinflussen.³⁶

Performance Parameter	Leistungsparameter
Transmission time	Übertragungszeit
Message reception interval	Nachrichtenempfangsintervall
Response time	Antwortzeit
Data throughput	Datendurchsatz
Message loss ratio	Nachrichtenverlustquote
Network availability	Netzverfügbarkeit
Time between failures	Zeit zwischen Ausfällen

Baseline Parameter	Basisparameter
Spatial extent of the industrial facility	Räumliche Ausdehnung der Industrieanlage
Spatial extent of the application	Räumliche Ausdehnung der Anwendung
Type of propagation environment	Art der Übertragungsumgebung
Ambient conditions	Umgebungsbedingungen
Number of wireless devices	Anzahl der drahtlosen Geräte
Number of logical links	Anzahl der logischen Verbindungen
Type of communication service	Art des Kommunikationsdienstes
Positions of wireless devices and distances between them	Positionen der drahtlosen Geräte und Entfernungen zwischen ihnen
Device mobility	Mobilität der Geräte
Active device factor	aktiven Geräte Faktor
User data length	Länge der Nutzdaten
Transfer interval	Übertragungsintervall
Data traffic volume per unit of area	Datenverkehrsvolumen pro Flächeneinheit
Survival time	Überlebenszeit
Observation time	Beobachtungszeit

Abb. 36: Übersicht über Leistungs- und Basisparameter, Quelle: 5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation (2021), S. 4-6.

Die Testprozedur umfasst mindestens eine, in der Regel jedoch mehrere Testgruppen. In jeder davon werden die für den Anwendungsfall relevanten Leistungsparameter überwacht, während eine Variation eines oder multipler Basisparameter stattfindet. Dieses Vorgehen ermöglicht es, den Einfluss der jeweiligen Basisparameter auf das System zu ermitteln. Zu beachten ist hierbei, dass die Variation eines Basisparameters die Ergebnisse mehrerer Leistungsparameter beeinflussen kann. Ein mögliches Testszenario wird in Abb. 37 veranschaulicht.³⁷

³⁶ Vgl. 5G Alliance for Connected Industries and Automation (Hrsg.) (2021), S. 4.

³⁷ Vgl. 5G Alliance for Connected Industries and Automation (Hrsg.) (2021), S. 6-7.

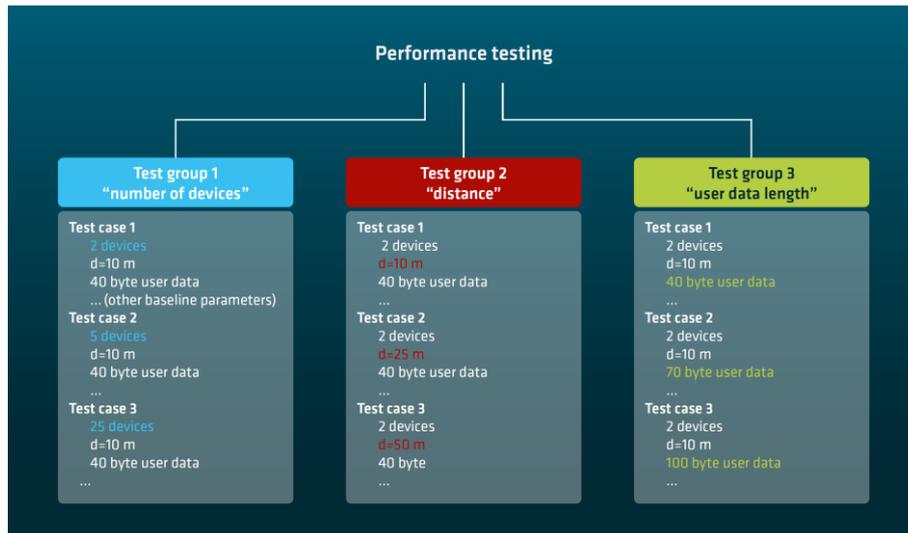


Abb. 37: Testgruppen mit Basisparametervariation,
 Quelle: 5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation (2021), S. 6.

Bei einem wie in Abb. 38 gezeigten Konzept werden das zu prüfende System und die dazugehörigen Basisparameter vom Testsystem kontrolliert und variiert. Letzteres muss in der Lage sein, die Basisparameter exakt zu reproduzieren sowie die Leistungsparameter in ausreichender Genauigkeit zu messen bzw. zu berechnen, um belastbare Ergebnisse zu erzielen.³⁸

Performance-Tests können in drei verschiedenen Umgebungen durchgeführt werden:

1. vor Ort in der Zielumgebung: Fabrikhalle, vor oder bei laufendem Betrieb, während oder nach der Funkplanungsphase, während Fehlersuche,
2. in der Referenzumgebung: physikalisch emuliert z. B. fabrikähnliches Gebäude,
3. in der Laborumgebung: virtuell emuliert.³⁹

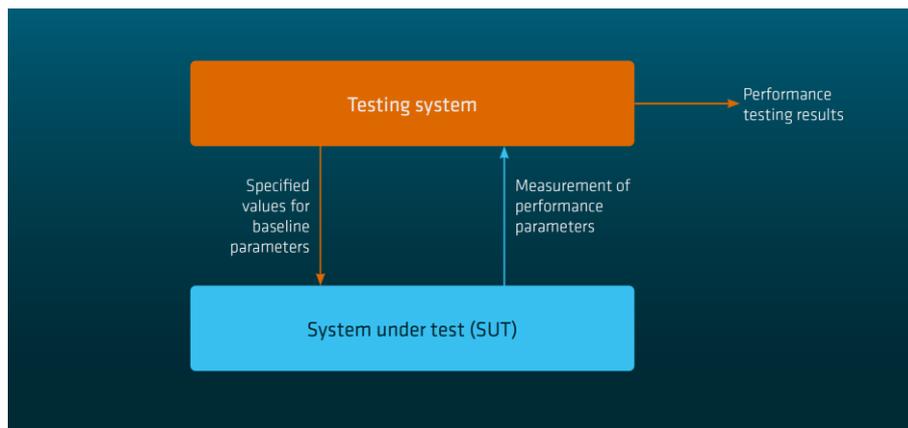


Abb. 38: Performance-Testing Konzept mit Interaktion zwischen dem Testsystem und zu prüfendem System,
 Quelle: 5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation (2021), S. 7.

³⁸ Vgl. 5G Alliance for Connected Industries and Automation (Hrsg.) (2021), S. 7.

³⁹ Vgl. 5G Alliance for Connected Industries and Automation (Hrsg.) (2021), S. 8.

Die Schemadarstellung einer industriellen Anwendung mit Mobilfunkkommunikation in Abb. 39 zeigt den Ablauf einer Automatisierungsaufgabe. Das Auslösen eines Eingangssignals mittels eines Schalters wird an eine speicherprogrammierbare Steuerung (SPS) übertragen. Diese verarbeitet die Information und erzeugt ein Signal, das an das Ausgangsmodul zur Steuerung des Motors übermittelt wird. Letzterer nimmt daraufhin eine Drehzahl- oder Drehrichtungsänderung vor bzw. stoppt.⁴⁰

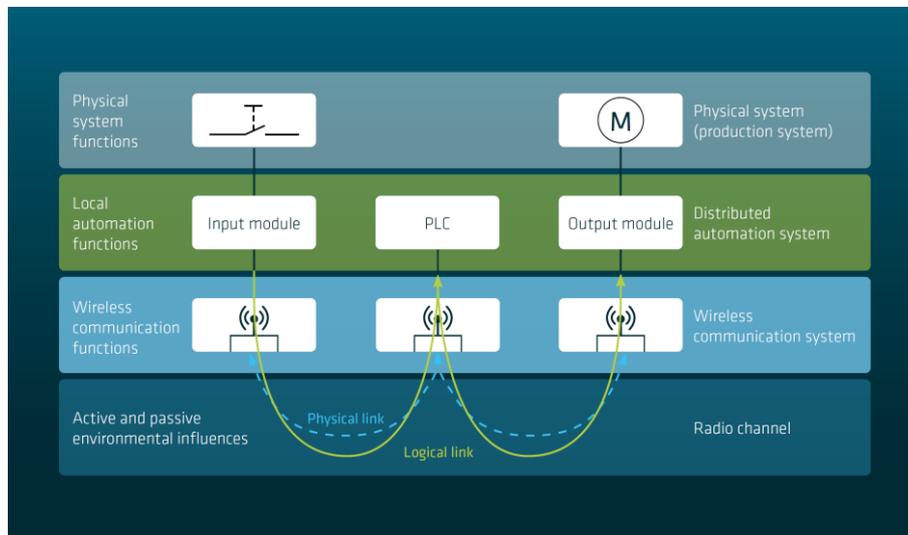


Abb. 39: Beispielschema einer 5G Industrieanlage,
Quelle: 5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation (2021), S. 8.

⁴⁰ Vgl. 5G Alliance for Connected Industries and Automation (Hrsg.) (2021), S. 8.

7 DEFINITION DES VERSUCHSAUFBAUS

Für die Durchführung von Datenübertragungen und Latenzzeitmessungen in einem 5G-Netzwerk ist es notwendig, einen Versuchsaufbau zu erstellen, der diese Anforderungen erfüllt. In Abb. 40 ist der Entwurf schematisch visualisiert. Folgende Aspekte wurden für die Komponentenauswahl definiert:

1. Der Versuchsaufbau muss über Batteriepacks betrieben werden können, damit in Zukunft eine Fernbedienbarkeit oder Montage auf einem Roboterchassis realisiert werden kann.
2. Die Videoaufnahme ist mittels eines stereoskopischen Kamerasetups umzusetzen.
3. Das Kamerastativ muss drehbar und motorisiert sein, damit der Aufnahmebereich gesteuert werden kann.
4. Eine VR-Brille dient als Wiedergabegerät.

Basierend auf den Anforderungskriterien wurde die Schemazeichnung angefertigt. Im Anschluss erfolgten die Recherche und die Auswahl der Komponenten. Dieser Prozess wird in Kapitel 8 beschrieben.

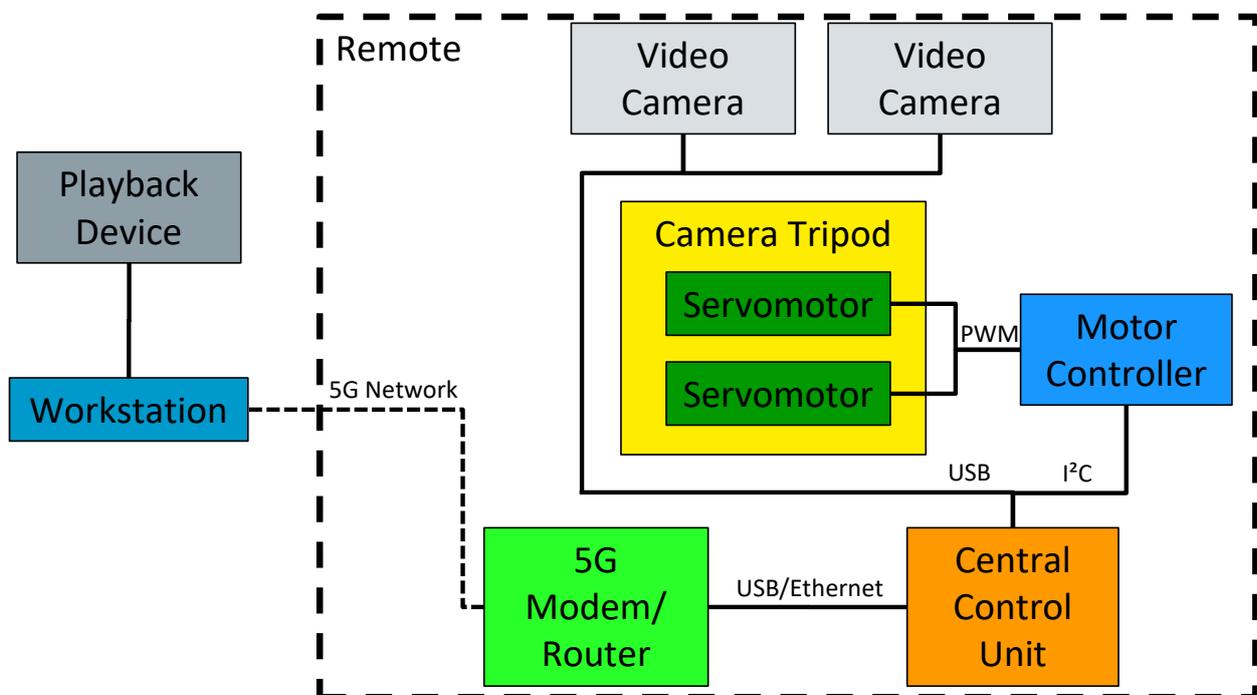


Abb. 40: Schema des Versuchsaufbaus, Quelle: Eigene Darstellung.

8 KOMPONENTENAUSWAHL

In diesem Kapitel wird auf die ausgewählten Komponenten eingegangen und deren Auswahl begründet. Einleitend soll festgehalten werden, dass zur Realisierung des Versuchsaufbaus die Grundsatzentscheidung getroffen wurde, den Einplatinenrechner Raspberry Pi Model B als zentrales Steuerungselement zu definieren. Davon ausgehend wurden mögliche Hardwarekomponenten recherchiert und einander gegenübergestellt. Abschließend wurde eine Auswahl für die Beschaffung getroffen.

8.1 Raspberry Pi Model B 5G HAT

Zum Zeitpunkt der Erstellung dieser Arbeit im Jahr 2022 ist die Anzahl der am Markt verfügbaren 5G-Hardware-Kits für den Einplatinenrechner Raspberry Pi Model B auf wenige Lösungen beschränkt. Jeweils ein Produkt der Hersteller Sixfab und Waveshare Electronics wurden für eine Endauswahl näher betrachtet. Beide Varianten verfügen im Wesentlichen über folgende Komponenten:

- 5G HAT (Hardware Attached on Top),
- 5G-Modem,
- Antennen,
- Gehäuse.

Beide Kits sind mit Übertragungsgeschwindigkeiten von bis zu 5,0 GBit/s (Sixfab) bzw. 2,4 GBit/s (Waveshare Electronics) für Downloads und bis zu 1,0 GBit/s (Sixfab) bzw. 0,5 GBit/s (Waveshare Electronics) für Uploads ausgewiesen. Das Sixfab-Kit beinhaltet ein Quectel M502Q-AE-Modem und das Waveshare-Electronics-Kit ein SIMcom SIM8200EA-M2-Modem, die über folgende Spezifikationen verfügen:⁴¹

		Quectel M502Q-AE	SIMcom SIM8200EA-M2
Download	5G SA Sub-6	4,2 GBit/s	4,0 GBit/s
	5G NSA Sub-6	5,0 GBit/s	
Upload	5G SA Sub-6	0,45 GBit/s	0,5 GBit/s
	5G NSA Sub-6	0,6 GBit/s	

5G Standalone Network (5G SA)
5G Non-Standalone Network (5G NSA)

Tab. 1: Datenübertragungsgeschwindigkeit 5G-Modem Quectel M502Q-AE und SIMcom SIM8200EA-M2, Quelle: Eigene Darstellung.⁴²

Da beide Produkte die Anforderungen erfüllen, wurde letztlich aufgrund der Gehäuseausführung die Entscheidung getroffen, das Hardware-Kit von Sixfab zu beschaffen. Die geschlossene Ausführung des

⁴¹ Vgl. Sixfab, Inc. (2022), Online-Quelle [15.06.2022]., Vgl. Waveshare Electronics (2022), Online-Quelle [01.06.2022].

⁴² Vgl. Quectel Wireless Solutions Co., Ltd. (Hrsg.) (2021), S. 2., Vgl. Zhang (2020), S. 11.

Indoor-Gehäuses und die Verfügbarkeit eines Outdoor-Gehäuses mit Schutzklasse IP54 bieten einen höheren Komponentenschutz bzw. zusätzliche Anwendungsmöglichkeiten.



Abb. 41: Sixfab Raspberry Pi 5G Development Kit,

Quelle: <https://sixfab.com/product/raspberry-pi-5g-development-kit-5g-hat/> [Stand: 19.05.2022].



Abb. 42: Waveshare SIM8200EA-M2 5G HAT for Raspberry Pi,

Quelle: <https://www.waveshare.com/product/iot-communication/long-range-wireless/4g-gsm-gprs/sim8200ea-m2-5g-hat.htm> [Stand: 24.05.2022].

8.2 Schwenk- / Neigesystem

Für die Aufnahme der Kameras wurden zwei Systeme evaluiert. Beim Ersten handelt es sich um das SPT200-Direct-Drive-Schwenk-/Neigesystem von ServoCity und beim Zweiten um das Lynx-B-Schwenk-/Neigesystem von Lynxmotion. Beide Systeme sind für den Betrieb mit Servomotoren des Herstellers Hitec ausgelegt. Das SPT200-Direct-Drive-Schwenk-/Neigesystem ist strukturell für eine Traglast von bis zu 2 lb bzw. 0,9 kg ausgelegt und laut Herstellerangaben mit folgenden Servomotoren kompatibel: HS-485HB, HS-645MG, HS-5685MH, HS-7950TH und HSB-9380TH.⁴³ Das Lynx-B-Schwenk-/Neigesystem wird standardmäßig mit HS-422-Servomotoren ausgeliefert. Es wird zwar eine Kompatibilität der Halterungen mit Standard-Servomotoren von Hitec ausgewiesen, diese wird jedoch nicht weiter spezifiziert.⁴⁴ Aufgrund der detaillierteren zugänglichen Informationen und damit des besseren Plug-and-Play-Ansatzes wurde das SPT200-Direct-Drive-Schwenk-/Neigesystem von ServoCity in Kombination mit Hitec HS-645MG gewählt. Die mit Kugellagern ausgeführten Servomotoren stellen eine robuste Lösung dar und sind für das Aufnehmen von Lasten von bis zu 1 lb bzw. 0,45 kg auf dem Stativ geeignet.⁴⁵

⁴³ Vgl. ServoCity (2022), Online-Quelle [15.06.2022].

⁴⁴ Vgl. Lynxmotion (2022), Online-Quelle [15.06.2022].

⁴⁵ Vgl. ServoCity (2022), Online-Quelle [15.06.2022].



Abb. 43: ServoCity-SPT200-Direct-Drive-Schwenk-/Neigesystem,
Quelle: <https://www.servocity.com/spt200-pan-tilt-kit/> [Stand: 15.06.2022].



Abb. 44: Lynxmotion-Lynx-B-Schwenk-/Neigesystem,
Quelle: <http://www.lynxmotion.com/p-287-lynx-b-pan-and-tilt-kit-black-anodized.aspx> [Stand: 15.06.2022].

8.3 Kamera

Bei der Auswahl der Kamera zur Videoaufzeichnung wurde ein Augenmerk auf Auflösung, Bildfrequenz, USB-Schnittstelle und kompakte Abmessungen gelegt. Eine hohe Auflösung sollte einerseits Nutzen aus der bestehenden Bandbreite ziehen und andererseits die Grundlast der zu übertragenden Daten bilden. Um eine ansprechende Wiedergabe auf der VR-Brille zu gewährleisten, wurde eine Mindestbildfrequenz der Kamera von 30 fps (*Frames per Second*) definiert. Durch die Datenübertragung über eine USB-Schnittstelle ergeben sich zwei wesentliche Vorteile. Erstens, der Raspberry Pi Model B verfügt über vier USB-Schnittstellen, die von außen leicht zugänglich sind. Zweitens, die Kameras lassen sich leicht an unterschiedlichen Systemen installieren und testen. Die kompakten Abmessungen bieten einerseits Vorteile für portable Einsatzszenarien und sind andererseits Voraussetzung für einen stereoskopischen Kameraaufbau, bei dem die beiden Kameras das Augenpaar repräsentieren. Im Zuge der Recherche wurden Kamerasysteme von den Herstellern ARDUCAM TECHNOLOGY CO., LIMITED und Shenzhen Ailipu Technology Co.,Ltd unter dem Markennamen ELP einander gegenübergestellt. Arducam bietet Lösungen für unterschiedliche Plattformen (Arduino, Raspberry Pi, Jetson Nano usw.) und mit unterschiedlichen Schnittstellen an. Des Weiteren wird auch ‚*Customized Turnkey Design and Manufacturing*‘ angeboten, neben Kameraevaluierung-Kits für Test- und Entwicklungszwecke.⁴⁶ Bei ELP liegt der Fokus auf USB-Kameralösungen für industrielle und kommerzielle Bildverarbeitungsanwendungen, wobei ‚*Customization Service*‘ ebenfalls im Angebotsportfolio enthalten ist.⁴⁷ Aufgrund der einfacheren Handhabung wurde die Einschränkung getroffen, auf Kameras mit

⁴⁶ Vgl. ARDUCAM TECHNOLOGY CO., LIMITED (2022), Online-Quelle [17.06.2022].

⁴⁷ Vgl. Shenzhen Ailipu Technology Co.,Ltd (2022), Online-Quelle [17.06.2022].

geschlossenem Gehäuse zurückzugreifen. Für die nähere Betrachtung wurden zwei Kameramodelle herangezogen, die ELP-USB4K02AF-KL100W und die Arducam B0280. Beide verfügen über einen Sony-Bildsensor und eine USB-2.0-Schnittstelle. Die maximale Auflösung des Sony IMX477, der Arducam B0280, liegt mit 4032(H) x 3040(V) etwas über dem in der ELP-USB4K02AF-KL100W verbauten Sony IMX415 mit 3840(H) x 2160(V). Jedoch kann der Vorteil der höheren Auflösung in dieser Anwendung nicht geltend gemacht werden, da die Arducam B0280 eine Bildfrequenz von 30 fps nur bis zu einer Auflösung von 2560(H) x 1440(V) (2K Quad HD) bietet. Hingegen kann die ELP USB4K02AF KL100W, bei kleineren Außenabmessungen, Aufnahmen mit 30 fps bis zu 3840(H) x 2160(V) (4K UHD) leisten.⁴⁸ Die höhere Bildfrequenz bei hoher Auflösung war letztlich das ausschlaggebende Kriterium für die Kamerawahl. Somit wurde die ELP-USB4K02AF-KL100W für den Versuchsaufbau beschafft.



Abb. 45: ELP-USB4K02AF-KL100W SONY-IMX415-4K-UHD-USB-Kamera mit *Auto Focus*,
Quelle: <http://www.elpcctv.com/elp-128-sony-imx415-sensor-night-vision-4k-usb-camera-with-white-led-p-169.html>
[Stand: 15.06.2022].



Abb. 46: Arducam-B0280-12MP-IMX477-USB-Kameramodul mit 6 mm-CS-Mount-Objektiv,
Quelle: <https://www.uctronics.com/download/Amazon/B0280.pdf> [Stand: 15.06.2022].

⁴⁸ Vgl. ARDUCAM TECHNOLOGY CO., LIMITED (2022), Online-Quelle [15.06.2022]., Vgl. Shenzhen Ailipu Technology Co.,Ltd (2022), Online-Quelle [16.06.2022].

8.4 Motorsteuerung

Um Beschädigungen am zentralen Steuerungselement, dem Einplatinenrechner Raspberry Pi Model B, vorzubeugen, wurde entschieden, ein externes Servo-Controller-Board zu verwenden. Durch die separate Stromversorgung für die Servomotoren ist die Trennung von Steuerkreis und Leistungskreis gegeben. Basierend auf der separaten Stromversorgung, der Kompatibilität zum Raspberry Pi Model B und der Möglichkeit zur Steuerung von mindestens zwei Servomotoren kamen zwei Platinen in die nähere Auswahl – die erste, auf Basis des NXP-PCA9685-Chips, die in Form von 16-Channel-12-Bit-PWM-Servo-Driver-Boards am Markt weit verbreitet ist und von unterschiedlichen Herstellern vertrieben wird;⁴⁹ die zweite, das Maestro-USB-Servo-Controller-Board von Pololu, das je nach Ausführung 6-, 12-, 18- oder 24-kanalig ist.⁵⁰ Nicht nur in der Anzahl der Ausgänge unterscheiden sich die Boards, sondern auch in den vorhandenen Schnittstellen. Während das PCA9685-Servo-Controller-Board über einen I²C(*Inter-Integrated Circuit*)-seriellen Datenbus verfügt,⁵¹ weist das Maestro-USB-Servo-Controller-Board eine USB- und eine UART-Schnittstelle zur Steuerung auf.⁵² Aufgrund der wesentlich besseren Marktverfügbarkeit fiel die Entscheidung letztlich auf ein PCA9685-Servo-Controller-Board.

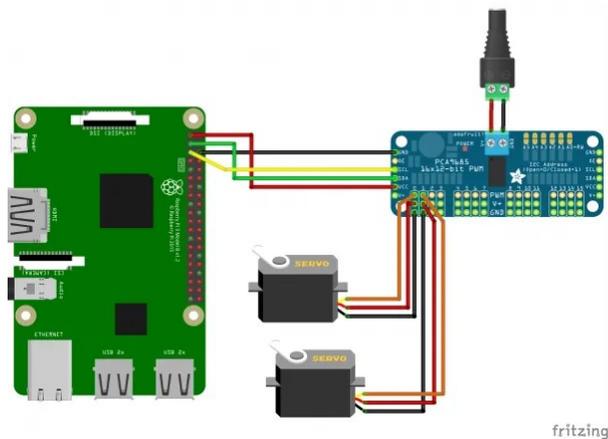


Abb. 47: Servomotoren mit PCA9685-Servo-Controller-Board und Raspberry Pi steuern,
Quelle: <https://tutorials-raspberrypi.de/mehrere-servo-motoren-steuern-raspberry-pi-pca9685/> [Stand: 18.06.2022].



Abb. 48: PCA9685 16 Channel 12 Bit PWM Servo Driver,
Quelle: http://wiki.sunfounder.cc/index.php?title=PCA9685_16_Channel_12_Bit_PWM_Servo_Driver [Stand: 18.06.2022].

⁴⁹ Vgl. Semiconductors, N.X.P. (2022), Online-Quelle [19.06.2022], S. 2.

⁵⁰ Vgl. Pololu Corporation (2022), Online-Quelle [20.06.2022], S. 4.

⁵¹ Vgl. Semiconductors, N.X.P. (2022), Online-Quelle [19.06.2022], S. 2.

⁵² Vgl. Pololu Corporation (2022), Online-Quelle [20.06.2022], S. 17.

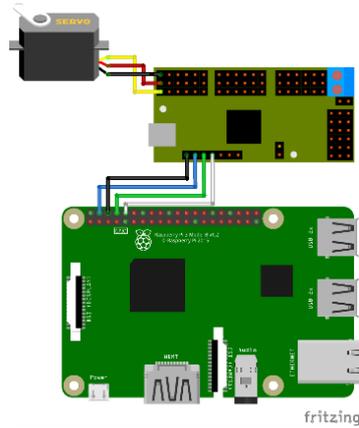


Abb. 49: Servomotoren mit Pololu Micro-Maestro-USB-Servo-Controller-Board und Raspberry Pi steuern,
Quelle: <https://www.aranacorp.com/en/command-a-serial-controller-maestro-with-raspberry-pi/> [Stand: 20.06.2022].

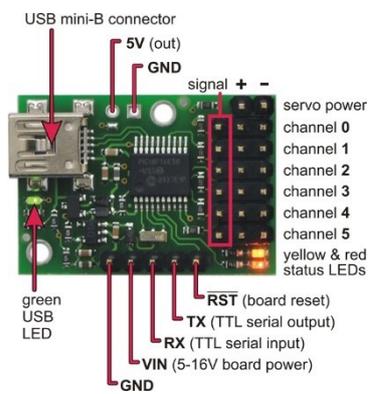


Abb. 50: Pololu Micro Maestro USB Servo Controller,
Quelle: <https://www.pololu.com/product/1350> [Stand: 20.06.2022].

8.5 VR-Headset

Die Fachhochschule besitzt zwei VR-Brillen-Systeme, zum einen die Oculus Quest 2 und zum anderen die Vive Pro. Die VR-Brillen verfügen über zwei Controller und über 6-DoF(Degrees of Freedom)-Tracking. Ein wesentlicher Unterschied der beiden Systeme besteht in der Art des Motion-Trackings. Die Vive Pro funktioniert nach dem Outside-in-Prinzip und ist auf zwei *Vive Base Stations* als externe Marker angewiesen. Beim Inside-out-Prinzip der Oculus Quest 2 hingegen kommen vier integrierte Kameras zum Einsatz. Dieses System ist somit auf keine zusätzliche Hardware für das Tracking angewiesen.⁵³ Da die Oculus Quest 2 zeitlich besser verfügbar ist und ortsungebunden eingesetzt werden kann, wurde dieses System für die Umsetzung des Versuchsaufbaus gewählt.

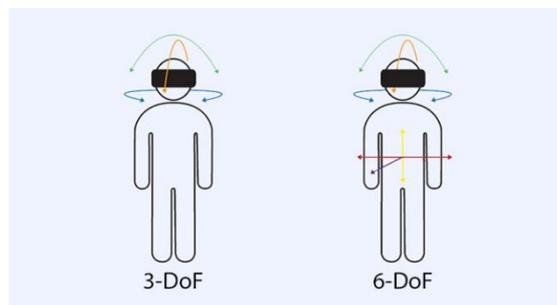


Abb. 51: 3-DoF (Rotationsbewegung) und 6-DoF (Rotations- und Translationsbewegung) mit einem VR-Headset, Quelle: <https://virtualspeech.com/blog/degrees-of-freedom-vr> [Stand: 13.07.2022].



Abb. 52: VR-Brille Oculus Quest 2, Quelle: <https://business.oculus.com/products/specs/> [Stand: 13.07.2022].



Abb. 53: VR-Brille Vive Pro, Quelle: <https://www.vive.com/de/product/vive-pro/> [Stand: 13.07.2022].

⁵³ Vgl. Meta (2022), Online-Quelle [13.07.2022], Vgl. HTC Corporation (2022), Online-Quelle [13.07.2022].

9 PRAKTISCHE UMSETZUNG

Die praktische Umsetzung umfasst das Aufsetzen des Raspberry Pi. Des Weiteren die Herstellung einer WebRTC-Videoübertragung in einem lokalen Netzwerk und über Internet. Gefolgt von der Integration des VR-Headsets zur Videowiedergabe und Steuerung der Servomotoren. Geschlossen wird das Kapitel mit aufsetzen eines Netzwerkmesssystems zur Latenzzeitmessung und anderer Netzwerkparameter.

9.1 Raspberry Pi

Der Einplatinenrechner Raspberry Pi 4 stellt das zentrale Steuerungselement für die Videoübertragung und die Servomotoren dar. Als Betriebssystem wurde das zu diesem Zeitpunkt aktuelle Release des Raspberry Pi OS verwendet, da es speziell für die Anwendung auf Raspberry Pis zugeschnitten ist. Grundsätzlich wären auch andere Linux-Distributionen denkbar, jedoch ist dabei zu beachten, dass dadurch in den darauffolgenden Kapiteln das beschriebene Vorgehen zur Umsetzung abweichen kann. Um Softwarekompatibilitätsproblemen vorzubeugen, wurde aufgrund der kurzen Marktverfügbarkeit der 64-bit-Version von Raspberry Pi OS auf mögliche Performancevorteile verzichtet. Damit fiel die endgültige Wahl auf Raspberry Pi OS mit Desktop und 32-bit. Da es sich bei dieser Arbeit um kein Vorgehen nach einem vorhandenen Schema handelt, bietet der implementierte Desktop eine Erleichterung im Umgang mit dem Einplatinenrechner Raspberry Pi. Um einen Remote-Zugang zu ermöglichen, empfiehlt es sich, ebenfalls SSH (*Secure Shell*) und VNC (*Virtual Network Computing*) zu aktivieren. Wird an mehreren Standorten und an unterschiedlichen Netzwerken mit einem Raspberry Pi operiert, so bietet sich an, umgehend eine Remote-Verbindung via Ethernet/LAN zu konfigurieren. Es können neue WLAN-Netzwerke direkt auf der microSD-Karte im Config-File `/boot/wpa_supplicant.conf` konfiguriert werden. Bei diesem Vorgehen sind einige Aspekte zu beachten. Erstens ist zur Ermittlung der IP-Adresse nicht immer ein Zugang zum Netzwerkrouter vorhanden. Zweitens ist der Nutzen von Tools wie *Advanced IP Scanner* bei großen Subnetzmasken oder bei vielen gleichartigen Geräten begrenzt. Eine äußerst komfortable Lösung, um dieses Problem zu umgehen, ist die Nutzung eines dynamischen DNS-Services (*Domain Name System*) mit automatisiertem IP-Update. Weitere Ausführungen hierzu befinden sich in Kapitel 0. Abschließend ist zu erwähnen, dass die Anwendung PiShrink zur Minimierung der Imagegröße der microSD-Karte einerseits den Speicherbedarf und die Schreibzeit reduziert und andererseits die Flexibilität erhöht, da Speicherkarten mit unterschiedlichsten Spezifikationen zum Bespielen verwendet werden können.

9.2 WebRTC Videoübertragung

Die Durchführung eines ersten Funktionstests auf Basis von Motion zeigt, dass die gängigen Applikationen wie Motion und ffmpeg die Anforderungen für die Anwendung in dieser Arbeit nicht erfüllen.⁵⁴ Die in Kapitel 4 beschriebene tieferegehende Recherche hat den Wechsel zu einem Programm mit Unterstützung des WebRTC-Protokolls zur Folge. Für die ausgewählte Plattform, den Einplatinenrechner Raspberry Pi 4 in Kombination mit einem Linux-Betriebssystem, gibt es im Wesentlichen drei frei verfügbare

⁵⁴ Vgl. motion-project.github.io (2022), Online-Quelle [16.11.2022], Vgl. FFmpeg (2022), Online-Quelle [16.11.2022].

Softwarelösungen: webrtc-streamer, GStreamer und UV4L (User space Video4Linux). Aus dieser ausgewählten Gruppe stellen GStreamer und UV4L die professionellen Lösungen dar, mit online abrufbarer Dokumentation, Tutorials und Demos. Aufgrund der guten Dokumentation und der großen Community in diesem Bereich wird UV4L für diese Arbeit ausgewählt.

9.2.1 UV4L Basic Setup

Die Installation für die ARM-Architektur erfolgt weitestgehend nach den auf der Webseite <https://www.linux-projects.org/uv4l/installation/> dokumentierten Instruktionen. Zur Reduzierung von Fehlerquellen wird eine nahezu vollumfängliche Installation mit zahlreichen Zusatzpaketen durchgeführt. Bevor die eigentliche Installation durchgeführt werden kann, ist es notwendig, die externe Paketquelle von UV4L der Paketliste hinzuzufügen. In diesem Fall handelt es sich um einen *Uniform Resource Locator* (URL). Der vollständige Befehl wird in Quellcode 1 gezeigt.⁵⁵

```
Pi5G@RaspberryPiTeleoperation5G:~ $ curl https://www.linux-projects.org/listing/uv4l_repo/lpkey.asc | sudo apt-key add -
Pi5G@RaspberryPiTeleoperation5G:~ $ echo "deb https://www.linux-projects.org/listing/uv4l_repo/raspbian/stretch stretch main" | sudo tee /etc/apt/sources.list.d/uv4l.list
```

Quellcode 1: Hinzufügen der externen Paketquelle von UV4L im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Vor jeder Installation sollte eine Aktualisierung der Paketlisten mittels eines Update-Befehls durchgeführt werden. Dabei werden die neuesten Versionen der bereits installierten Programme in den Dateilisten hinterlegt. Der darauffolgende Upgrade-Befehl vergleicht die vorhandenen Paketversionen mit den aktualisierten Listen und installiert diese bei Vorhandensein neuerer Programmversionen.⁵⁶

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get update
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get upgrade
```

Quellcode 2: Aktualisierung der Paketliste und Installation neuerer Paketversionen im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Anschließend kann, wie in Quellcode 3 ausgeführt, UV4L mit dem Raspicam-Treiber installiert werden. Dieses Treiberpaket ist auf die Verwendung von Raspberry-Pi-Kameramodulen mit Mobile-Industry-Processor-Interface(MIPI)-Schnittstelle zugeschnitten. Trotz des Einsatzes von Videokameras mit USB-Schnittstelle ist die Installation des Raspicam-Treibers notwendig, da alle weiteren Installationspakete darauf aufbauen. Der Vollständigkeit halber soll an dieser Stelle erwähnt werden, dass alternativ der Treiber `uv4l-raspicam-ai` verwendet werden könnte. Dieser verfügt über erweiterte Funktionalitäten hinsichtlich Objekterkennung, Objektverfolgung usw. Aufgrund des abweichenden Kamerasystems und der abweichenden Vorgehensweise spielen beide Treiber im weiteren Verlauf dieser Arbeit eine untergeordnete Rolle.⁵⁷

⁵⁵ Vgl. Follmann (2018), S. 27, Vgl. Dembowski (2019), S. 195.

⁵⁶ Vgl. Follmann (2018), S. 27-28, Vgl. Dembowski (2019), S. 196.

⁵⁷ Vgl. [linux-projects/uv4l/installation/](https://www.linux-projects.org/uv4l/installation/) (2022), Online-Quelle [19.10.2022].

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l uv4l-raspicam
```

Quellcode 3: Installation des UV4L-Raspicam-Treibers im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Das Paket `uv4l-raspicam-extras` fügt den Treiber zur Bootliste hinzu, damit dieser beim Systemstart geladen wird. Des Weiteren beinhaltet es ein Service-Skript für `start`, `stop` und `restart` des Treibers.⁵⁸

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-raspicam-extras
```

Quellcode 4: Installation von UV4L-Raspicam-Extras im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Damit ist die Installation des UV4L-Kernmoduls abgeschlossen. Gegebenenfalls ist für einen fehlerfreien Betrieb die Kamera zu aktivieren und ausreichend Speicher der *Graphics Processing Unit* (GPU) zuzuweisen (Empfehlung > 256 MB). Die Konfiguration erfolgt in der `raspi-config`, wie in Quellcode 5 und Abb. 54 abgebildet. Zum Wirksamwerden der neuen Einstellungen ist ein System-Reboot durchzuführen.⁵⁹

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo raspi-config
```

Quellcode 5: Aufrufen der `raspi-config` im Pi-OS-Terminal, Quelle: Eigene Darstellung.

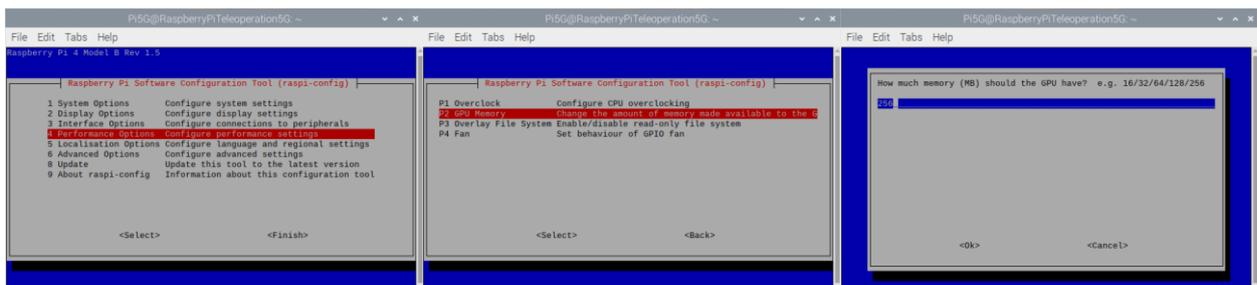


Abb. 54: Zuweisung von GPU-Speicher in der *Raspberry Pi Software Configuration*, Quelle: Eigene Darstellung.

Anschließend werden alle weiteren optionalen Pakete installiert. Für den in dieser Arbeit gegebenen Anwendungsfall sind die Pakete `uv4l-server` und `uv4l-uvc` essenziell. Die Webserverfunktionalitäten sind in `uv4l-server` eingebettet, das Einbinden von USB-Kameras erfolgt mittels des `uv4l-uvc`-Treibers (*USB Video Device Class*).⁶⁰

⁵⁸ Vgl. [linux-projects/uv4l/installation/](#) (2022), Online-Quelle [19.10.2022].

⁵⁹ Vgl. [linux-projects/uv4l/installation/](#) (2022), Online-Quelle [19.10.2022].

⁶⁰ Vgl. [linux-projects/uv4l/installation/](#) (2022), Online-Quelle [19.10.2022], Vgl. [linux-projects/documentation/uv4l-server/](#), Online-Quelle [19.10.2022], Vgl. [linux-projects/documentation/uv4l-uvc/](#), Online-Quelle [19.10.2022].

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-server uv4l-uvc uv4l-xscreen uv4l-mjpegstream uv4l-dummy uv4l-raspisp
```

Alternative:

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-server
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-uvc
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-xscreen
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-mjpegstream
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-dummy
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-raspisp
```

Quellcode 6: Installation optionaler UV4L-Pakete im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Für die Erweiterung des Streaming-Servers um WebRTC stehen zwei Pakete zur Verfügung – das eine für die Plattformen Raspberry Pi 1, Compute Module 1, Zero oder Zero W (Wireless) und das andere für alle anderen Raspberry-Pi-Plattformen (2, 3, 4). Zweites wird in der vorliegenden Arbeit verwendet.⁶¹

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-webrtc
```

Quellcode 7: Installation des WebRTC-Pakets für Raspberry Pi 2, 3 und 4 im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Der Streaming-Server ist in der Lage jede beliebige benutzerdefinierte Webanwendung zu bedienen. Das uv4l-demos Paket enthält einige Beispiele. Zu Testzwecken wurde neben der Standard-Webanwendung, auf die im weiteren Verlauf noch genauer eingegangen wird, die benutzerdefinierte Webanwendung mit Gesichtserkennung ausgeführt (<https://www.linux-projects.org/uv4l/tutorials/custom-webapp-with-face-detection/>).⁶²

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-demos
```

Quellcode 8: Installation des Demo-Pakets, einer externen Webanwendung, im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Der Webserver kann als *Hypertext Transfer Protocol* (http) oder *Hypertext Transfer Protocol Secure* (https) ausgeführt werden. Für das Betreiben eines Webserver mittels https ist es notwendig, einen passwortlosen privaten Schlüssel und ein gültiges Zertifikat bereitzustellen. Diese werden in die Server-Konfiguration eingebettet, anschließend ist der Streaming-Server mittels https zu erreichen. In Quellcode 9 und Abb. 55 sind eine Möglichkeit der Erstellung eines privaten Schlüssels und eines selbstsignierten Zertifikats abgebildet.⁶³

```
Pi5G@RaspberryPiTeleoperation5G:~ $ openssl genrsa -out selfsign.key 2048
&& openssl req -new -x509 -key selfsign.key -out selfsign.crt -sha256
```

Quellcode 9: Erstellung eines privaten Schlüssels und eines selbstsignierten Zertifikats im Pi-OS-Terminal, Quelle: Eigene Darstellung.

⁶¹ Vgl. [linux-projects/uv4l/installation/](https://www.linux-projects.org/uv4l/installation/) (2022), Online-Quelle [19.10.2022].

⁶² Vgl. [linux-projects/uv4l/installation/](https://www.linux-projects.org/uv4l/installation/) (2022), Online-Quelle [19.10.2022], Vgl. [linux-projects/uv4l/tutorials/custom-webapp-with-face-detection/](https://www.linux-projects.org/uv4l/tutorials/custom-webapp-with-face-detection/) (2022), Online-Quelle [19.10.2022].

⁶³ Vgl. [linux-projects/uv4l/installation/](https://www.linux-projects.org/uv4l/installation/) (2022), Online-Quelle [19.10.2022].

```
Pi5G@RaspberryPiTeleoperation5G:~ $ openssl genrsa -out selfsign.key 2048 && openssl req -new -x509 -key selfsign.key -out selfsign.crt -sha256
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:AT
State or Province Name (full name) [Some-State]:Austria
Locality Name (eg, city) []:Graz
Organization Name (eg, company) [Internet Widgits Pty Ltd]:CAMPUS 02
Organizational Unit Name (eg, section) []:5G
Common Name (e.g. server FQDN or YOUR name) []:FS
Email Address []:
```

Abb. 55: Optionale Angaben zur Erstellung des privaten Schlüssels und des selbstsignierten Zertifikats, Quelle: Eigene Darstellung.

Die WebRTC-Erweiterung ermöglicht unter anderem eine Live-Audio- und Videoübertragung zu allen Teilnehmern einer Jitsi-Meet-Webkonferenz. Durch das Paket `uv4l-xmpp-bridge` kann die Konfiguration bzw. Start/Stop ohne Browser oder grafische Benutzeroberfläche direkt im Pi-OS-Terminal durchgeführt werden.⁶⁴

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt-get install uv4l-xmpp-bridge
```

Quellcode 10: Installation des XMPP-Pakets im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Der `uv4l-raspicam`-Treiber erkennt Raspberry-Pi-Kameras mit MIPI-Schnittstellen automatisch. Diese Funktion ist beim `uv4l-uvv`-Treiber für UVC-Geräte nicht vorhanden. Daher ist es erforderlich, diese vorab bzw. bei der Konfiguration des Streams mittels Angabe der Device-ID oder des Device-Path zu registrieren.⁶⁵ Die Ermittlung der Device-ID bzw. des Device-Path kann durch den Befehl `lsusb` im Pi-OS-Shell durchgeführt werden. Dieser Befehl gibt alle angeschlossenen und erkannten USB-Geräte aus. Wie in Abb. 56 ersichtlich, sind der Gerätebezeichnung Device-Path und Device-ID vorangestellt.⁶⁶ Bei mehr als einem identen angeschlossenen Gerät kann eine eindeutige Zuordnung nur über den Device-Path erfolgen, da alle bauartgleichen Geräte die gleiche Device-ID teilen.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ lsusb
```

Quellcode 11: `lsusb` Befehl in Pi OS Terminal, Quelle: Eigene Darstellung.

⁶⁴ Vgl. [linux-projects/uv4l/installation/](#) (2022), Online-Quelle [19.10.2022].

⁶⁵ Vgl. [linux-projects/documentation/uv4l-core/](#), Online-Quelle [20.10.2022].

⁶⁶ Vgl. Follmann (2018), S. 36-37, Vgl. Dembowski (2019), S.175.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 32e4:0415 4K USB Camera 4K USB Camera
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

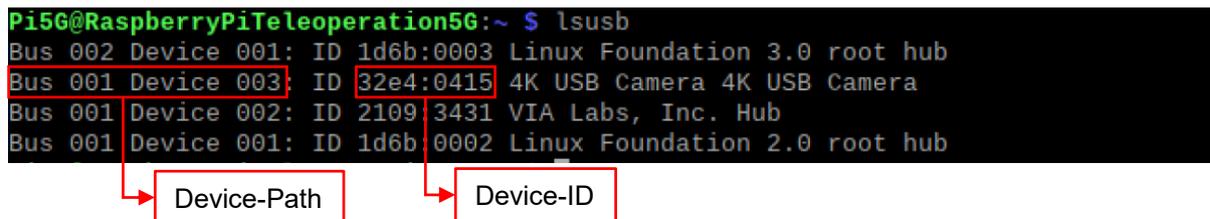


Abb. 56: Ausgabe des lsusb-Befehls im Pi-OS-Terminal, Quelle: Eigene Darstellung.

In Quellcode 12 ist die Registrierung eines USB-Geräts gezeigt, die dazu dient, dass das Gerät für den uv4l-uvc-Treiber in Verbindung mit der Standard-Webserverkonfiguration abrufbar ist. Mit UV4L lassen sich eine Vielzahl von Einstellungen für den Web-Stream vornehmen. Auf diese wird erst im weiteren Verlauf der Arbeit genauer eingegangen. In Abb. 57 ist eine erfolgreiche Registrierung mit dazugehöriger Rückgabe ersichtlich. Im Anschluss erfolgt ein Systemneustart und ein erster Funktionstest kann durchgeführt werden.⁶⁷

```
Pi5G@RaspberryPiTeleoperation5G:~ $ uv4l --syslog-host localhost --driver
uvc --device-path 001:003

Alternative:
Pi5G@RaspberryPiTeleoperation5G:~ $ uv4l --syslog-host localhost --driver
uvc --device-id 32e4:0415
```

Quellcode 12: Registrierung eines USB-Geräts im Pi-OS-Terminal, Quelle: Eigene Darstellung.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ uv4l --syslog-host localhost --driver uvc --
device-id 32e4:0415
<notice> [core] Trying to loading driver 'uvc' from built-in drivers...
<notice> [core] Loading driver 'uvc' from external plug-in's...
<notice> [driver] Video functionality '4K USB Camera' recognized at 32e4:0415
<notice> [core] Device detected!
<notice> [core] Registering device node /dev/uv4l
```

Abb. 57: Registrierung eines USB-Geräts und dazugehörige Rückgabe im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Der uv4l-uvc-Treiber wird bereits mit Systemboot ausgeführt, im Gegensatz zum uv4l-raspicam-Treiber, der erst mit dem Befehl `sudo service uv4l_raspicam restart` gestartet werden muss. Im Anschluss kann der Webserver von einem Gerät im selben Netzwerk in einem Browser aufgerufen werden. Die Standard-URLs sind für uv4l-raspicam `http(s)://Hostname oder IP-Adresse:8080/` und für uv4l-uvc `http(s)://Hostname oder IP-Adresse:8090/`.⁶⁸ In Abb. 58 ist die UV4L-Streaming-Server-Webanwendung mit all ihren Implementierungen abgebildet, in Abb. 59 ist mit inaktivem und aktivem Stream in der WebRTC-Two-Way-Audio/Video-Umgebung eine davon gezeigt.

⁶⁷ Vgl. linux-projects/documentation/uv4l-core/, Online-Quelle [20.10.2022].

⁶⁸ Vgl. linux-projects/uv4l/installation/ (2022), Online-Quelle [19.10.2022].

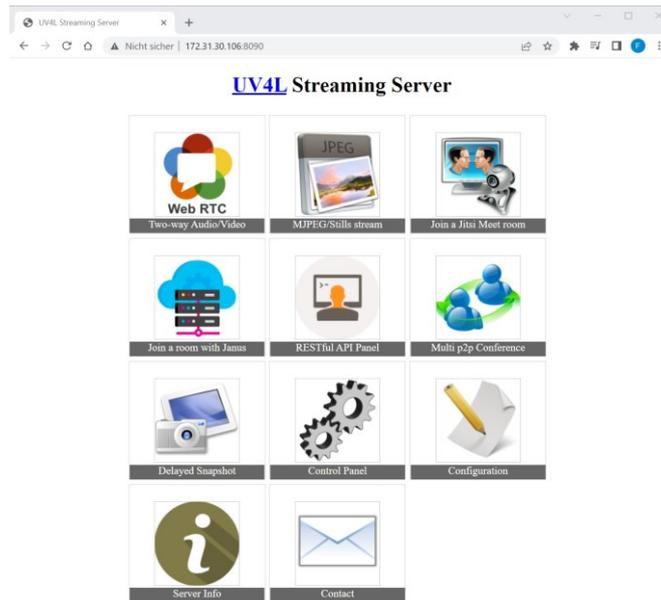


Abb. 58: UV4L-Streaming-Server-Webanwendung, Quelle: Eigene Darstellung.

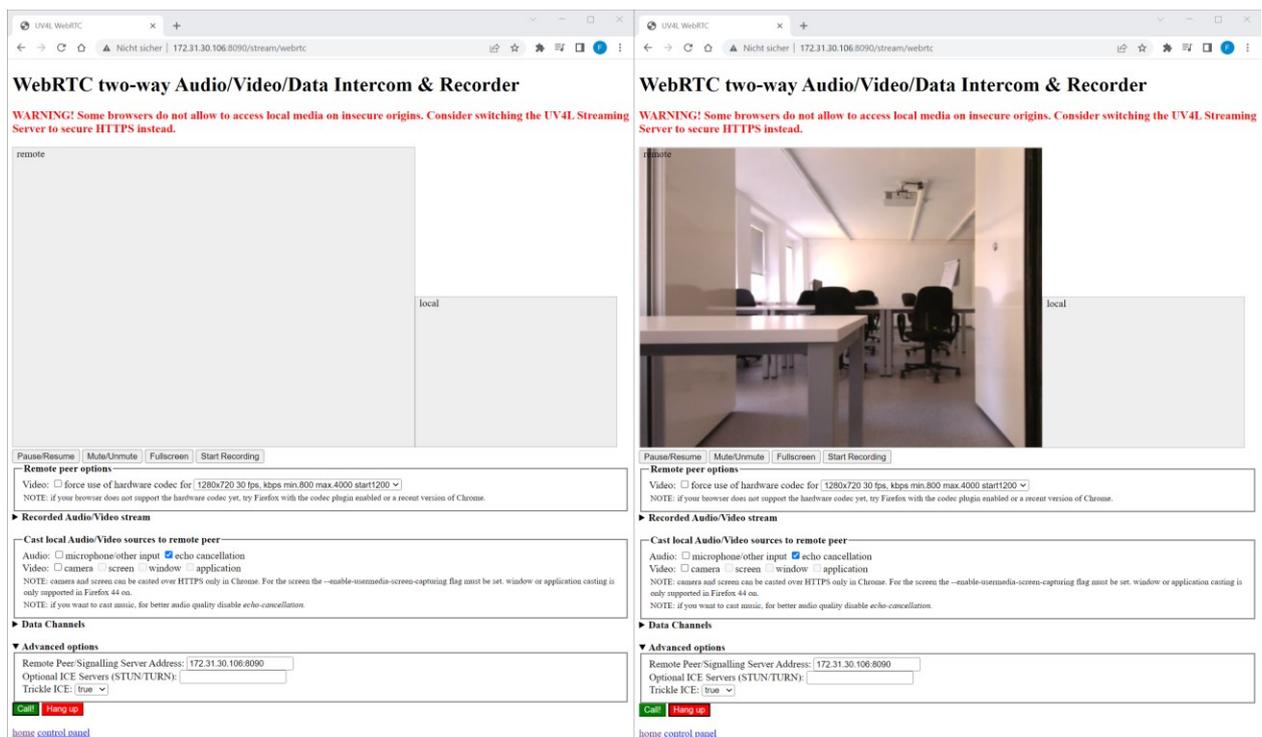


Abb. 59: UV4L-WebRTC-Stream-Weboberfläche und gestartete Videoübertragung, Quelle: Eigene Darstellung.

Damit ist die grundlegende Einrichtung und Funktionsprüfung abgeschlossen. Abschließend soll erwähnt werden, dass sich umfangreiche Informationen zu Funktionen und Konfigurationen mit Quellcode 13 über das Pi-OS-Terminal aufrufen lassen oder dass mittels eines Texteditors die Config-Files eingesehen werden können (/etc/uv4l/uv4l-uvc.conf, /usr/share/uv4l).

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo uv4l --driver uvc --help --
driver-help

Alternative:
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo uv4l --driver raspicam --help --
driver-help
```

Quellcode 13: Befehl zum Aufrufen der Treiberfunktionen und Konfigurationen in Pi OS Terminal, Quelle: Eigene Darstellung.

9.2.2 UFW Uncomplicated Firewall

Der Linux-Kernel verfügt mit Netfilter über eine bereits integrierte Firewall. Die Konfiguration und Festlegung des Firewall-Regelwerks erfolgt mittels Tabellen. Diese Tabellen können mit Programmen wie iptables den Bedürfnissen angepasst werden. Durch dieses direkte Vorgehen lässt sich die Firewall am genauesten konfigurieren und optimieren, jedoch ist es auch äußerst komplex. Um den Benutzern die Möglichkeit zu geben, auf einfache Weise Firewall-Regeln zu definieren, sind Programme wie *Uncomplicated Firewall* (UFW) zur Konfiguration der iptables entstanden. Im Vergleich zur direkten Anpassung der iptables bieten diese Programme nur eine eingeschränkte Funktionalität. Da jedoch keine spezialisierte oder optimierte Firewall für diese Arbeit vonnöten ist, reicht der Fähigkeitsumfang dieser Programme völlig aus, was dementsprechend auch auf das ausgewählte UFW zutrifft. Eine Übersicht über bereits reservierte Ports in einem Linux-System kann in der Regel in der Datei `/etc/services` eingesehen werden.

Wie in Quellcode 14 gezeigt, erfolgt die Installation im Pi-OS-Terminal typisch einer Linux-Distribution. Daraufaufgehend können die Firewall-Regeln für IP-Adressen, Ports oder Programme bzw. Anwendungen definiert werden und UFW kann gestartet werden.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo apt install ufw
```

Quellcode 14: Installation von *Uncomplicated Firewall* (UFW) im Pi-OS-Terminal, Quelle: Eigene Darstellung.

In UFW gibt es mit `limit` und `allow` zwei Befehle, um Datentransfer zu erlauben, wie in Quellcode 15 und Quellcode 16 gezeigt wird. Der `limit`-Befehl limitiert die Anzahl der Verbindungen auf sechs innerhalb 30 Sekunden. Die Freischaltung eines anwendungsspezifischen Ports ermöglicht die Ausführung der dazugehörigen Anwendung. Dasselbe gilt bei umgekehrtem Vorgehen, sprich der Freischaltung eines Programms. Soll die Freigabe auf ein bestimmtes Protokoll beschränkt werden, so kann dies durch die Angabe von `/tcp` bzw. `/udp` nach dem Port definiert werden. Für den Remotezugang ist es erforderlich, Port 22 (SSH) und Port 5900 (realVNC) freizugeben. Bei Port 80 und Port 443 handelt es sich um Standard-Ports für http und https. Für die Videoübertragung sind zwei freie Ports zu definieren. Im konkreten Beispiel erfolgt die Übertragung mittels Port 8090 und Port 9000. Da UV4L für das WebRTC-Signaling in einem lokalen Netzwerk die Ports 33635 bis 65535 beansprucht sind diese ebenfalls in der Firewall freizugeben.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw limit 22
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw limit 5900
```

Quellcode 15: Port-Freischaltung mit limitierter Verbindungsanzahl UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw allow 80
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw allow 443
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw allow 8090
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw allow 9000
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw allow 33635:65535
```

Quellcode 16: Port-Freischaltung UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Das mittels UFW festgelegte Regelwerk kann im Anschluss durch den in Quellcode 17 dargestellten Befehl im Pi-OS-Terminal ausgegeben und geprüft werden (Abb. 60).

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw show added
```

Quellcode 17: Installation von UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw show added
Added user rules (see 'ufw status' for running firewall):
ufw limit 22
ufw allow 80
ufw allow 443
ufw allow 8090
ufw limit 5900
ufw allow 9000
```

Abb. 60: Rückgabe der hinzugefügten Firewall-Regeln im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Sobald alle notwendigen Konfigurationen abgeschlossen sind, kann UFW gestartet werden, wie in Quellcode 18 und Abb. 61 dargestellt ist. In diesem Zug wird das Programm ebenfalls der Bootliste hinzugefügt und ist somit mit Systemstart aktiv.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw enable
```

Quellcode 18: Aktivierung von UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw enable
Firewall is active and enabled on system startup
```

Abb. 61: Aktivierung von UFW mit Rückgabe im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Der Status von UFW und des Regelwerks kann durch Quellcode 19 abgefragt und wie in Abb. 62 nummeriert ausgegeben werden. Für die Entfernung von Regeln kann dies sinnvoll sein, da die Nummerierung herangezogen werden kann.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw status
Alternative:
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw status numbered
```

Quellcode 19: Statusabfrage von UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo ufw status numbered
Status: active

      To Action From
      --
[ 1] 22  LIMIT IN Anywhere
[ 2] 80  ALLOW IN Anywhere
[ 3] 443  ALLOW IN Anywhere
[ 4] 8090 ALLOW IN Anywhere
[ 5] 5900 LIMIT IN Anywhere
[ 6] 9000 ALLOW IN Anywhere
[ 7] 22 (v6) LIMIT IN Anywhere (v6)
[ 8] 80 (v6) ALLOW IN Anywhere (v6)
[ 9] 443 (v6) ALLOW IN Anywhere (v6)
[10] 8090 (v6) ALLOW IN Anywhere (v6)
[11] 5900 (v6) LIMIT IN Anywhere (v6)
[12] 9000 (v6) ALLOW IN Anywhere (v6)
```

Abb. 62: Statusabfrage mit Nummerierung von UFW mit Rückgabe in Pi-OS-Terminal, Quelle: Eigene Darstellung.

Detailliertere Informationen zu UFW können durch den Befehl `man ufw` im Pi-OS-Terminal oder auf der Webseite <https://help.ubuntu.com/community/UFW> nachgeschlagen werden.

9.3 Dynamic DNS Service

Ein *Dynamic DNS Service* ermöglicht die Zuordnung eines festen Domain-Namens zu einer IP-Adresse. Dadurch ist es möglich, mit einer einzelnen Webadresse auf eine sich wechselnde IP-Adresse zu verweisen, wie es für Privatanwender üblich ist. Für diese Zwecke stehen eine Vielzahl an kostenpflichtigen und kostenlosen Anbietern zur Verfügung: Dynu, freedns.afraid.org, DuckDNS, No-IP, Securepoint DynDNS, Dynv6, ClickIP, CloudDNS, YDNS usw. Aufgrund der Übersichtlichkeit und der guten Dokumentation wird in dieser Arbeit der Dynu-dynamic-DNS-Service herangezogen.

Durch eine kostenlose Registrierung können bis zu vier Domains erstellt werden. Für diese Arbeit wird eine Sub-Domain mit der Bezeichnung `campus02-5G-VR.accesscam.org` erstellt, wie in Abb. 63 abgebildet. Die angelegten Domains und alle weiteren Einstellungen können durch das Einloggen über den Browser auf die Dynu-Webseite vorgenommen werden.⁶⁹

Abb. 63: Erstellung einer Sub-Domain, Quelle: Eigene Darstellung.

⁶⁹ Vgl. [dynu/Resources/Tutorials/DynamicDNS/GettingStarted](https://dynu.com/Resources/Tutorials/DynamicDNS/GettingStarted) (2022), Online-Quelle [22.10.2022].

Im Anschluss gilt es, ein Auto-IP-Update-Skript auf dem Raspberry Pi bzw. dem gewünschten Gerät zu implementieren. Dabei wird, wie in Quellcode 20 dargestellt, ein neues Verzeichnis angelegt und geöffnet.⁷⁰

```
Pi5G@RaspberryPiTeleoperation5G:~ $ cd ~
mkdir dynudns
cd dynudns
```

Quellcode 20: Erstellung und anschließendes Öffnen eines neuen Ordners im Pi-OS-Terminal, Quelle: Eigene Darstellung.

In diesem neuen Verzeichnis wird ein neues Dokument erstellt, in welches wiederum der Quellcode 22 geschrieben wird. Der Username ist im *Control Panel* von Dynu einsehbar, das Passwort kann im Plain-Text oder als MD5-Hash angegeben werden. Als Passwort wird das Account-Passwort verwendet, außer es wird ein eigenes IP-Update-Passwort angelegt. Der gezeigte Befehl aktualisiert alle Domains des Benutzers. Für die Aktualisierung einzelner Domain-Namen steht ein eigener Befehl zur Verfügung.⁷¹

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo nano dynu.sh
```

Quellcode 21: Erstellung und Öffnen eines neuen Dokuments im Pi-OS-Terminal, Quelle: Eigene Darstellung.

```
Pi5G@RaspberryPiTeleoperation5G:~ $ echo
url="https://api.dynu.com/nic/update?username=USERNAME&password=PASSWORD"
| curl -k -o ~/dynudns/dynu.log -K -
```

Quellcode 22: Dokumentinhalt für Auto-IP-Update im Pi OS Terminal, Quelle: Eigene Darstellung.

Das zuvor erstellte Dokument wird mittels des Quellcode 23 in eine ausführbare Datei umgewandelt, um es anschließend als Cron-Job im Hintergrund automatisiert in einem definierten Zeitintervall auszuführen.⁷²

```
Pi5G@RaspberryPiTeleoperation5G:~ $ chmod 700 dynu.sh
```

Quellcode 23: Umwandlung des Dokuments in eine ausführbare Datei im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Hierzu muss das erstellte Skript in die Cron-Tabelle hinzugefügt werden. Diese wird durch Quellcode 24 aufgerufen. Der neue Befehl wird an das Ende der Tabelle geschrieben und beinhaltet unter anderem das Zeitintervall, das mit fünf Minuten gewählt ist, sowie den Dateipfad des Skripts. Durch diese Implementation ist gewährleistet, dass das Gerät netzwerkunabhängig automatisch die aktuelle IP-Adresse an den Dynamic-DNS-Service übermittelt, sobald eine Internetverbindung besteht.⁷³

```
Pi5G@RaspberryPiTeleoperation5G:~ $ crontab -e
```

Quellcode 24: Öffnen der Cron-Tabelle im Pi-OS-Terminal, Quelle: Eigene Darstellung.

⁷⁰ Vgl. dynu/DynamicDNS/IPUpdateClient/RaspberryPi-Dynamic-DNS (2022), Online-Quelle [22.10.2022].

⁷¹ Vgl. dynu/DynamicDNS/IPUpdateClient/RaspberryPi-Dynamic-DNS (2022), Online-Quelle [22.10.2022].

⁷² Vgl. dynu/DynamicDNS/IPUpdateClient/RaspberryPi-Dynamic-DNS (2022), Online-Quelle [22.10.2022].

⁷³ Vgl. dynu/DynamicDNS/IPUpdateClient/RaspberryPi-Dynamic-DNS (2022), Online-Quelle [22.10.2022].

```
#Script executed every 5 minutes to keep hostnames updated to the most
current IP address.
*/5 * * * * ~/dynudns/dynu.sh >/dev/null 2>&1
```

Quellcode 25: Hinzufügen des Auto-IP-Update-Skripts zur Cron-Tabelle im Pi-OS-Terminal, Quelle: Eigene Darstellung.

9.4 Port Forwarding

Damit nach dem Einrichten eines dynamischen DNS-Service Zugriff von außerhalb des eigenen Netzwerks über das Internet erfolgen kann, ist eine Konfiguration des Routers erforderlich. Dafür stehen abhängig vom Modell bis zu zwei Lösungen zur Verfügung. Wird ein dynamischen DNS-Service vom gewählten Anbieter durch den Router unterstützt, so kann dieser im Web-Interface des Geräts eingerichtet werden. Ist diese Option nicht vorhanden, sind die entsprechenden Ports durch *Port Forwarding* weiterzuleiten, wie in Abb. 64 gezeigt. In dieser Arbeit werden die Ports *Secure Shell(SHH)* 22, *Virtual Network Computing(VNC)* 5900 sowie Web-Server 8090 und 9000 freigegeben. Mit der Einrichtung eines dynamischen DNS-Service wird zwar der Zugriff von außerhalb ermöglicht, jedoch kann es zu Problemen mit der Verbindung im eigenen Netzwerk kommen, wenn der Router nicht mit einem Network-Address-Translation-(NAT)-Loopback ausgestattet ist.

Application Name	WAN Connection	WAN Port	LAN Port	Device Name	Internal Client	Protocol	Status	Delete
Custom Settings	5gwan_TR069_INTERNET_1	8090~8090	8090~8090	RaspberryPiTeleoperation5G	192.168.1.169	TCP/UDP	Active	Delete
Custom Settings	5gwan_TR069_INTERNET_1	5900~5900	5900~5900	RaspberryPiTeleoperation5G	192.168.1.169	TCP/UDP	Active	Delete
Custom Settings	5gwan_TR069_INTERNET_1	9000~9000	9000~9000	RaspberryPiTeleoperation5G	192.168.1.169	TCP/UDP	Active	Delete
Custom Settings	5gwan_TR069_INTERNET_1	22~22	22~22	RaspberryPiTeleoperation5G	192.168.1.169	TCP/UDP	Active	Delete

Abb. 64: Web-Interface Nokia FastMile 5G Gateway 3.1 Routerkonfiguration, Quelle: Eigene Darstellung, <https://www.nokia.com/networks/products/fastmile-5g-gateways/> [Stand: 24.11.2022].

9.5 VR-Headset Programmierung

Für die Entwicklung von AR- und VR-Applikationen existieren neben diversen nativen Software-Development-Kits (SDKs), z. B. Varjo, mit den Game-Engines Unity und Unreal zwei große Entwicklungsplattformen. Diese verfügen bereits eine integrierte Unterstützung für einige AR/VR-Headsets. Alternativ werden Plug-ins von den Headset-Herstellern zur Verfügung gestellt. Ein weiterer Vorteil ist die Möglichkeit der Integration des Application-Interface OpenXR. Dabei handelt es sich um einen offenen Standard für AR- und VR-Anwendungen, durch den alle Geräte der beteiligten Hersteller unterstützt werden. Der Entwicklungsaufwand wird somit maßgeblich reduziert, da die Applikationsentwicklung auf einer Plattform für eine Vielzahl an Headsets durchgeführt werden kann. Zu Beginn der Vermarktung von VR-Headsets war es notwendig, die gerätenativen Entwicklungsplattformen zu nutzen. Dementsprechend musste dieser Vorgang oft mehrmals wiederholt werden. OpenXR ist für die Game-Engines Unity, Unreal, Blender und Godot verfügbar.

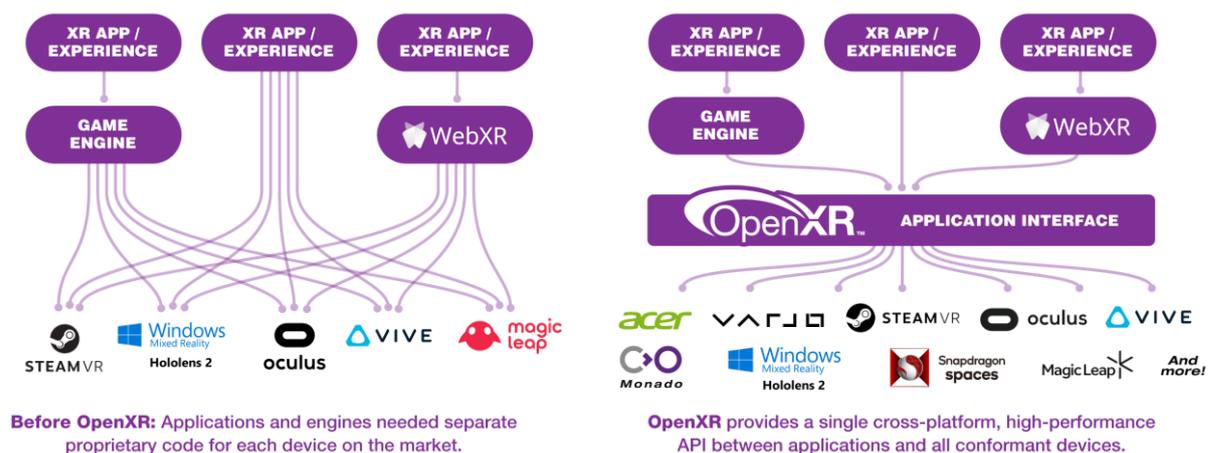


Abb. 65: OpenXR Application Interface, Quelle: <https://www.khronos.org/openxr/> [Stand: 24.11.2022].

Aufgrund der Verwendung der Oculus Quest 2 als Wiedergabegerät und der größeren Community sowie des höheren Marktanteils wurde die Entwicklungsplattform Unity gegenüber Unreal für diese Arbeit gewählt. Ein weiterer wesentlicher Aspekt, in dem sich die Game-Engines unterscheiden, ist die Programmiersprache. In Unity werden die Skripts in C# geschrieben, während in Unreal C++ verwendet wird.

Der ursprüngliche Ansatz, das UV4L-WebRTC-Interface für die Video-, Audio- und Datenübertragung zu nutzen, musste angesichts mangelnder Unterstützung einer Browserintegration in Unity aufgegeben werden. Zwar sind mit Unity-Webview und 3D WebView Software-Plug-ins verfügbar, jedoch wird von ersterem Windows nicht unterstützt und bei zweiterem handelt es sich um ein Produkt von Vuplex, das über den Unity Asset Store erworben werden kann. Daher blieb als Alternative nur die Entwicklung einer Unity-Applikation mit direkter Integration von WebRTC.⁷⁴

⁷⁴ Vgl. Nakamaru, Online-Quelle [24.11.2022], Vgl. Vuplex, Online-Quelle [24.11.2022].

WebRTC wurde erstmals 2011 veröffentlicht und wird zu den ‚neuen‘ *Real-Time-Communication*-Protokollen gezählt. Da es diverse Besonderheiten aufweist, ist es zunächst erforderlich, zu verstehen, wie ein Verbindungsaufbau abläuft, bevor mit einer Implementierung in Unity begonnen werden kann. Mit WebRTC ist es möglich, eine Peer-to-Peer-Verbindung zur Medien- bzw. Datenübertragung herzustellen. Dazu müssen die Teilnehmer zuerst lokalisiert werden. Hierfür ist der Austausch der Verbindungsbedingungen notwendig, der als *Signaling* bezeichnet wird. Der Prozess ist zwar erforderlich für den Aufbau einer WebRTC-Verbindung, jedoch nicht Teil des Protokolls oder standardisiert. Dadurch kann jedes *Signaling*-Protokoll und Gateway dafür herangezogen werden, wie HTTPS, WebSocket, XMPP und MQTT. Der Austausch des Session Description Protocol (SDP) via der WebRTC-offer/answer-Architektur zwischen den Teilnehmern A und B über einen Signaling-Server wird in Abb. 66 dargestellt. Der Zusammenhang zwischen *Signaling* sowie dem STUN- und TURN-Server beim Verbindungsaufbau ist anhand der Topologien in Abb. 67 ersichtlich.⁷⁵

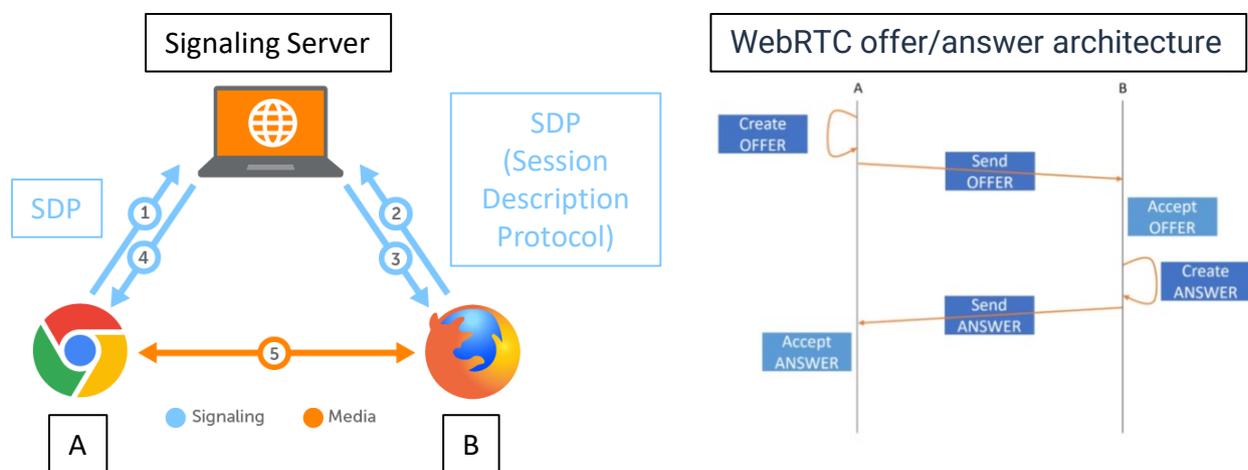


Abb. 66: WebRTC-Signaling Server und WebRTC offer/answer Architektur,
 Quelle: <https://www.wowza.com/blog/webrtc-signaling-servers>,
<https://bloggeek.me/webrtcglossary/isep/>,
 [Stand: 24.11.2022] (leicht modifiziert).

⁷⁵ Vgl. Grigorik (2013), S. 320-324, Vgl. Levent-Levi (2020), Online-Quelle [24.11.2022], Vgl. Levent-Levi, Online-Quelle [24.11.2022], Vgl. Levent-Levi (2020), Online-Quelle [24.11.2022], Vgl. Levent-Levi (2020), Online-Quelle [24.11.2022], Vgl. Levent-Levi (2020), Online-Quelle [24.11.2022], Vgl. freedomrobotics, Online-Quelle [24.11.2022].

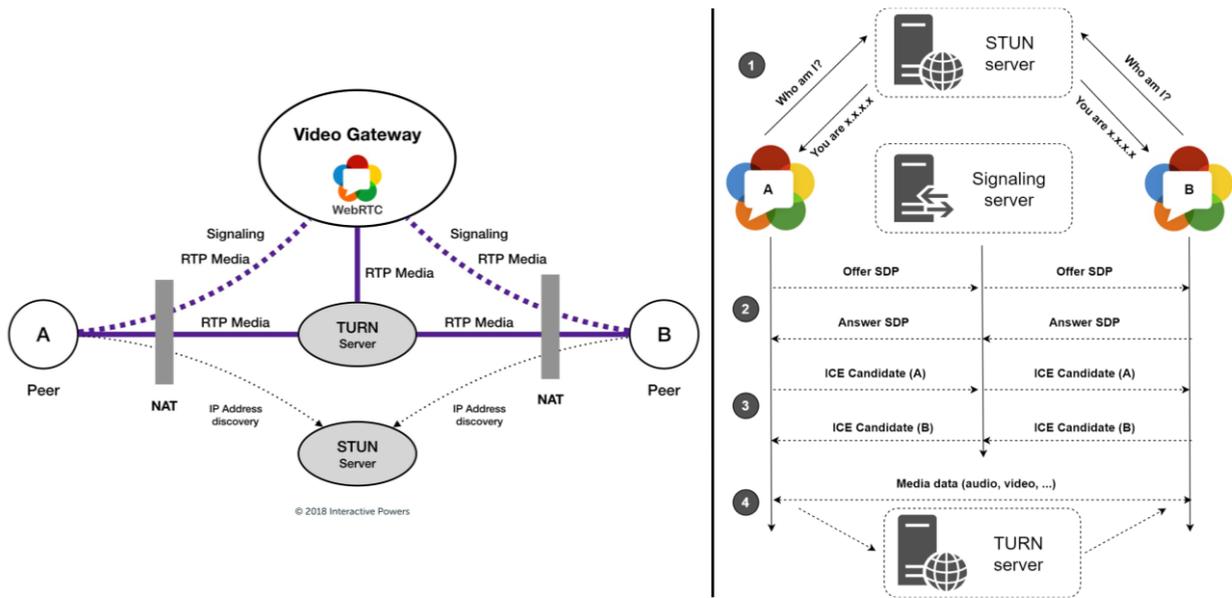


Abb. 67: WebRTC-Verbindungsherstellung, Quelle: <https://docs.freedomrobotics.ai/docs/how-webrtc-works>, [Stand: 24.11.2022].

Das Streaming-Server-Modul von UV4L enthält ein *Signaling*-Protokoll und basiert auf dem Nachrichtenaustausch via WebSocket im JSON-Format.⁷⁶ Die Kommunikation kann in C# durch einbinden unterschiedlicher Plug-in-Pakete wie System.Net.Sockets, SignalR, NativeWebSocket und WebSocket-Sharp erfolgen. Ist ein WebSocket-Kanal zwischen den Teilnehmern etabliert kann das offer/answer-Prozedere für das Signaling ausgeführt werden. In Quellcode 26 ist die erste JSON-Nachricht dargestellt, mit der der Verbindungsprozess gestartet wird.

```
{
  what: "call",
  options: {
    force_hw_vcodec: true,
    vformat: 30,
    trickle_ice: true
  }
}
```

Quellcode 26: Call-Request UV4L Signaling-Protokoll, Quelle: Eigene Darstellung.

Die weitere Implementierung des WebRTC-Verbindungsprozesses kann dem Unity-WebRTC-Paket entnommen werden. Darin sind zahlreiche Beispiele mit lokalem Stream für unterschiedliche Szenarien enthalten. Die Videoübertragung mit einer Kamera ist Teil der VideoReceive-Szene.

⁷⁶ Vgl. [linux-projects/webrtc-signalling/](https://docs.freedomrobotics.ai/docs/how-webrtc-works), Online-Quelle [24.11.2022].

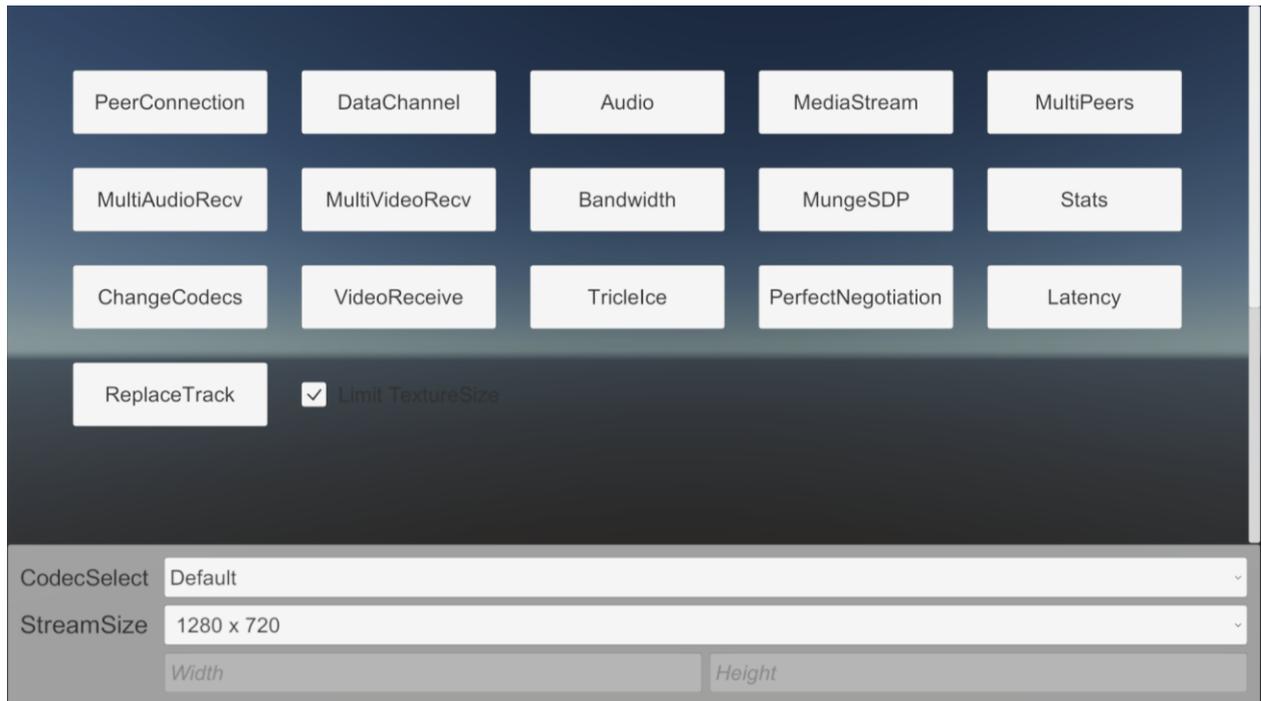


Abb. 68: Unity-WebRTC-Paket, Quelle: Eigene Darstellung.

In Abb. 69 ist eine User-Interface(UI)-Oberfläche mit dem Minimum an Objekten dargestellt, die für eine WebRTC-Übertragung notwendig sind. Das Projekt muss mit einer der beiden Rendering-Umgebungen – Universal Render Pipeline (URP) oder High Definition Render Pipeline (HDRP) – erstellt werden. Ist eine VR-Unterstützung gewünscht, so ist eine Ergänzung der Umgebungen zu URP + VR bzw. HDRP + VR erforderlich. Wie in der Grafik sichtbar ist, wird ein Call-Button als Auslöser zur Herstellung der Verbindung benötigt. Die Übertragung wird mit dem Hang-up-Button beendet. Dasselbe geschieht beim Schließen der Anwendung durch das WebRTC-Protokoll. Die Darstellung der Videoübertragung erfolgt durch projizieren der Bilddaten auf ein Flächenobjekt.

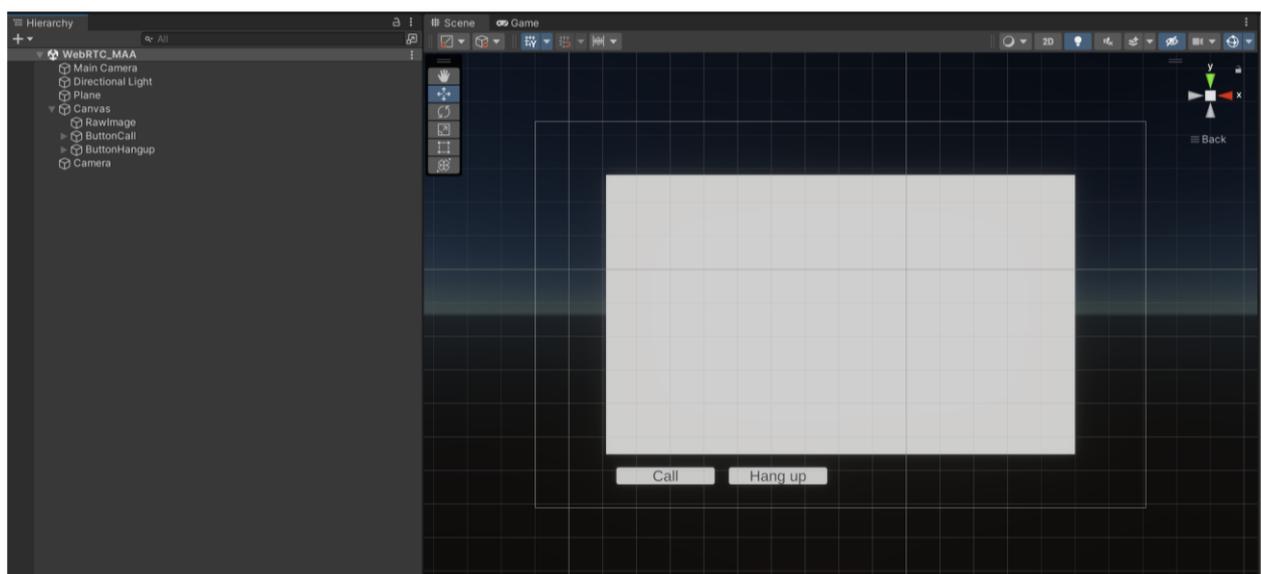


Abb. 69: Unity-WebRTC-UI, Quelle: Eigene Darstellung.

Für VR-Anwendungen ist das Hinzufügen des XR-Plug-in-Management-Pakets erforderlich, um die VR-Headset-Unterstützung und Ähnliches zu konfigurieren. Zwar lässt sich auch die Main-Camera in ein XR-Rig mit separater Ausgabe für das linke und rechte Auge konvertieren, jedoch ist dieses Vorgehen nicht für alle Anwendungsfälle geeignet. Daher wird ein XR-Rig wie in Abb. 70 gezeigt verwendet. Dieses ist in der Struktur an das ZED-Unity-Plug-in angelehnt, das auf der Firmenwebseite von Stereolabs bezogen werden kann. Durch den Aufbau aus zwei getrennten Kameras kann die Bildausgabe für jedes Auge unabhängig voneinander bestimmt werden. Abschließend muss das in Abb. 69 gezeigte WebRTC-Interface mit dem XR-Rig aus Abb. 70 kombiniert werden. Dabei ist es sinnvoll, mit mehreren Szenen zu arbeiten. Darüber hinaus sollte in Betracht gezogen werden, die Handhabung durch Hinzufügen einer zusätzlichen Kamera für die Ausgabe an einem Desktopmonitor zu erleichtern.

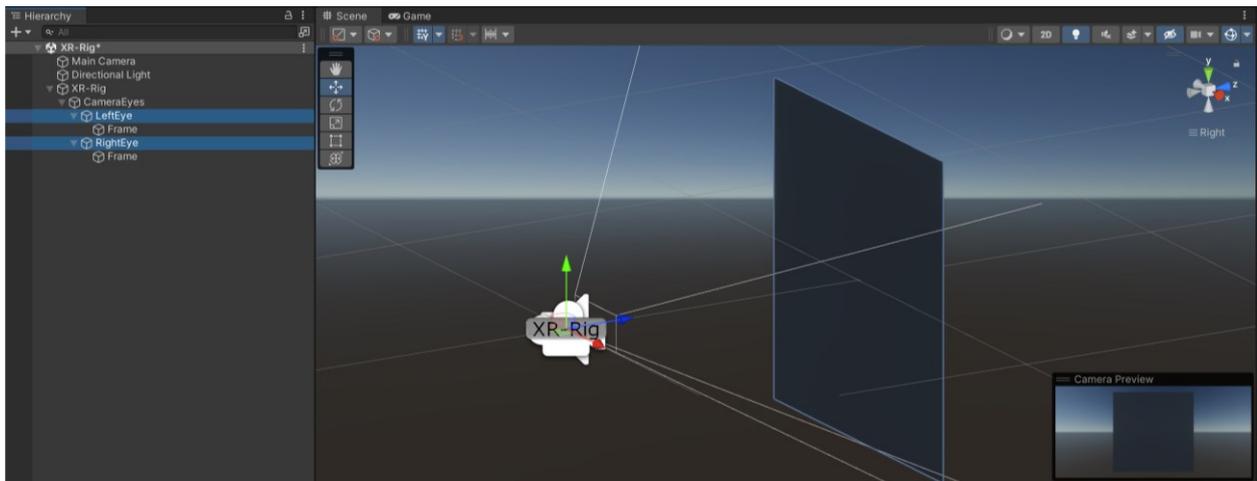


Abb. 70: Unity-XR-Stereo-Rig, Quelle: Eigene Darstellung.

9.5.1 Datenkanal

Sobald eine Peer-to-Peer-WebRTC-Verbindung etabliert ist, kann neben Video- und Audioübertragung ein Datenkanal geöffnet werden. Dieser kann für Sprachchat, Filetransfer und Ähnliches genutzt werden. Für die Entwicklung von Virtual-Reality(VR)-, Mixed-Reality(MR)- und Augmented-Reality(AR)-Anwendungen wird von der Game-Engine das Unity XR-Plug-in-Framework bereitgestellt. Dieses unterstützt die folgenden Plattformen: ARKit, ARCore, Microsoft HoloLens, Windows Mixed Reality, Magic Leap, Oculus, OpenXR und PlayStation VR. Dadurch ist es möglich, via InputTracking auf die Rotationsbewegung des VR-Headsets zuzugreifen. Die abgefragten Parameter können im Anschluss z. B. als JSON-String an den anderen Teilnehmer übermittelt werden. Eine Übersicht der Unity XR-Plug-in-Framework-Struktur ist in Abb. 71 ersichtlich.⁷⁷

⁷⁷ Vgl. Unity Technologies (2022), Online-Quelle [25.11.2022].

Unity XR Tech Stack

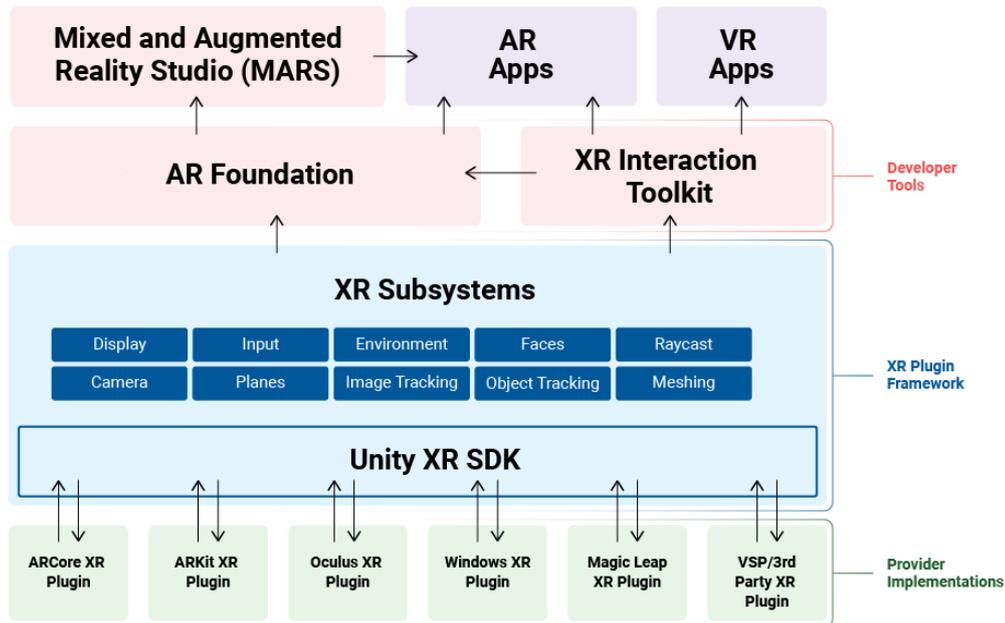


Abb. 71: Unity XR-Plugin-Framework-Struktur, Quelle: <https://docs.unity3d.com/Manual/XRPluginArchitecture.html>, [Stand: 25.11.2022].

Auf der Seite des Empfängers, des Raspberry Pi 4, ist ein Python-Skript zu schreiben, das ebenfalls eine Verbindung zum Datenkanal herstellt. Anschließend kann der JSON-String geparkt werden und für die Steuerung der Servo-Motoren-Positionen herangezogen werden. Ein Beispiel für die Implementierung einer unidirektionalen Datenübertragung zur Servo-Steuerung kann über die Quellenangabe in der Fußnote bezogen werden.⁷⁸

9.5.2 Servo-Motor-Steuerung

Die Steuerung der Servomotoren wird mittels der Adafruit-CircuitPython-Servokit-Bibliothek für das PCA9685-Servo-Controller-Board bewerkstelligt. Der Installation des Servokits ist jene der Adafruit-Blinka-Bibliothek vorgelagert, wie in Quellcode 27 ersichtlich.⁷⁹

```
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo pip3 install adafruit-blinka
Pi5G@RaspberryPiTeleoperation5G:~ $ sudo pip3 install adafruit-circuitpython-servokit
```

Quellcode 27: Installation der Adafruit-CircuitPython-Servokit Bibliothek im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Eine Überprüfung, ob das Servo-Controller-Board vom Raspberry Pi 4 an der I²C-Schnittstelle erkannt wurde, kann durch Ausführung des Quellcode 28 erreicht werden.

⁷⁸ Vgl. linux-projects/uv4l/tutorials/web rtc-data-channels/, Online-Quelle [25.11.2022].

⁷⁹ Vgl. Earl (2012), Online-Quelle [25.11.2022].

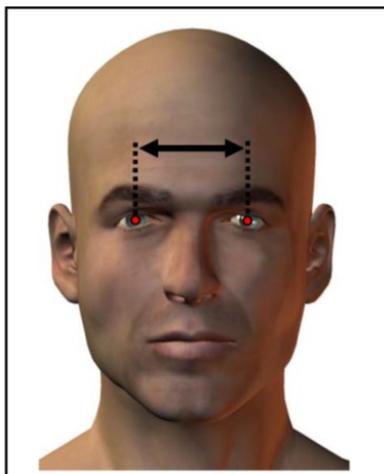
```
Pi5G@RaspberryPiTeleoperation5G:~ $ i2cdetect -y 1
```

Quellcode 28: I²C-Schnittstellendetektion im Pi-OS-Terminal, Quelle: Eigene Darstellung.

Im Anschluss können die Servomotoren durch diese Pakete mit einem Python-Skript gesteuert werden. Dadurch können der Empfang und das Parsen der JSON-Strings für den Erhalt der Kopffrotation in einem Skript ausgeführt werden. Abhängig von den verwendeten Servomotoren und vom Aufbau bzw. Stativ kann es erforderlich sein, den üblicherweise 180° umfassenden Stellbereich zu begrenzen, um Kollisionen zu vermeiden. Weiters kann der zu durchlaufende Bewegungsrahmen von 0–180° an den einzelnen Servomotoren einer Streuung unterliegen. Diesem Effekt kann durch die Parametrierung der minimalen und maximalen Pulsweite entgegengewirkt werden.⁸⁰

9.6 Stereoskopisches Kamerasetup

Bei einem stereoskopischen Setup werden zwei Kameras verwendet, um das linke und rechte Auge separat zu repräsentieren. Für das Erreichen des bestmöglichen Tiefenwahrnehmungseffektes ist es notwendig, das Aufnahme- und Wiedergabesetting für die jeweilige Person einzurichten. Demnach muss der Abstand zwischen den Mittelpunkten der Kameralinsen und der Wiedergabebildschirme dem Pupillenabstand (Interpupillary Distance, IPD) des Benutzers entsprechen. Diesbezüglich wurde eine Recherche durchgeführt, um festzustellen, welche Schwankungsbreite ein stereoskopisches Kamerasetup aufnehmen können sollte, damit es für eine möglichst große Zahl an Individuen geeignet ist. Als repräsentative Datenquelle wird im weiteren Verlauf des Kapitels der *TECHNICAL REPORT NATICK/TR-15/007* der U.S. ARMY herangezogen. In Tab. 2 wird die Messgröße des Pupillenabstands spezifiziert und diverse statistische Kenngrößen für die weibliche und männliche Testgruppe angeführt. Die Messwerte und ihre Verteilung sind in Tab. 3 tabellarisch dargestellt und in Abb. 72 mittels eines Säulendiagramms bildlich visualisiert.



FEMALES	
	mm
MEAN	61,7
STD ERROR (MEAN)	0,10
STANDARD DEVIATION	3,60
STD ERROR (STD DEV)	0,10
MINIMUM	5,10
MAXIMUM	7,45
SKEWNESS	0,01
KURTOSIS	3,00
COEFFICIENT OF VARIATION	5,9%
NUMBER OF PARTICIPANTS	1986

MALES	
	mm
MEAN	64,0
STD ERROR (MEAN)	0,10
STANDARD DEVIATION	3,40
STD ERROR (STD DEV)	0,00
MINIMUM	5,30
MAXIMUM	7,70
SKEWNESS	0,14
KURTOSIS	3,30
COEFFICIENT OF VARIATION	5,3%
NUMBER OF PARTICIPANTS	4082

Tab. 2: Statistische Kennwerte der Interpupillary Distance, Quelle: 2012 ANTHROPOMETRIC SURVEY OF U.S. ARMY PERSONNEL: METHODS AND SUMMARY STATISTICS, S. 149-150 (leicht modifiziert).

⁸⁰ Vgl. Earl (2012), Online-Quelle [25.11.2022].

FREQUENCY TABLE										
FEMALES					MALES					
F	FPct	CumF	CumFPct	mm		F	FPct	CumF	CumFPct	
1	0,05	1	0,05	50,5		50,5				
5	0,25	6	0,3	51,5	-	51,5				
6	0,30	12	0,6	52,5	-	52,5				
33	1,66	45	2,27	53,5	-	53,5	3	0,07	3	0,07
33	1,66	78	3,93	54,5	-	54,5	6	0,15	9	0,22
62	3,12	140	7,05	55,5	-	55,5	15	0,37	24	0,59
83	4,18	223	11,23	56,5	-	56,5	31	0,76	55	1,35
106	5,34	329	16,57	57,5	-	57,5	36	0,88	91	2,23
151	7,60	480	24,17	58,5	-	58,5	81	1,98	172	4,21
198	9,97	678	34,14	59,5	-	59,5	118	2,89	290	7,1
193	9,72	871	43,86	60,5	-	60,5	209	5,12	499	12,22
243	12,24	1114	56,09	61,5	-	61,5	309	7,57	808	19,79
208	10,47	1322	66,57	62,5	-	62,5	442	10,83	1250	30,62
191	9,62	1513	76,18	63,5	-	63,5	448	10,98	1698	41,6
160	8,06	1673	84,24	64,5	-	64,5	512	12,54	2210	54,14
112	5,64	1785	89,88	65,5	-	65,5	478	11,71	2688	65,85
80	4,03	1865	93,91	66,5	-	66,5	393	9,63	3081	75,48
51	2,57	1916	96,48	67,5	-	67,5	324	7,94	3405	83,41
28	1,41	1944	97,89	68,5	-	68,5	250	6,12	3655	89,54
21	1,06	1965	98,94	69,5	-	69,5	143	3,5	3798	93,04
12	0,60	1977	99,55	70,5	-	70,5	123	3,01	3921	96,06
5	0,25	1982	99,8	71,5	-	71,5	74	1,81	3995	97,87
3	0,15	1985	99,95	72,5	-	72,5	36	0,88	4031	98,75
0	0,00	1985	99,95	73,5	-	73,5	25	0,61	4056	99,36
1	0,05	1986	100	74,5	-	74,5	13	0,32	4069	99,68
				75,5	-	75,5	6	0,15	4075	99,83
				76,5	-	76,5	4	0,1	4079	99,93
				77,5	-	77,5	3	0,07	4082	100

F...Anzahl der Probanden
 FPct...prozentualer Anteil der Probanden
 CumF...kumulierte Anzahl der Probanden
 CumFPct...kumulierter prozentualer Anteil der Probanden

Tab. 3: Häufigkeitstabelle Interpupillary Distance,
 Quelle: 2012 ANTHROPOMETRIC SURVEY OF U.S. ARMY PERSONNEL: METHODS AND SUMMARY STATISTICS,
 S. 150 (leicht modifiziert).

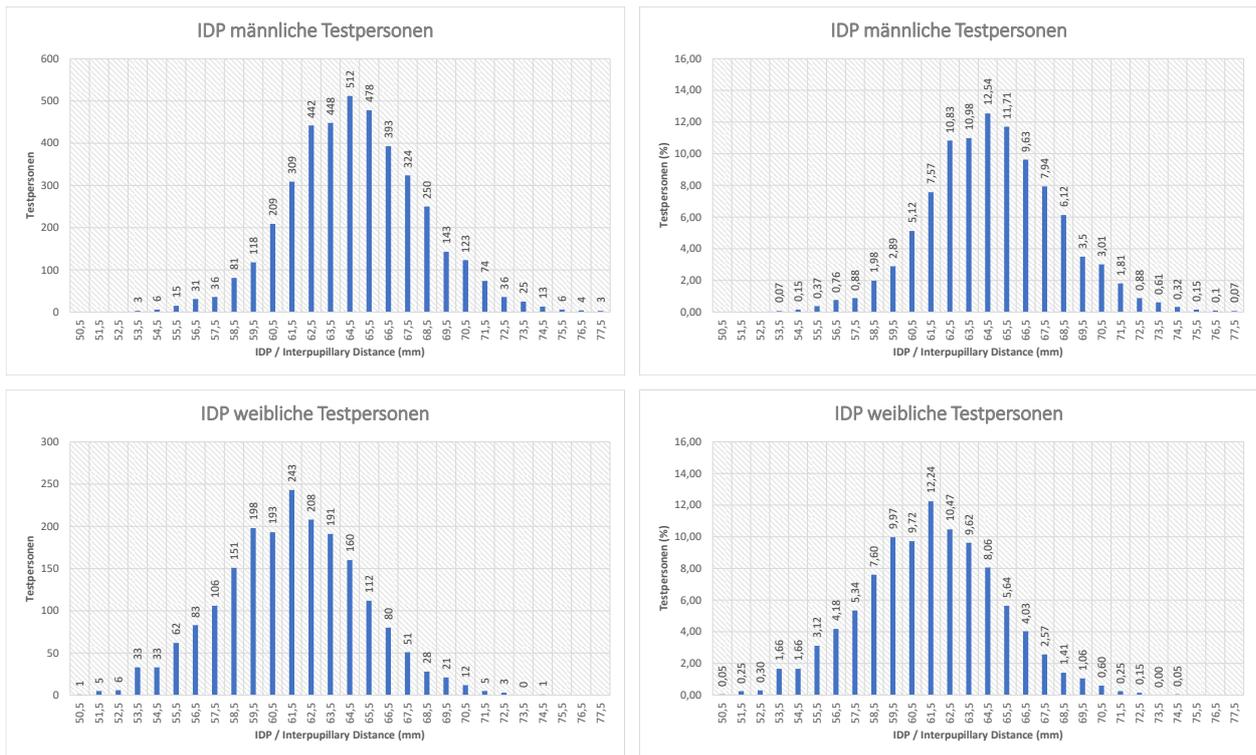


Abb. 72: Verteilung der Interpupillary Distance der männlichen und weiblichen Testpersonen absolut und prozentual, Quelle: Eigene Darstellung.

Wird der Verstellbereich der verfügbaren VR-Headsets – der Oculus Quest 2 mit drei Positionen (58 mm, 63 mm und 68 mm) und der HTC Vive Pro (61–72 mm) – auf die Diagramme in Abb. 72 angewandt, lassen sich die nachfolgend angeführten Aussagen treffen.⁸¹ Beide Geräte bilden einen großen Teil der Distribution ab, befinden sich jedoch im oberen Bereich des Spektrums. Dies trifft männliche Nutzer in geringerem Ausmaß als weibliche. Ein beträchtlicher Prozentsatz der Nutzerinnen hat demnach nicht die Möglichkeit, das Gerät an die eigenen Bedürfnisse anzupassen. Ob die vorhandenen Verstellrahmen von den Produzenten definiert wurden oder durch technische Limitationen zustande kommen, ist nicht bekannt. Ungeachtet dieser Erkenntnis wurde hinsichtlich der Konstruktion der Adapterplatte für das stereoskopische Kamerasetup ein Verstellbereich von 58–72 mm definiert. Dieses Vorgehen ist damit begründet, dass eine einseitige Erweiterung als nicht sinnvoll erachtet wurde. Bei zukünftigen Beschaffungen von VR-Headsets sollte die Variabilität zur Anpassung an Nutzerbedürfnisse ein fixer Bestandteil der Auswahlkriterien sein, um die Technologie einer möglichst breiten Masse zugänglich zu machen. In der praktischen Umsetzung wurde eine Adapterplatte für die in Kapitel 8 ausgewählten Komponenten via Computer-aided-Design-Software (CAD) konstruiert und mit einem 3D-Drucker hergestellt. Darüber hinaus wurde eine Skala zur Erleichterung der Einstellung des Linsenabstandes angebracht. Der Aufbau des stereoskopischen Kamerasetups ist in Abb. 73 dargestellt.

⁸¹ Vgl. Brown, Online-Quelle [26.11.2022].

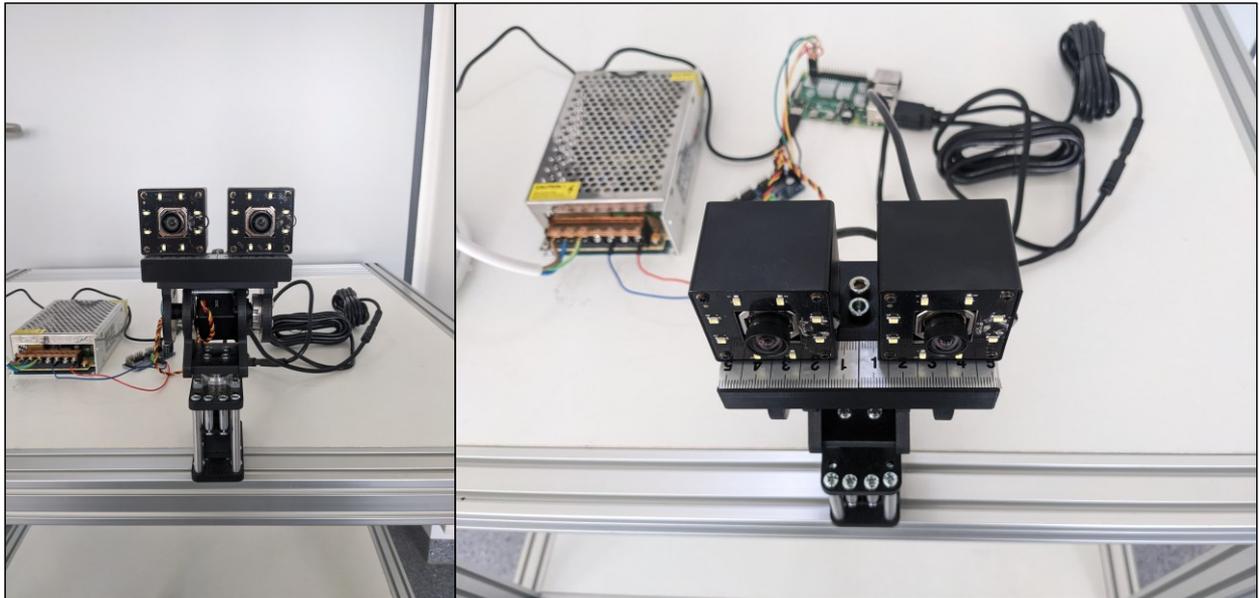


Abb. 73: Stereoskopisches Kamerasetup auf motorisiertem Kamerastativ, Quelle: Eigene Darstellung.

9.7 Netzwerkmessung

Dieses Kapitel beinhaltet die Schritte ausgeführt, die für das Durchführen von Messungen in einem 5G-Netzwerk notwendig sind. Einleitend mit der Beschreibung der Ausgangslage, über Konfiguration der Testgeräte bis zum Ausführen erster Testmessungen.

9.7.1 Ausgangslage

Neben den in Kapitel 6 beschriebenen Verfahren gibt es noch eine Vielzahl anderer Arten und Möglichkeiten, Netzwerkmessungen durchzuführen. Da in dieser Arbeit der Fokus auf einer niedrigen Latenzzeit liegt, wird das auf UDP basierende Streaming-Protokoll WebRTC zur Datenübertragung verwendet. Für diesen Anwendungsfall wird TWAMP als geeignet angesehen, weshalb in diesem Kapitel darauf der Fokus gelegt wird. In einem ersten Versuch wurde in einem lokalen Netzwerk eine Messung zwischen zwei Computern mittels der Twamp-gui⁸² durchgeführt. Dieser Ansatz wurde jedoch zugunsten der Open-Source-Software perfSONAR (**performance Service-Oriented Network monitoring ARchitecture**) verworfen. Dieses Programm dient zur Durchführung von End-to-End-Netzwerkmessungen, Archivierung und Visualisierung. Eine schematische Darstellung der einzelnen Pakete bzw. der gesamten Software kann Abb. 74 entnommen werden.⁸³

⁸² Vgl. Demirten (2022), Online-Quelle [16.11.2022].

⁸³ Vgl. perfSONAR/intro_about (2022), Online-Quelle [16.11.2022].

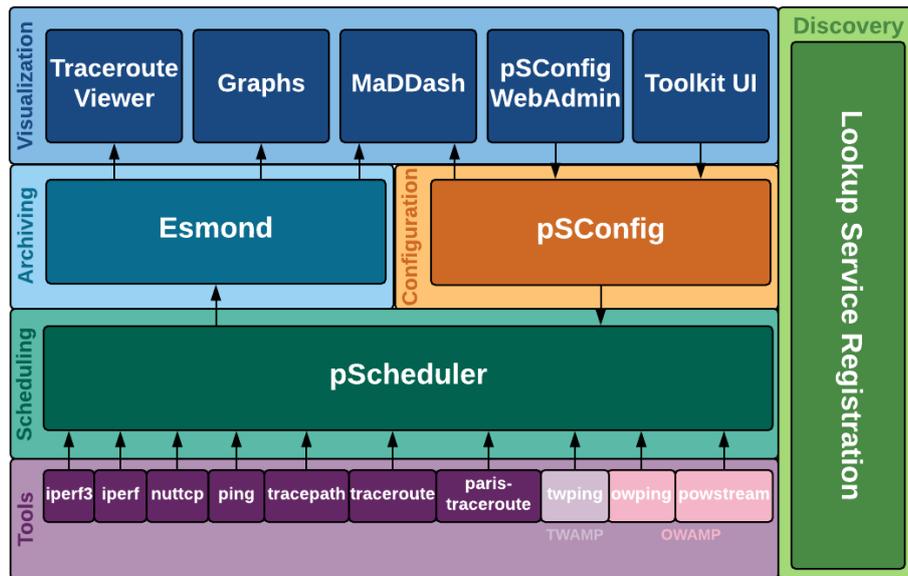


Abb. 74: Übersicht perfSONAR-Umfang, Quelle: https://docs.perfsonar.net/intro_about.html [Stand: 16.11.2022].

perfSONAR kann in unterschiedlichem Umfang auf den Linux-Distributionen CentOS 7, Ubuntu und Debian installiert und mittels Docker-Container auch auf Mac OS und Windows betrieben werden. An dieser Stelle ist zu erwähnen, dass die Entwicklung auf CentOS 7 stattfindet und dann auf die anderen Systeme ausgerollt wird.⁸⁴ Da zum Zeitpunkt der Umsetzung nicht alle Softwareinhalte vollständig verfügbar bzw. aktualisiert waren, musste der ursprüngliche geplante Messaufbau abgewandelt werden. Wie in Abb. 75 gezeigt, musste die Messung von Variante 1, Rechner zu Raspberry, durch Variante 2, Rechner zu Rechner, aufgrund der fehlenden Unterstützung der aktuellen Debian-Version 11 Bullseye ersetzt werden. Eine Kompatibilitätsherstellung war nicht möglich, da eine vollständige Auflösung der Paketabhängigkeiten nicht erreicht werden konnte. Für den PC konnte ebenfalls kein *CentOS perfSONAR Toolkit ISO* als Download bezogen werden, weshalb es erst selbst Schritt für Schritt erstellt werden musste.

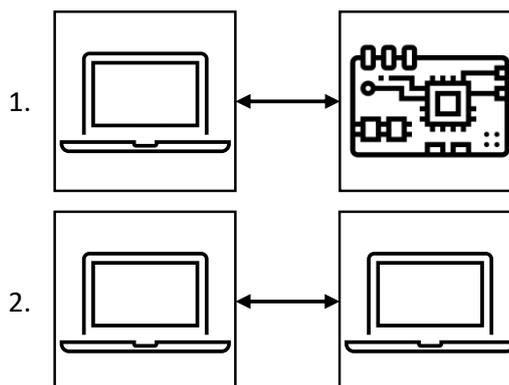


Abb. 75: Netzwerktestendgeräte: 1. Computer zu Raspberry Pi 4, 2. Computer zu Computer, Quelle: Eigene Darstellung, Flaticon.com(Raspberry Pi).

⁸⁴ Vgl. [perfsonar/install_options](#) (2022), Online-Quelle [16.11.2022].

9.7.2 Installation CentOS 7

Um möglichst flexibel agieren zu können, erfolgt die Installation von CentOS 7 auf einem USB-Stick. Hierfür werden zwei USB-Sticks benötigt, einer als Träger einer CentOS 7-Live-Version und ein zweiter als Zieldatenträger für die vollwertige Installation. Dazu kann ein jeder Rechner mit zwei freien USB-Ports verwendet werden, um die CentOS 7-Live-Version beim Systemstart zu booten und anschließend die Installation auszuführen. Eine kabelgebundene Netzwerkverbindung ist bei diesem Vorgang zu empfehlen, da wie beim hier verwendeten Rechner hp probook 650 g2 die Unterstützung des WLAN-Chips erst händisch hergestellt werden muss. Des Weiteren ist zu empfehlen, den Netzwerkmanager zur Bootliste hinzuzufügen, um ihn mit Systemstart auszuführen, wie in Quellcode 29.⁸⁵

```
[Cnetos7USB@centos7 ~]$ sudo systemctl enable NetworkManager
```

Quellcode 29: Netzwerkmanager mit Systemstart ausführen im CentOS 7-Terminal, Quelle: Eigene Darstellung.

Die Ermittlung des WLAN-Chipsets kann mittels Quellcode 30 durchgeführt werden. Bei der hier verwendeten Hardware ist ein Broadcom-Corp.-BCM43228-Chip verbaut, der Ablauf der Treiberinstallation kann als Anschauungsbeispiel der Fußnote ⁸⁶ entnommen werden.

```
[Cnetos7USB@centos7 ~]$ lspci  
or  
[Cnetos7USB@centos7 ~]$ lshw -class network
```

Quellcode 30: Ermitteln des WLAN-Chipsets im CentOS 7-Terminal, Quelle: Eigene Darstellung.

Die Einrichtung eines TigerVNC-Servers wurde kurz getestet, jedoch aufgrund des begrenzten Nutzens eines virtuellen Desktops und des erhöhten Aufwandes, um eine Spiegelung des Desktops zu ermöglichen, wieder verworfen. Im Gegensatz dazu stellt die Herstellung der Kompatibilität zum Extended-File-Allocation-Table(exFAT)-Dateisystem eine erhebliche Vereinfachung des Datentransfers dar. Dadurch ist es möglich, mittels USB-Stick Dateien zwischen CentOS 7 und Windows auszutauschen oder config-Files und Ähnliches extern zu sichern. Die Unterstützung von exFAT in CentOS 7 wird wie in Quellcode 31 gezeigt ermöglicht.⁸⁷

```
[Cnetos7USB@centos7 ~]$ sudo yum install epel-release  
[Cnetos7USB@centos7 ~]$ sudo rpm -v --import  
http://li.nux.ro/download/nux/RPM-GPG-KEY-nux.ro  
[Cnetos7USB@centos7 ~]$ sudo rpm -Uvh  
http://li.nux.ro/download/nux/dextop/el7/x86_64/nux-dextop-release-0-  
5.el7.nux.noarch.rpm  
[Cnetos7USB@centos7 ~]$ sudo yum install exfat-utils fuse-exfat
```

Quellcode 31: Herstellung der Unterstützung von exFAT im CentOS 7-Terminal, Quelle: Eigene Darstellung.

⁸⁵ Vgl. Red Hat Customer Portal (2022), Online-Quelle [16.11.2022].

⁸⁶ ELRepo.org (2022), Online-Quelle [16.11.2022].

⁸⁷ Vgl. Linuxize (2022), Online-Quelle [16.11.2022].

9.7.3 perfSONAR

Dieses Unterkapitel beinhaltet lediglich die elementaren Punkte, die für den praktischen Teil dieser Arbeit relevant sind, und stellt nur einen Ausschnitt der Softwareumfänge dar. Eine umfangreiche Dokumentation zur Handhabung, Konfiguration und zu Einsatzmöglichkeiten von perfSONAR ist auf der Webseite verfügbar. Wie in Abb. 76 gezeigt, können Netzwerktests in unterschiedlichem Umfang und mit unterschiedlicher Komplexität in lokalen und globalen Netzwerken durchgeführt werden. Wie bereits in Kapitel 9.7.1 erläutert, umfasst der betrachtete Anwendungsfall lediglich zwei Testknoten.

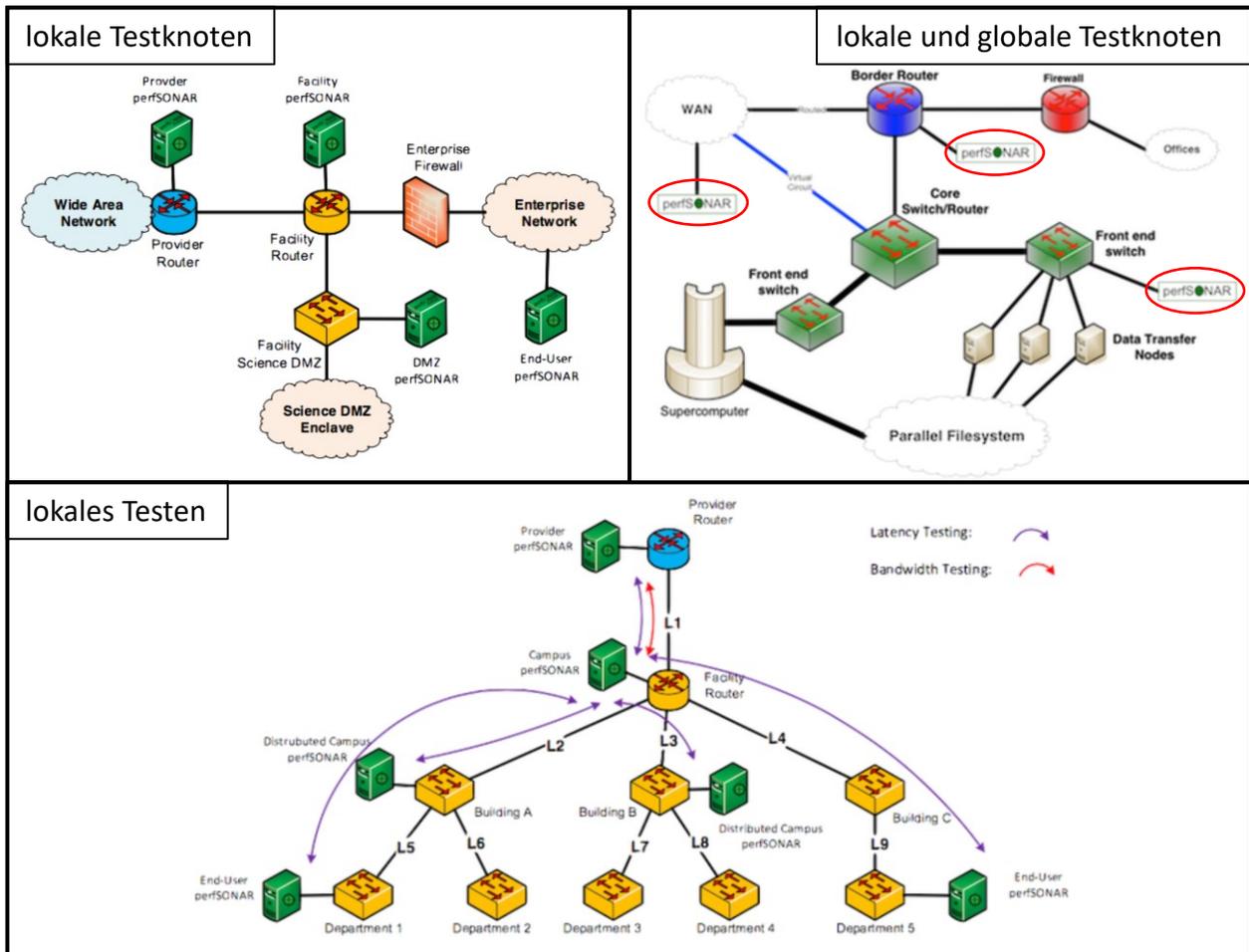


Abb. 76: Platzierung von Testknoten und lokales Testen,
 Quelle: https://www.perfsonar.net/deployment_local.html, https://www.perfsonar.net/deployment_local.html [Stand: 17.11.2022]
 (leicht modifiziert).

Nach Abschluss der Installation des Betriebssystems und der Konfiguration des Geräts können die Installation des perfSONAR-Toolkits und die Startkonfiguration mit Benutzerdefinition durchgeführt werden.⁸⁸ Eines der Testgeräte wird für die Testkonfiguration, Archivierung und Visualisierung herangezogen und unterscheidet sich in diesen Punkten von den anderen Geräten. Daher ist dieser Zeitpunkt günstig, um ein Image vom USB-Stick zu erstellen und damit die Anzahl der Testgeräte zu

⁸⁸ Vgl. [perfsonar/install_centos](#) (2022), Online-Quelle [16.11.2022], Vgl. [perfsonar/install_config_first_time](#) (2022), Online-Quelle [16.11.2022], Vgl. Crichigno (2021) S. 5-18.

vervielfältigen. perfSONAR bietet drei Möglichkeiten zur Durchführung von Messungen – tools, pScheduler und pSConfig. Die Parametereinstellung und Ausführung erfolgt bei tools und pScheduler im Terminal, für pSConfig wird ein JSON-File erstellt oder alternativ das Web-User-Interface verwendet. Zur Veranschaulichung werden nachfolgend die Ergebnisse einer TWAMP-Messung mit den unterschiedlichen Methoden gezeigt.

Tools

Tools eignet sich für einzelne Messungen. Wie in Abb. 77 zu sehen ist, werden eine Vielzahl an Parametern zurückgegeben, wie Anzahl der Pakete, Paketverlust, *Round-Trip Time* (RTT), Sendezeit, Antwortzeit, Jitter und Hops.

```
[Centos7USB@centos7 ~]$ twping 192.168.1.14
Approximately 13.0 seconds until results available

--- twping statistics from [centos7.lan]:9904 to [192.168.1.14]:18823 ---
SID:      c0a8010ee71e0da7ef442a4071031aa8
first:    2022-11-15T14:07:53.173
last:     2022-11-15T14:08:03.013
100 sent, 0 lost (0.000%), 0 send duplicates, 0 reflect duplicates
round-trip time min/median/max = 2.06/3.5/104 ms, (err=5.39 ms)
send time min/median/max = 1.35/1.8/102 ms, (err=2.69 ms)
reflect time min/median/max = 0.659/1.8/6.12 ms, (err=2.69 ms)
reflector processing time min/max = 0.00286/0.0429 ms
two-way jitter = 3.5 ms (P95-P50)
send jitter = 2.3 ms (P95-P50)
reflect jitter = 0.2 ms (P95-P50)
send hops = 0 (consistently)
reflect hops = 0 (consistently)
```

Abb. 77: TWAMP-Latenzzeit-Messung mittels perfSONAR-Tools im CentOS 7-Terminal, Quelle: Eigene Darstellung.

pScheduler

Mittels pScheduler lassen sich einzelne wie auch zeitlich wiederholte Messungen konfigurieren, mit Archivierung oder JSON-Export der Ergebnisse. TWAMP wird von den Testklassen Latency-Test und RTT-Test unterstützt. Diese weisen eine voneinander abweichende Parametrierung auf.⁸⁹ Wie in Abb. 78 gezeigt, enthält die Rückgabe einige zusätzliche Inhalte, die bei einer Messung mittels perfSONAR-Tools nicht abgebildet werden. Die Rückgabe einer RTT-Messung mit TWAMP als definiertes Protokoll ist in Abb. 79 dargestellt.

⁸⁹ Vgl. [perfsonar/pscheduler_ref_tests_tools](#) (2022), Online-Quelle [17.11.2022].

```
[Centos7USB@centos7 ~]$ pscheduler task --tool twping latency --source
192.168.1.180 --dest 192.168.1.14 --ip-version 4
Submitting task...
Task URL:
https://192.168.1.180/pscheduler/tasks/e4b3f491-3eef-4bab-aff0-
90f21377d351
Running with tool 'twping'
Fetching first run...

Next scheduled run:
https://192.168.1.180/pscheduler/tasks/e4b3f491-3eef-4bab-aff0-
90f21377d351/runs/d3bbce04-1ca7-437c-97cb-9c2c3ae6abd8
Starts 2022-11-15T14:26:17+01 (~2 seconds)
Ends 2022-11-15T14:26:38+01 (~20 seconds)
Waiting for result...
```

<pre>Packet Statistics ----- Packets Sent 100 packets Packets Received 100 packets Packets Lost 0 packets Packets Duplicated ... 0 packets Packets Reordered ... 0 packets One-way Latency Statistics ----- Delay Median 2.90 ms Delay Minimum 1.94 ms Delay Maximum 4.80 ms Delay Mean 2.93 ms Delay Mode 2.95 ms Delay 25th Percentile ... 2.75 ms Delay 75th Percentile ... 3.02 ms Delay 95th Percentile ... 3.36 ms Max Clock Error 2.62 ms Common Jitter Measurements: P95 - P50 0.46 ms P75 - P25 0.27 ms Variance 0.12 ms Std Deviation ... 0.35 ms</pre>	<pre>2.84 ms: 2 packets 2.85 ms: 2 packets 2.86 ms: 2 packets 2.87 ms: 1 packets 2.88 ms: 2 packets 2.89 ms: 1 packets 2.90 ms: 1 packets 2.92 ms: 1 packets 2.93 ms: 2 packets 2.94 ms: 2 packets 2.95 ms: 6 packets 2.96 ms: 3 packets 2.97 ms: 4 packets 2.98 ms: 4 packets 3.00 ms: 1 packets 3.01 ms: 1 packets 3.03 ms: 1 packets 3.04 ms: 1 packets 3.05 ms: 2 packets 3.07 ms: 2 packets 3.08 ms: 4 packets 3.09 ms: 1 packets 3.10 ms: 1 packets 3.11 ms: 2 packets 3.15 ms: 1 packets 3.23 ms: 1 packets 3.27 ms: 1 packets 3.28 ms: 1 packets 3.29 ms: 1 packets 3.32 ms: 1 packets 3.36 ms: 1 packets 4.04 ms: 1 packets 4.05 ms: 2 packets 4.80 ms: 1 packets</pre>
<pre>Histogram: 1.94 ms: 1 packets 2.15 ms: 1 packets 2.52 ms: 1 packets 2.56 ms: 1 packets 2.59 ms: 2 packets 2.60 ms: 1 packets 2.62 ms: 1 packets 2.66 ms: 1 packets 2.69 ms: 3 packets 2.70 ms: 3 packets 2.71 ms: 1 packets 2.72 ms: 2 packets 2.73 ms: 2 packets 2.74 ms: 2 packets 2.75 ms: 3 packets 2.76 ms: 4 packets 2.77 ms: 1 packets 2.78 ms: 2 packets 2.79 ms: 4 packets 2.80 ms: 3 packets 2.83 ms: 1 packets</pre>	<pre>TTL Statistics ----- TTL Median 255.00 TTL Minimum 255.00 TTL Maximum 255.00 TTL Mean 255.00 TTL Mode 255.00 TTL 25th Percentile ... 255.00 TTL 75th Percentile ... 255.00 TTL 95th Percentile ... 255.00 Histogram: 255: 100 packets</pre>

Abb. 78: TWAMP-Latenzzeit-Messung mittels perfSONAR-pScheduler im CentOS 7-Terminal, Quelle: Eigene Darstellung.

```
[Centos7USB@centos7 ~]$ pscheduler task rtt --protocol twamp --source
192.168.1.180 --dest 192.168.1.14 --ip-version 4
Submitting task...
Task URL:
https://192.168.1.180/pscheduler/tasks/fbc114a6-f236-42d9-a9d3-
e73aebd60ae
Running with tool 'twping'
Fetching first run...

Next scheduled run:
https://192.168.1.180/pscheduler/tasks/fbc114a6-f236-42d9-a9d3-
e73aebd60ae/runs/bb73e6d0-4f3d-4fb6-940f-b24800242791
Starts 2022-11-15T14:29:04+01 (~2 seconds)
Ends 2022-11-15T14:29:25+01 (~20 seconds)
Waiting for result...

1 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 5.2040 ms
2 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 4.7670 ms
3 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 4.7800 ms
4 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 4.8040 ms
5 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 4.8640 ms
.
.
95 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 5.3150 ms
96 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 5.1200 ms
97 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 5.1310 ms
98 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 5.1440 ms
99 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 6.0460 ms
100 centos7.lan (192.168.1.14) 41 Bytes TTL 255 RTT 6.5130 ms

0% Packet Loss RTT Min/Mean/Max/StdDev =
2.847000/4.917000/6.513000/0.464000 ms
```

Abb. 79: TWAMP-RTT-Messung mittels perfSONAR-pScheduler im CentOS 7 Terminal, Quelle: Eigene Darstellung.

pSConfig

In einem pSConfig-File lassen sich eine beliebige Anzahl an Messungen und Geräten definieren. Um es auszuführen, muss es von einem Teilnehmer veröffentlicht werden und anschließend müssen alle beteiligten Geräte hinzugefügt werden. Das pSConfig-File wird vom pSConfig-pScheduler-Agenten in Tasks zerlegt, die in weiterer Folge vom pScheduler abgearbeitet werden. Eine schematische Darstellung ist in Abb. 80 ersichtlich.

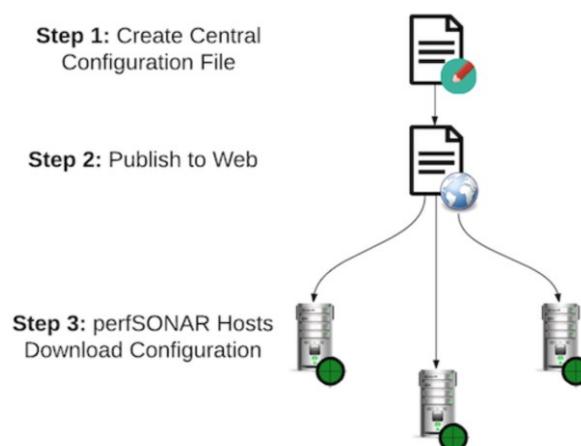


Abb. 80: perfSONAR-pSConfig-Workflow, Quelle: https://docs.perfsonar.net/psconfig_intro.html [Stand: 17.11.2022].

Im Pfad `/usr/share/doc/perfsonar/psconfig` befinden sich einige pSConfig-Templates, die zur Verständnisbildung dienen. Die Struktur gliedert sich nach folgendem Schema: Namen der Messung, Archive, Adressen, Gruppen, Tests, Zeitplan (Schedule) und Tasks. Die Kategorie Archive definiert die Orte, an denen Messdaten gespeichert werden sollen. Die Adressen beschreiben die teilnehmenden Geräte und die Gruppen beschreiben, wie diese miteinander vermessen werden sollen. Das Verhalten der verschiedenen Gruppentypen kann Abb. 81 entnommen werden. Im Bereich Tests werden die Parameter definiert und der Zeitplan definiert das Zeitintervall zwischen den Messungen. Abschließend werden alle vorherigen Einstellungen in Tasks zusammengefasst, also das z. B. Gruppe A mit Test B, Zeitintervall C und Tool XY ausgeführt und in Archive Z abzulegen ist. In Abb. 82 ist ein Ablaufschema der Ableitung von Tasks aus dem pSConfig-File gezeigt, die anschließend im pScheduler ausgeführt werden.⁹⁰

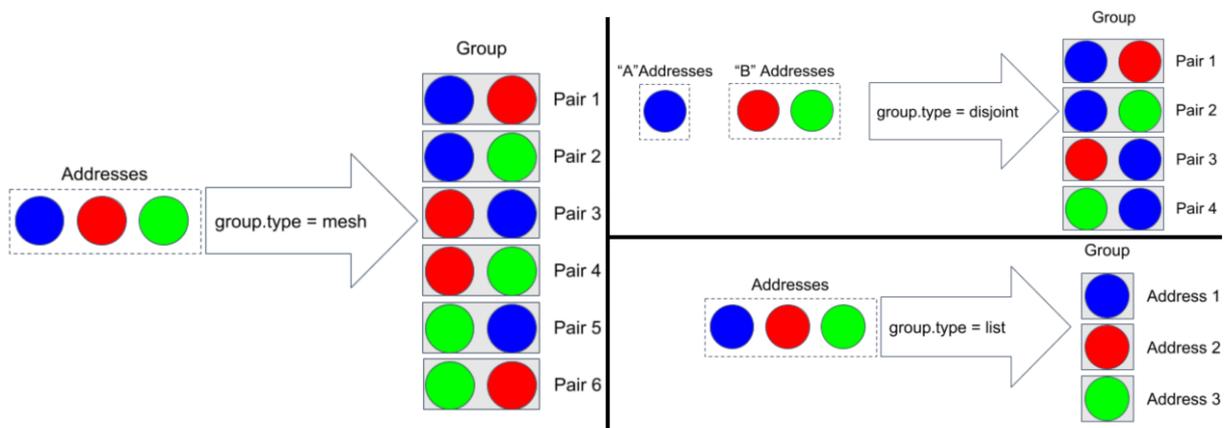


Abb. 81: perfSONAR-pSConfig Gruppentypen, Quelle: https://docs.perfsonar.net/psconfig_templates_intro.html [Stand: 17.11.2022] (leicht modifiziert).

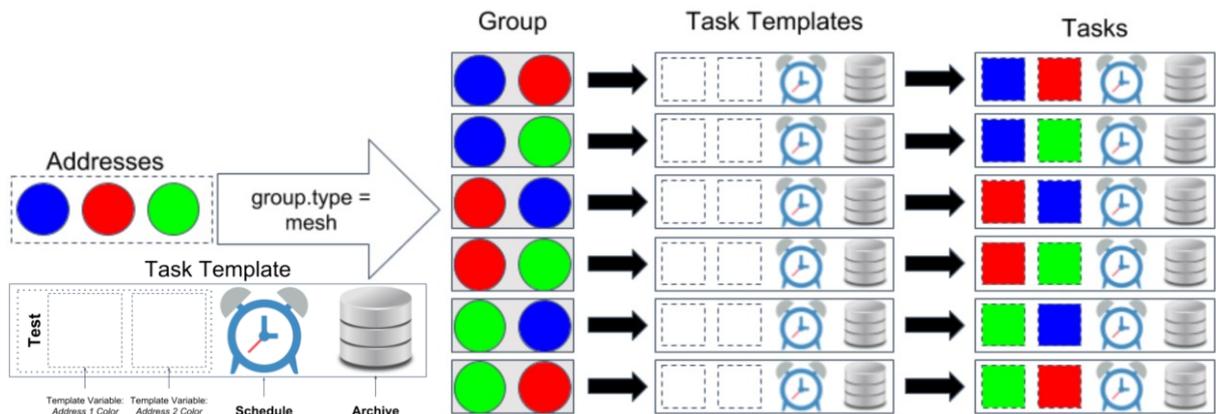


Abb. 82: perfSONAR-Schema der Ableitung von Tasks aus dem pSConfig-File, Quelle: https://docs.perfsonar.net/psconfig_templates_intro.html [Stand: 17.11.2022] (leicht modifiziert).

Bei der Konfiguration einer TWAMP-Latenzzeitmessung ist, wie in Abb. 83 dargestellt, darauf zu achten, dass bei unverschlüsselter Übertragung mindestens 27 Byte und mit Verschlüsselung mindestens 56 Byte angehängt werden müssen, um eine symmetrische Übertragung zu erhalten.⁹¹

⁹⁰ Vgl. [perfsonar/psconfig_templates_intro](https://docs.perfsonar.net/psconfig_templates_intro.html) (2022), Online-Quelle [17.11.2022].

⁹¹ Vgl. A. Morton (2010), Online-Quelle [17.11.2022].

Octets in:	Unauthenticated Mode	Auth/Encrypted Mode
Reflector Header	41	104
Sender Header	14	48
Truncated Padding	27	56

Abb. 83: TWAMP-Paket-Padding für symmetrische Übertragung, Quelle: <https://datatracker.ietf.org/doc/rfc6038/> [Stand: 17.11.2022].

Ist die Konfiguration abgeschlossen, kann diese, wie in Quellcode 32 beschrieben, auf Syntaxfehler geprüft und veröffentlicht werden. Anschließend ist, wie in Quellcode 33 gezeigt, an allen Testgeräten das pSConfig-File zu beziehen, um die Messserie zu starten.⁹²

```
[Cnetos7USB@centos7 ~]$ jq . /home/Centos7USB/Schreibtisch/psconfig-testing-files/latency-test-Wifi-01.json

[Centos7USB@centos7 ~]$ sudo psconfig publish
/home/Centos7USB/Schreibtisch/psconfig-testing-files/latency-test-Wifi-01.json
```

Quellcode 32: Prüfen des JSON-Files und anschließende Veröffentlichung im CentOS 7-Terminal, Quelle: Eigene Darstellung.

```
[Cnetos7USB@centos7 ~]$ sudo psconfig remote add
"https://192.168.1.180/psconfig/latency-test-Wifi-01.json"
```

Quellcode 33: Beziehen des pSConfig-Files im CentOS 7-Terminal, Quelle: Eigene Darstellung.

Mittels Quellcode 34 können die konfigurierten Tasks aufgerufen und deren Ausführung nachverfolgt werden.⁹³

```
[Cnetos7USB@centos7 ~]$ pscheduler monitor
```

Quellcode 34: Aufrufen aller geplanten Tasks des pScheduler im CentOS 7-Terminal, Quelle: Eigene Darstellung.

MaDDash-Visualisierung

Um die Visualisierung in MaDDash nutzen zu können, müssen vorab in den Firewall-Regeln die Ports 80 und 443 freigeschaltet werden.

⁹² Vgl. [perfonar/psconfig_publish](#) (2022), Online-Quelle [17.11.2022], Vgl. [perfonar/psconfig_pscheduler_agent](#) (2022), Online-Quelle [17.11.2022].

⁹³ Vgl. [perfonar/pscheduler_client_schedule](#) (2022), Online-Quelle [17.11.2022].

Tool or Service	TCP	UDP
ICMP		
iperf2	5001	
iperf3	5201	
Lookup Service	8090	
Management Interfaces	80, 443	
Nuttcp	5000, 5101	
NTP	123	
OWAMP (control)	861	
OWAMP (testing)	8760-9960	8760-9960
Simplestream	5890-5900	
Traceroute		33434-33634
TWAMP (control)	862	
TWAMP (testing)	18760-19960	18760-19960

Abb. 84: perfSONAR-Firewall-Regeln,
Quelle: https://www.perfsonar.net/deployment_security.html [Stand: 17.11.2022].

Für das Ablegen der Messdaten im Archiv ist es erforderlich, den entsprechenden Geräten Schreibrechte zu geben.⁹⁴

```
[Cnetos7USB@centos7 ~]$ sudo /usr/sbin/esmond_manage add_user_ip_address
Centos7USB 192.168.1.14
```

Quellcode 35: Testgeräte Schreibrechte für Archive erteilen im CentOS 7-Terminal, Quelle: Eigene Darstellung.

Auf dem Gerät, auf dem die Visualisierung stattfindet, wird der Quellcode 36 ausgeführt, um die Datenbank und das Webinterface zu starten.⁹⁵

```
[Cnetos7USB@centos7 ~]$ sudo systemctl restart cassandra
[Cnetos7USB@centos7 ~]$ sudo systemctl restart maddash-server
[Cnetos7USB@centos7 ~]$ sudo systemctl restart psconfig-maddash-agent
```

Quellcode 36: Neustart von Datenbank und MaDDash im CentOS 7-Terminal, Quelle: Eigene Darstellung.

Die Archiv-Konfiguration im pSConfig-File wird standardmäßig bei remote add ignoriert. Damit eine Datenübermittlung zum Zielarchiv durchgeführt werden kann, muss die Archiv-Konfiguration daher mittels Quellcode 37 gesondert angestoßen werden.⁹⁶

```
[Cnetos7USB@centos7 ~]$ sudo psconfig remote add --configure-
archives "https://192.168.1.180/psconfig/latency-test-wifi-01.json"
```

Quellcode 37: Beziehen des pSConfig-Files im CentOS 7-Terminal, Quelle: Eigene Darstellung.

⁹⁴ Vgl. perfsonar/multi_ma_install (2022), Online-Quelle [17.11.2022].

⁹⁵ Vgl. Crichigno (2021), S. 140.

⁹⁶ Vgl. perfsonar/psconfig_pscheduler_agent (2022), Online-Quelle [17.11.2022].

Nach der abgeschlossenen Konfiguration kann mithilfe eines Browsers das Webinterface mittels <https://192.168.1.180/maddash-webui/> oder <https://HOSTNAME/maddash-webui/> aufgerufen werden. Im Dashboard können die angelegten Messungen und deren Status eingesehen werden. In der Detailansicht kann der zeitliche Messverlauf zwischen zwei Testknoten als Graph ausgegeben werden, wie in Abb. 85 gezeigt.⁹⁷

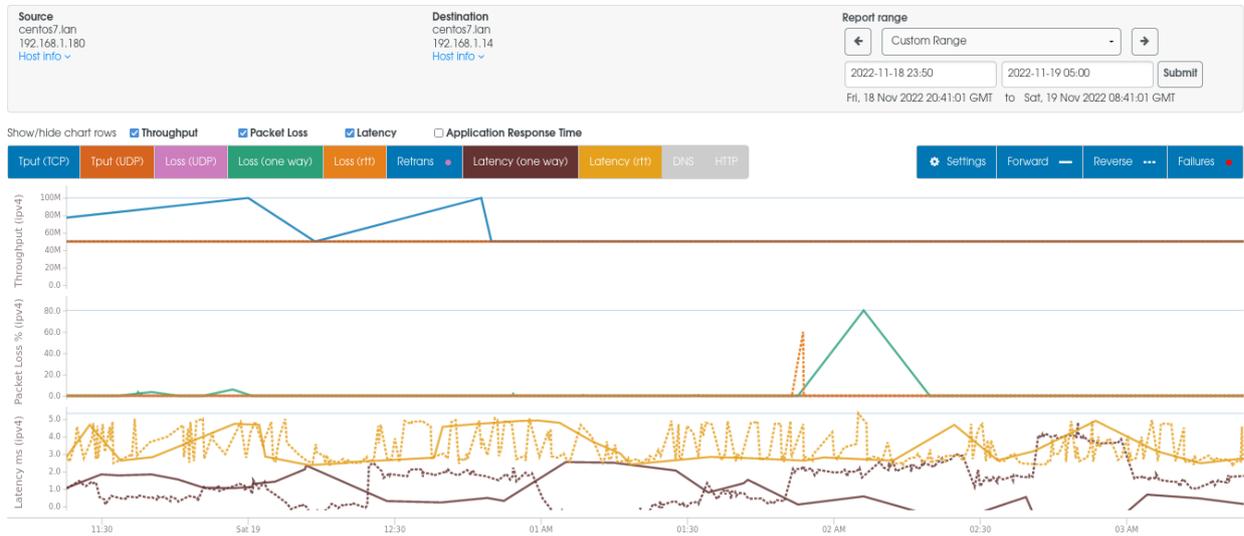


Abb. 85: perfSONAR-MaDDash-Graph eines Messverlaufs, Quelle: eigene Darstellung

⁹⁷ Vgl. Crichigno (2021), S. 142-147.

10 ERGEBNISSE UND AUSBLICK

In diesem Kapitel werden alle erzielten Ergebnisse bezüglich der Videoübertragung, der VR-Headsets und der Netzwerkmesung zusammengefasst. Weiters wird ein Ausblick auf die Weiterentwicklung der 5G-Infrastruktur der FH CAMPUS 02 gegeben. Das Kapitel und die Masterarbeit werden mit einem Fazit abgeschlossen.

10.1 Ergebnisse Videoübertragung

Im Zuge dieser Arbeit konnten die Videoübertragung und -wiedergabe mit zwei Kameras im Webbrowser via WebRTC umgesetzt werden. Das UV4L-Paket uv4l-server ermöglichte es, pro Gerät einen Stream mit der dazugehörigen UV4L-Instanz zu starten und zu übermitteln. In Abb. 86 ist die Übertragung beider Kameras über die Ports 8090 und 9000 anhand zweier Tabellen dargestellt. Wie ersichtlich ist, erfolgt jedoch nur eine Wiedergabe der zuerst gestarteten Übertragung. Dies geschieht ungeachtet des Umstands, dass, wie im Task-Manager ausgegeben, beide Verbindungen aufgebaut bzw. beide Prozesse ausgeführt und vollständig geladen werden. Das hat zur Folge, dass, wie in Abb. 87 erkennbar, die zweite Wiedergabe unmittelbar in jenem Augenblick gestartet wird, in dem die erste Übertragung beendet wird. Zur Ursachenfindung wurden Tests mit unterschiedlichen Browsern, getrennt in unterschiedlichen Browsern und unterschiedlichen Rechnern mit unterschiedlichen Browsern durchgeführt. Jegliche Kombination lieferte dasselbe Ergebnis, ohne einen weiteren Erkenntnisgewinn zu generieren. Die Versuche der Lösungsfindung durch Studium der Konfigurationsparameter und Kontaktaufnahme mit linux-projects.org blieben erfolglos. So konnte nicht eruiert werden, ob eine tiefergehende Konfiguration für eine zweite Videoübertragung bzw. -wiedergabe notwendig ist oder ob diese Anwendung im uv4l-uvc-Treiber anders als beim uv4l-raspicam-Treiber nicht unterstützt wird. Abschließend ist zu erwähnen, dass der Datenkanal mit dem implementierten *Response-Server* auf seine Funktion getestet werden konnte.

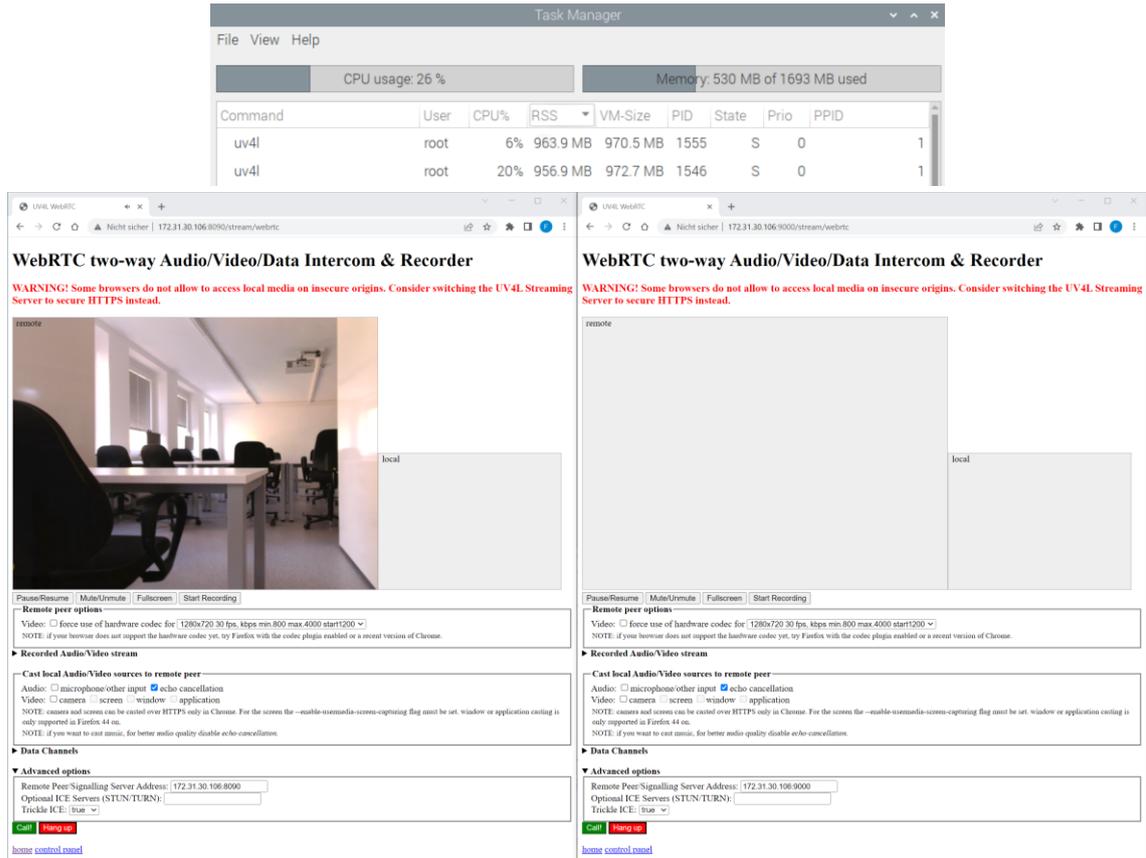


Abb. 86: Stereoskopische Videoübertragung mit Wiedergabe der linken Kamera, Quelle: Eigene Darstellung.

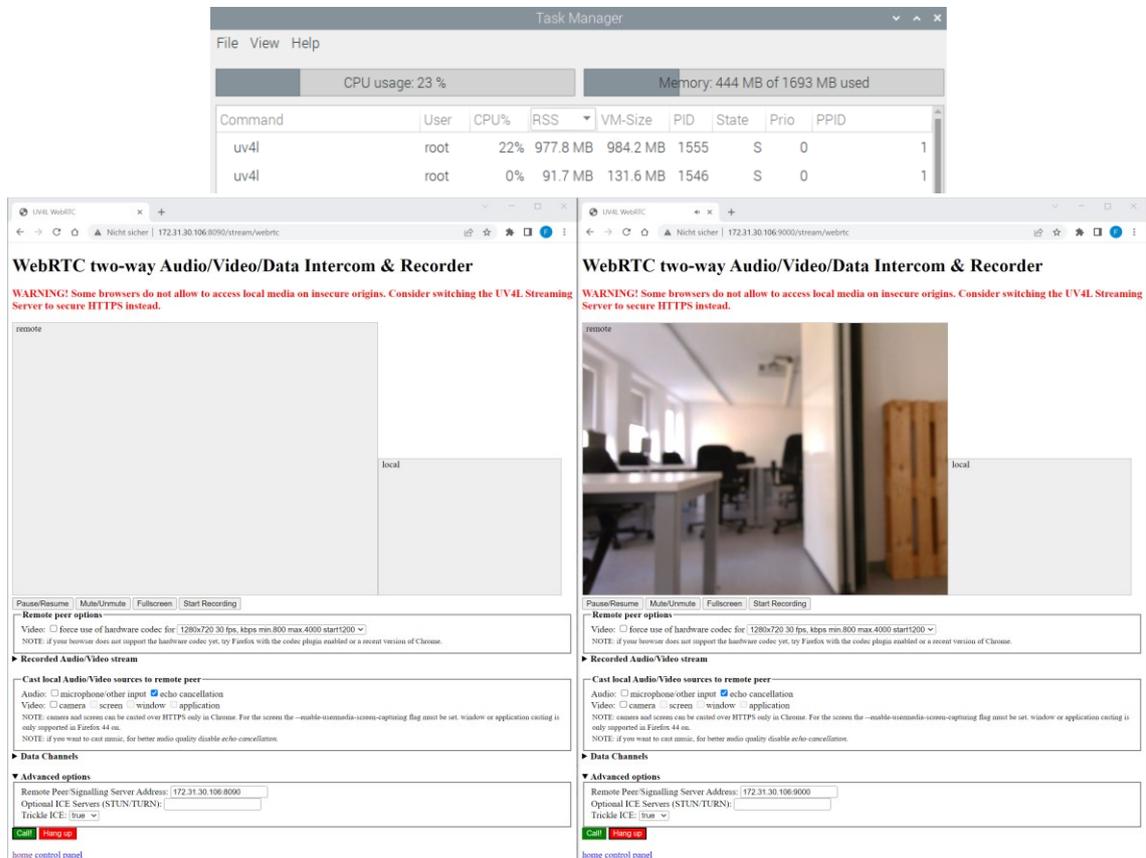


Abb. 87: Stereoskopische Videoübertragung mit Wiedergabe der rechten Kamera, Quelle: Eigene Darstellung.

10.2 Ergebnisse VR-Headset

Die Umsetzung konnte, wie in Kapitel 9.5 beschrieben, im Rahmen dieser Arbeit aufgrund der zeitlichen Beschränkung nicht erfolgen. Ungeachtet dessen wurden alle nötigen Informationen recherchiert und ein Durchführungsplan erstellt. Es zeigte sich jedoch, dass das Wissen zur Umsetzung, gepaart mit den erforderlichen Programmierfähigkeiten für eine Realisierung in Unity nicht ausreichend sind. Vielmehr ist eine umfassende Kenntnis im Umgang mit der Game-Engine und deren Fähigkeiten Voraussetzung, damit der Anwendungsfall in Unity verwirklicht bzw. die VR-Headset-Funktionalität hergestellt werden kann. Um eine grundlegende Vertrautheit mit dem Programm zu erhalten, werden, wie in Abb. 88 gezeigt, diverse Online-Lernpfade angeboten. Anhand der offiziellen Angaben ergibt sich ein Zeitaufwand von 80–100 Stunden, um Grundkenntnisse über die Entwicklungsumgebung des VR-Developments zu erlangen. Weiters ist zu bedenken, dass die WebRTC-Verbindung über ein User-Interface (UI) mit eingebettetem C#-Skript zur Herstellung aller notwendigen Funktionscharakteristika umzusetzen ist. Das Ausrollen des Unity XR-Plug-in-Framework löste die bisherigen AR- bzw. VR-Funktionen ab, die nur noch bedingt unterstützt werden. Somit ist zum aktuellen Zeitpunkt die Suche nach Informationen auf die offizielle Dokumentation beschränkt, da ein Großteil der Inhalte aus der Community obsolet geworden ist. Abschließend ist zu erwähnen, dass für die Handhabung des VR-Headsets im Hinblick auf Videowiedergabe und Bewegungstracking eine Auseinandersetzung mit Camera.stereoTargetEye und InputTracking erforderlich ist.

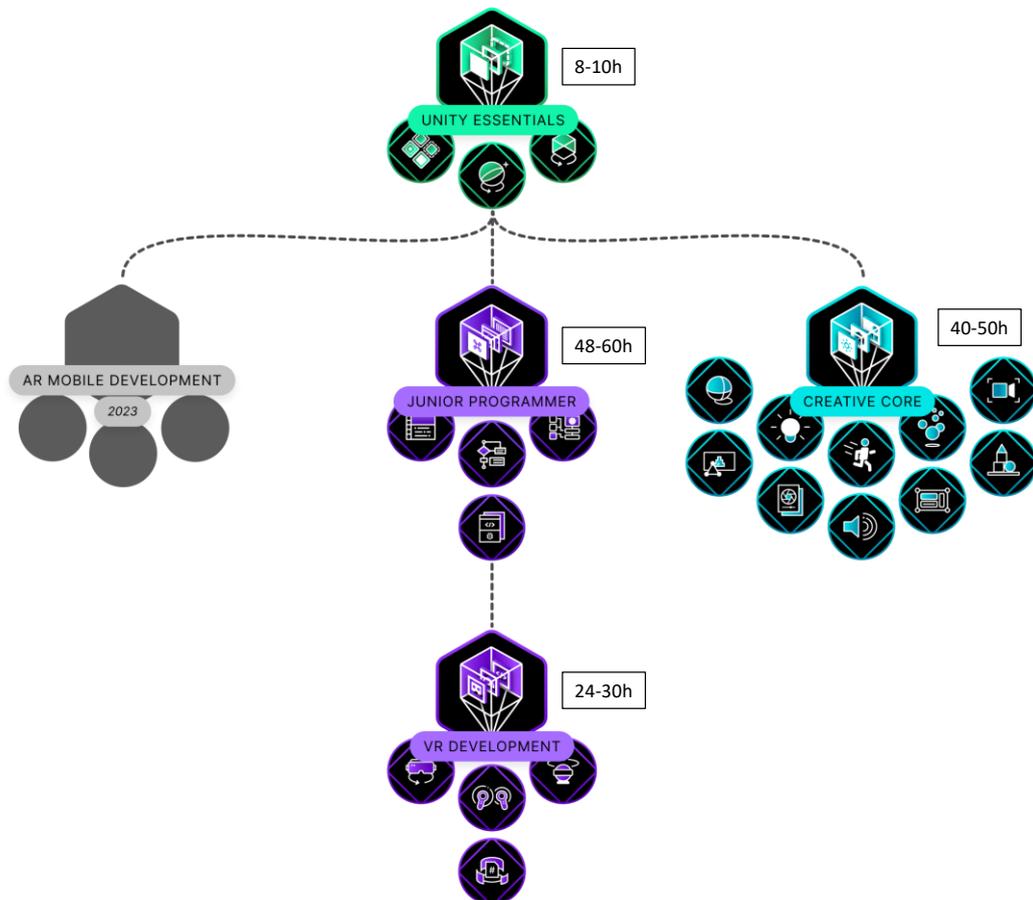


Abb. 88: Unity-Lernpfad, Quelle: <https://learn.unity.com/pathways>, [Stand: 25.11.2022].

10.3 Ergebnisse Netzwerkmessung

Nach Errichten der Testinfrastruktur, wie in Kapitel 9.7 beschrieben, wurde ein pSConfig-File zur Konfiguration von Durchsatz- und Latenzzeitmessungen erstellt. Eine End-to-End-Netzwerkmessung im 5G-Netzwerk war jedoch nicht möglich, da Ping und Traceroute blockiert wurden. Im Laufe des Austausches mit dem Netzbetreiber stellte sich heraus, dass vermutlich die eingesetzten Router die Ursache dafür sind. Ein weiterer Test mit Geräten eines anderen Herstellers konnte zeitlich nicht durchgeführt werden. Als alternativer Lösungsweg wurde das Testsystem in einem lokalen WLAN-Netzwerk (Abb. 89) auf seine Funktion geprüft, da anschließend das Ausrollen in jedes andere Netzwerk durch geringfügige Anpassungen möglich ist.

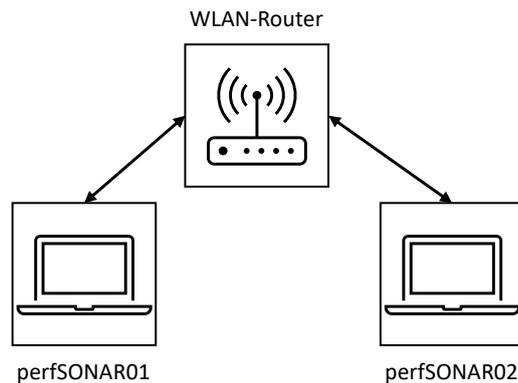


Abb. 89: Messaufbau, Quelle: Eigene Darstellung.

In Abb. 90 ist eine Übersicht über die Tests und deren Parameter dargestellt, wobei die Durchsatzmessungen eine untergeordnete Rolle spielen. Der Fokus lag auf der Ermittlung der Latenzzeit bzw. der *Round-Trip Time*. Die Durchsatzmessung wurde für TCP und UDP weitestgehend ident mit iperf3 und einer Testdauer von zehn Sekunden durchgeführt. Sie wurde in Zeitintervallen von ein bis zwei Stunden wiederholt, da für den einständigen Abstand eine Variation von weiteren 60 Minuten vorgesehen ist. Die Bandbreite wurde auf 50 Mbit/s begrenzt und die maximale TCP-Fenstergröße (Eingangspuffer, Window-Size) mit 1 024 000 Byte festgelegt. Für die Latenzzeitmessung kam TWAMP zum Einsatz. Durch die Parameterwahl wurde die Nachstellung des Testfalles aus Kapitel 6.2 angestrebt. Ein Paketverlust wurde nach zehn Sekunden registriert. Mit einem Sendeintervall von 0,0077 Sekunden, einer Paketanzahl von 1300 und einer Paketfüllung von 1436 Byte zuzüglich des Headers sollte eine Bandbreite von 15 Mbit/s erreicht werden. Das Zeitintervall von zehn Minuten wurde um eine Variation von weiteren zehn Minuten ergänzt. Zudem wurde die *Bucket-Width* für die Histogramm-Auswertung mit 0,001 Sekunden definiert. Die *Round-Trip Time* sollte ursprünglich ebenfalls mit TWAMP ausgeführt werden. Dies konnte, wie in Kapitel 9.7.3 dargestellt, mit dem pScheduler umgesetzt werden. Jedoch war es nicht möglich, die Konfiguration in das pSConfig-File zu übertragen. Als Ursache wird eine mangelnde Integration in den pSConfig-pScheduler-Agent oder ein Konfigurationsfehler vermutet. Daher war es notwendig, auf das Internet Control Message Protocol (ICMP) bzw. Ping auszuweichen. Um die Testzeit bei dieser Messung in einem vertretbaren Rahmen zu halten, wurde die Anzahl mit zehn Wiederholungen festgelegt. Paketlänge und Zeitintervalle wurden aus der Latenzmessung übernommen.

Throughput TCP		Throughput UDP	
Duration	10 s	Duration	10 s
IP-Version	4	IP-Version	4
Bandwidth	50.000.000 bps	Bandwidth	50.000.000 bps
Window-Size	1024000 Byte	Repeat	1H
Repeat	1H	Slip	1H
Slip	1H	Tools	iperf3
Tools	iperf3		
Latency TWAMP		Round Trip Time ICMP	
IP-Version	4	IP-Version	4
Packet-Timeout	10 s	Count	10
Packet-Interval	0,00077 s	Length	1450 byte
Packet-Count	1300	Repeat	10M
Packet-Padding	1436 byte	Slip	10M
Bucket-Width	0,001 s	Tools	ping
Repeat	10M		
Slip	10M		
Tools	twping		

Abb. 90: Parameterkonfiguration der durchgeführten Netzwerktests, Quelle: Eigene Darstellung.

Eine grafische Darstellung der Messergebnisse, der Einzel-Richtungen *Forward* bzw. *Reverse* und beide zusammen *Forward and Reverse*, ist in Abb. 91 ersichtlich. Die Grafik beinhaltet die Resultate der Durchsatzermittlungen von TCP und UDP sowie die Paketverlustrate von UDP. Für die TWAMP-Latenzzeitmessung und die *Ping-Round-Trip Time* wird neben den Latenzzeiten die Paketverlustrate ausgegeben. Die gezeigte Grafik ist nur eine Variante der Ergebnisdarstellung. Da die Archivierung in einer Datenbank erfolgt und verschiedene Möglichkeiten zum Datenexport vorliegen, steht eine Vielzahl an externen Visualisierungs- und Auswertungsoptionen (z. B. Grafana) zur Auswahl. Diese Funktion soll in perfSONAR 5.X optimiert werden, das sich in Entwicklung befindet. Die Hauptplattformen werden hierbei aufgrund der Einstellung der Unterstützung von CentOS durch Red Hat die Open-Source-Betriebssysteme AlmaLinux 8 und RockyLinux 8 sein.



Abb. 91: Messdatenauszug im perfSONAR-MaDDash-Graph, Quelle: Eigene Darstellung.

10.4 Ausblick 5G Infrastruktur

Im Zuge dieser Arbeit wurden gemeinsam mit dem Internet-Service-Provider Citycom Telekommunikation GmbH Varianten für den Aus- und Umbau des 5G-Campus-Netzwerks konzeptionell ausgearbeitet, um ein weiteres Spektrum und mehr Möglichkeiten für zukünftige Forschungsarbeiten zu bieten.

Variante 1: Ist-Stand:

Den SIM-Karten sind statische IP-Adressen zugeordnet. Eine Kommunikation untereinander (Ping, Traceroute, Messungen) ist möglich. Wie in Kapitel 10.3 erwähnt, können sich hierbei die Endgeräte verschieden verhalten und gegebenenfalls den Vorgang unterbinden. Des Weiteren können Port-Forwarding und Firewall-Einstellungen eine Rolle spielen.

Variante 2: Privates Netzwerk / MPLS (Multiprotocol-Label-Switching)

In dieser Variante wird durch MPLS ein eigenes privates Campus-Netzwerk im 5G-Core mit eigenem *Access-Point-Name*(APN)-Zugang und IP-Pool geschaffen. Dadurch sind die Endgeräte untereinander sichtbar und können miteinander kommunizieren. Da dieses Netzwerk gekapselt ist, muss der Internetzugang über einen zusätzlichen Router geregelt werden, der sich im MPLS-Netz befindet.

Variante 3: Zugriff auf das Management der BBU (Baseband Unit):

Hier wird mittels LAN-Link der Management-Port der Campus-BBU auf einen dedizierten Port am Übergabepunkt im CAMPUS 02-Serverschrank aufgeschaltet. Zudem wird ein lokaler User auf der BBU bereitgestellt. Dadurch ist es möglich, Statistiken und Daten direkt auf der Campus-BBU auszulesen.

Die Ausarbeitung der beschriebenen Varianten in ihrer exakten Ausprägung und im konkreten Umfang ist Gegenstand aktueller Aktivitäten. Als Zeithorizont für die Implementierung ist die erste Hälfte des Jahres 2023 anberaumt.

In Abb. 92 ist ein Schema der 5G-Netzwerkinfrastruktur mit ihren logischen Zusammenhängen und dazugehörigen Hardwarestandorten dargestellt. Als Ausblick sind hier bereits die Neuverkabelung der Outdoor-Antenne zur Campus-BBU und die Schaffung eines APN-Zugangs im Serverschrank visualisiert.

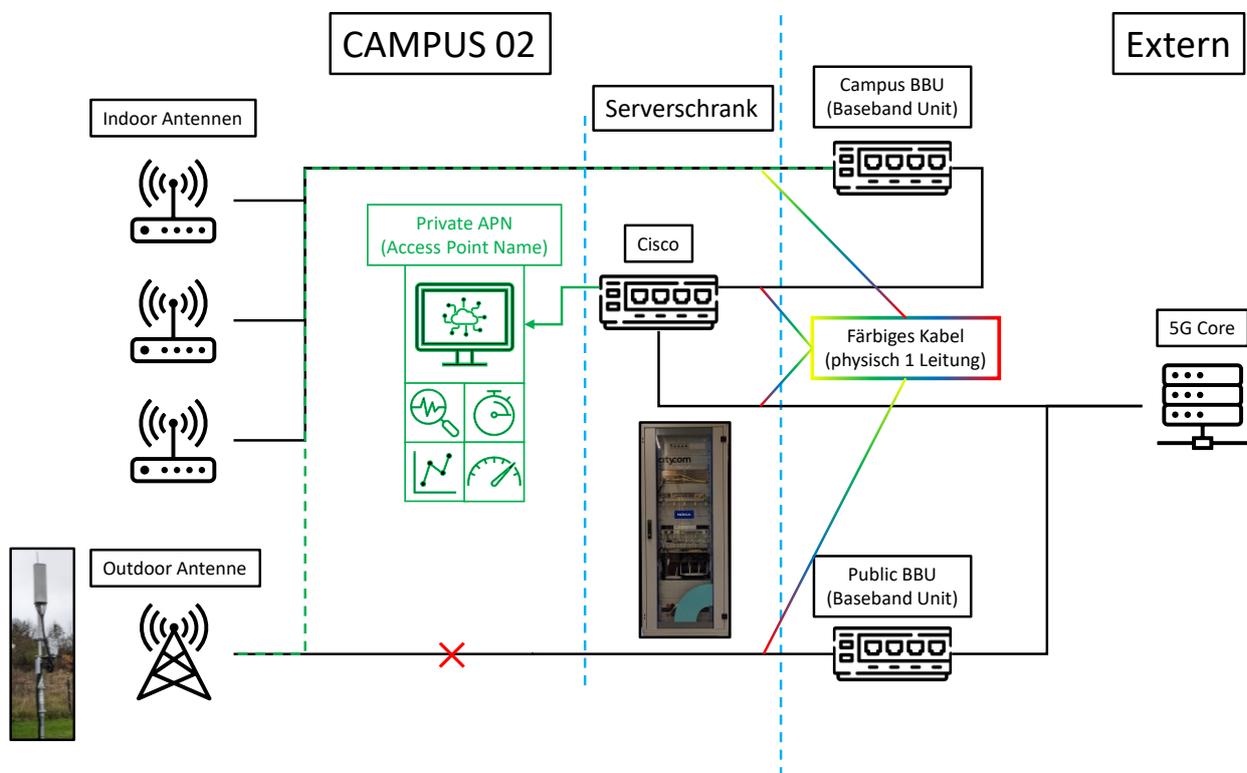


Abb. 92: Schema der FH CAMPUS 02 5G Infrastruktur, Quelle: Eigene Darstellung, Flaticon.com (Cisco, BBU).

Ein hinsichtlich der Latenzzeit idealisiertes Schema für Edge-Computing-Anwendungen und Ähnliches ist ebenfalls in Abb. 93 ersichtlich. Die Einbindung eines dedizierten UPF (User-Plane-Function)-Servers stellt die beste Lösung dar, ist jedoch am aufwändigsten in der Umsetzung.

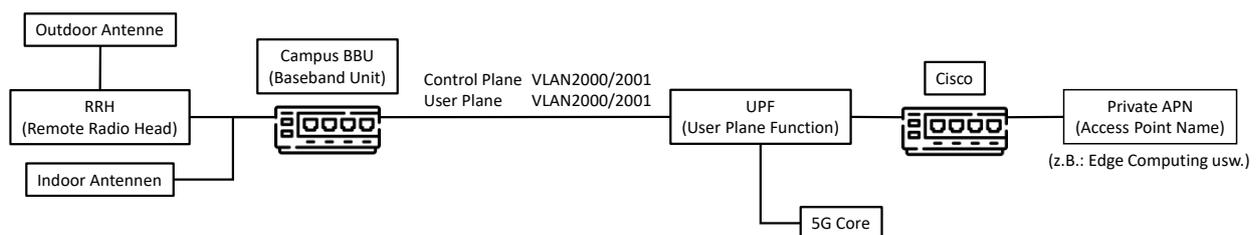


Abb. 93: Schema zur Schaffung eines Zugangs für Edge-Computing-Anwendungen und Ähnliches in der FH CAMPUS 02 5G-Infrastruktur, Quelle: Eigene Darstellung, Flaticon.com (BBU).

10.5 Fazit

Die Forschungsfrage, wie eine Datenübertragung mit niedriger Latenz sowie Latenzzeitmessungen in einem 5G-Netzwerk umgesetzt werden können, wurde erfolgreich beantwortet. Bezüglich der Datenübermittlung stellte sich hierbei WebRTC als das Protokoll heraus, das die geforderten Voraussetzungen für den definierten Versuchsaufbau erfüllt. Durch die bidirektionale Kommunikation und die Bereitstellung eines Video-, Audio- und Datenkanals in Kombination mit niedriger Latenzzeit bietet es großes Potenzial für eine Vielzahl an Einsatzgebieten. Die Latenzzeitmessung konnte durch die Implementierung von perfSONAR, einer Network-Performance-Monitoring-Software, professionell umgesetzt werden. Die während der Ausführung festgestellten Limitationen behinderten lediglich tiefergehende Untersuchungen. Ein vollständiges Ausrollen der erarbeiteten Lösungen ist nach Ausräumen der Hindernisse jederzeit möglich. Für das Einbinden der VR-Brille wurden alle erforderlichen Informationen recherchiert und ein Durchführungsplan konnte erstellt werden. Dessen Ausführung war jedoch aufgrund des Umfangs und der zeitlichen Restriktionen nicht möglich. Zwar wurden eingangs Recherchen zur Umsetzbarkeit durchgeführt, jedoch wurden die Komplexität und der nötige Zeitaufwand maßgeblich unterschätzt. Erschwerend war die Kombination dreier ‚neuen‘ Technologien: 5G-Mobilfunk, WebRTC und VR. Darüber hinaus ist die Art der Verwendung der VR-Brille ein Spezialfall, der in dieser Form nur äußerst selten umgesetzt wird. Zusammengefasst hat die Wahl der VR-Brille als Wiedergabegerät den Aufwand und die Komplexität auf unvorhergesehene Weise erhöht. Abschließend ist zu sagen, dass alle Themengebiete und damit verbundenen Fragen erfolgreich behandelt werden konnten und der Weg für weiterführende Forschung in der 5G-Infrastruktur der FH CAMPUS 02 mit dieser Arbeit geebnet wurde.

LITERATURVERZEICHNIS

Gedruckte Werke (18)

Quectel Wireless Solutions Co., Ltd. (Hrsg.) (2021): *Quectel_RM502Q-AE_5G_Specification_V1.2*

Statista (Hrsg.) (2021): *Mobilfunkstandard-5g*, Statista

3GPP (Hrsg.) (2019): *TR 121 915 - V15.0.0 - Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; Release description; Release 15 (3GPP TR 21.915 version 15.0.0 Release 15)*

3GPP (Hrsg.) (2022): *TS 122 261 - V17.10.0 - 5G; Service requirements for the 5G system (3GPP TS 22.261 version 17.10.0 Release 17)*

GSM Association (Hrsg.) (2021): *Operator Platform Telco Edge Requirements*, Version 1.0 Auflage

3GPP (Hrsg.) (2022): *TS 123 558 V17.5.0 5G; Architecture for enabling Edge Applications (3GPP TS 22.261 version 17.10.0 Release 17)*

5G Alliance for Connected Industries and Automation (Hrsg.) (2021): *5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation*

Brooksby, Jonathan; Murphy, Chris; Featherston, Walter; Thomas, Howard; Kangru, Per; Vaez-Ghaemi, Reza; Wei Koo, Eng; Yamany, Sameh (2021): *Understanding 5G A Practical Guide to Deploying and Operating 5G Networks*, 2. Auflage, VIAVI Solutions

Crichigno, Jorge (2021): *PERFSONAR LAB SERIES*

Dembowski, Klaus (2019): *Raspberry Pi - Das technische Handbuch*, 3., erweiterte und aktualisierte Auflage Auflage, Springer Vieweg, Wiesbaden; Heidelberg

Follmann, Rüdiger (2018): *Das Raspberry Pi Kompendium*, 2., erweiterte und überarbeitete Auflage Auflage, Springer Vieweg, Berlin, Heidelberg

Gordon, Claire; Blackwell, Cynthia; Bradtmiller, Bruce; Parham, Joseph; Barrientos, Patricia; Paquette, Stephen; Corner, Brian; Carson, Jeremy; Venezia, Joseph; Rockwell, Belva; Mucher, Michael; Kristensen, Shirley (2014): *TECHNICAL REPORT NATICK/TR-15/007 2012 ANTHROPOMETRIC SURVEY OF U.S. ARMY PERSONNEL: METHODS AND SUMMARY STATISTICS*, U.S. Army Natick Soldier Research, Development and Engineering Center, Massachusetts

Grigorik, Ilya (2013): *High-performance browser networking*, O'Reilly

Kim, Haesik (2020): *Design and optimization for 5G wireless communications*, Wiley-IEEE Press, Hoboken, NJ, USA

Schnabel, Patrick (2020): *Grundlagen der Computertechnik, Prozessortechnik, Halbleiterspeicher, Schnittstellen, Datenspeicher und Komponenten*, 5. überarbeitete Auflage Auflage, Ludwigsburg

Sprecher, Nurit; Martin, Jesus; Elloumi, Omar; Shen, Jane; Kuure, Pekka; Feng, Jiangping; Neal, Adrian; Garcia, Juan; Soloway, Alan; Yamamoto, Kenichi; Harmand, Alex; Lee, Hyeonsoo; Chitturi, Suresh; Mustapha, Mona; Reznik, Alex; Huang, Qian; Li, Alice; Rossbach, Ralf; Costa, Cristina; Huang, Rong;

Sabella, Dario; Kekki, Sami; Moses, Danny; Mishra, Shamik; Gunjal, Deepak; Kleber, Uli; Hall, Eddy; Featherstone, Walter; Chen, Gao; Yoon, Mingeun; Singh, Himanshu (2020): *ETSI White Paper #36 Harmonizing standards for edge computing - A synergized architecture leveraging ETSI ISG MEC and 3GPP specifications*, 1st edition Auflage, ETSI

Suzuki, Masaki; Joh, Tetsu; Lee, Hyeonsoo; Featherstone, Walter; Sprecher, Nurit; Sabella, Dario; Oliver, Neal; Shailendra, Samar; Granelli, Fabrizio; Costa, Cristina; Chen, Lijuan; Nieminen, Henrik; Berzin, Oleg; Naim, Faraz (2022): *ETSI White Paper No. 49 MEC federation: deployment considerations*, 1st edition Auflage, ETSI

Zhang, Emily (2020): *SIM8200EA-M2_Hardware_Design_V1.03*

Online-Quellen (59)

Waveshare Electronics (2022): *waveshare*

<https://www.waveshare.com/product/iot-communication/long-range-wireless/4g-gsm-gprs/sim8200ea-m2-5g-hat.htm> [Stand: 01.06.2022]

Sixfab, Inc. (2022):

<https://sixfab.com/product/raspberry-pi-5g-development-kit-5g-hat/> [Stand: 15.06.2022]

ServoCity (2022): *ServoCity*

<https://www.servocity.com/spt200-pan-tilt-kit/> [Stand: 15.06.2022]

Lynxmotion (2022): *Lynxmotion*

<http://www.lynxmotion.com/p-287-lynx-b-pan-and-tilt-kit-black-anodized.aspx> [Stand: 15.06.2022]

ARDUCAM TECHNOLOGY CO., LIMITED (2022): *arducam*

<https://www.arducam.com/> [Stand: 17.06.2022]

Shenzhen Ailipu Technology Co.,Ltd (2022): *elpcctv*

<http://www.elpcctv.com/about-us-ezp-14.html> [Stand: 17.06.2022]

Shenzhen Ailipu Technology Co.,Ltd (2022): *elpcctv*

<http://www.elpcctv.com/elp-128-sony-imx415-sensor-night-vision-4k-usb-camera-with-white-led-p-169.html> [Stand: 16.06.2022]

ARDUCAM TECHNOLOGY CO., LIMITED (2022): *uctronics*

<https://www.uctronics.com/download/Amazon/B0280.pdf> [Stand: 15.06.2022]

Semiconductors, N.X.P. (2022): *nxp*

<https://www.nxp.com/docs/en/data-sheet/PCA9685.pdf> [Stand: 19.06.2022]

Pololu Corporation (2022): *pololu*

<https://www.pololu.com/docs/pdf/0J40/maestro.pdf> [Stand: 20.06.2022]

HTC Corporation (2022):

<https://developer.vive.com/resources/hardware-guides/vive-pro-specs-user-guide/> [Stand: 13.07.2022]

Meta (2022):

<https://developer.oculus.com/resources/oculus-device-specs/> [Stand: 13.07.2022]

Anritsu Corporation (2021):

<https://dl.cdn-anritsu.com/en-en/test-measurement/files/Application-Notes/Application-Note/mt1000a-ef7101.pdf> [Stand: 16.09.2022]

linux-projects/uv4l/installation/ (2022): *Installation for ARM*

<https://www.linux-projects.org/uv4l/installation/> [Stand: 19.10.2022]

linux-projects/uv4l/tutorials/custom-webapp-with-face-detection/ (2022): *Example of custom web app*

<https://www.linux-projects.org/uv4l/tutorials/custom-webapp-with-face-detection/> [Stand: 19.10.2022]

dynu/Resources/Tutorials/DynamicDNS/GettingStarted (2022): *dynu Getting started*

<https://www.dynu.com/Resources/Tutorials/DynamicDNS/GettingStarted> [Stand: 22.10.2022]

dynu/DynamicDNS/IPUpdateClient/RaspberryPi-Dynamic-DNS (2022): *dynu Raspberry Pi*

<https://www.dynu.com/DynamicDNS/IPUpdateClient/RaspberryPi-Dynamic-DNS> [Stand: 22.10.2022]

motion-project.github.io (2022):

<https://motion-project.github.io/index.html> [Stand: 16.11.2022]

FFmpeg (2022):

<https://www.ffmpeg.org/> [Stand: 16.11.2022]

perfSONAR/intro_about (2022): *What is perfSONAR?*

https://docs.perfsonar.net/intro_about.html [Stand: 16.11.2022]

perfsonar/install_options (2022): *perfSONAR Installation Options*

https://docs.perfsonar.net/install_options.html [Stand: 16.11.2022]

Red Hat Customer Portal (2022):

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/networking_guide/sec-starting_networkmanager [Stand: 16.11.2022]

ELRepo.org (2022): *wl-kmod Source RPMs for Broadcom drivers*

http://elrepo.org/tiki/wl-kmod#Source_RPMs_for_Broadcom_drivers [Stand: 16.11.2022]

Linuxize (2022):

<https://linuxize.com/post/how-to-mount-an-exfat-drive-on-centos-7/> [Stand: 16.11.2022]

perfsonar/install_centos (2022): *Bundle Installation on CentOS*

https://docs.perfsonar.net/install_centos.html [Stand: 16.11.2022]

perfsonar/install_config_first_time (2022): *Configuring the Toolkit for the First Time*

https://docs.perfsonar.net/install_config_first_time.html [Stand: 16.11.2022]

perfsonar/pscheduler_ref_tests_tools (2022): *Test and Tool Reference*

https://docs.perfsonar.net/pscheduler_ref_tests_tools.html [Stand: 17.11.2022]

perfsonar/psconfig_templates_intro (2022): *Introduction to pSConfig Templates*

https://docs.perfsonar.net/psconfig_templates_intro.html [Stand: 17.11.2022]

perfsonar/psconfig_publish (2022): *Publishing a pSConfig Template*

https://docs.perfsonar.net/psconfig_publish.html [Stand: 17.11.2022]

perfsonar/psconfig_pscheduler_agent (2022): *Running the pSConfig pScheduler Agent*

https://docs.perfsonar.net/psconfig_pscheduler_agent.html [Stand: 17.11.2022]

perfsonar/pscheduler_client_schedule (2022): *Viewing the Schedule*

https://docs.perfsonar.net/pscheduler_client_schedule.html [Stand: 17.11.2022]

perfsonar/multi_ma_install (2022): *Deploying a Central Measurement Archive*

https://docs.perfsonar.net/multi_ma_install.html [Stand: 17.11.2022]

Vuplex3D *WebView for Windows and macOS (Web Browser)*

<https://assetstore.unity.com/packages/tools/gui/3d-webview-for-windows-and-macos-web-browser-154144> [Stand: 24.11.2022]

freedomrobotics *How WebRTC works*

<https://docs.freedomrobotics.ai/docs/how-webrtc-works> [Stand: 24.11.2022]

linux-projects/documentation/uv4l-uvc/*uv4l-uvc*

<https://www.linux-projects.org/documentation/uv4l-uvc/> [Stand: 19.10.2022]

linux-projects/documentation/uv4l-server/*uv4l-server*

<https://www.linux-projects.org/documentation/uv4l-server/> [Stand: 19.10.2022]

linux-projects/documentation/uv4l-core/*uv4l core*

<https://www.linux-projects.org/documentation/uv4l-core/> [Stand: 20.10.2022]

linux-projects/webrtc-signalling/*WebRTC signaling*

<https://www.linux-projects.org/webrtc-signalling/> [Stand: 24.11.2022]

Unity Technologies (2022): *Unity Documentation XR*

<https://docs.unity3d.com/Manual/XR.html> [Stand: 25.11.2022]

linux-projects/uv4l/tutorials/webrtc-data-channels/*WebRTC data channels*

<https://www.linux-projects.org/uv4l/tutorials/webrtc-data-channels/> [Stand: 25.11.2022]

A. Morton, L. (2010): *datatracker.ietf.org*

<https://datatracker.ietf.org/doc/rfc6038/> [Stand: 17.11.2022]

Babiarz, Jozef; Krzanowski, Roman M.; Hedayat, Kaynam; Yum, Kiho; Morton, Al (2020):

datatracker.ietf.org

<https://datatracker.ietf.org/doc/rfc5357/> [Stand: 17.09.2022]

Berger, Jens (2022): *Interactivity test: Impact of changing network conditions on latency and jitter (part 5)*

https://www.rohde-schwarz.com/us/solutions/test-and-measurement/mobile-network-testing/stories-insights/article-interactivity-test-impact-of-changing-network-conditions_255460.html [Stand: 17.09.2022]

Brown, Rory *VRcompare*

<https://vr-compare.com/compare?h1=pDTZ02Pkt&h2=PzyrM6SuMh> [Stand: 26.11.2022]

Chitturi, Suresh (2021): *Enabling Edge Computing Applications in 3GPP*

<https://www.3gpp.org/news-events/3gpp-news/edge-sa6> [Stand: 20.11.2022]

Demirten, Murat (2022): *Twamp-gui*

<http://demirten.github.io/twamp-gui/> [Stand: 16.11.2022]

Earl, Bill (2012): *Adafruit PCA9685 16-Channel Servo Driver*

<https://learn.adafruit.com/16-channel-pwm-servo-driver?view=all#using-the-adafruit-library> [Stand: 25.11.2022]

Kottkamp, Meik; Rohde & Schwarz (2022): *5gtechnologyworld*

<https://www.5gtechnologyworld.com/measurements-show-5g-improves-latency-in-public-networks/> [Stand: 17.09.2022]

Levent-Levi, Tsahi (2020): *WebRTC Signaling Servers: Everything You Need to Know*

<https://www.wowza.com/blog/webrtc-signaling-servers> [Stand: 24.11.2022]

Levent-Levi, Tsahi *WebRTC Glossary JSEP*

<https://bloggeek.me/webrtcglossary/jsep/> [Stand: 24.11.2022]

Levent-Levi, Tsahi (2020): *WebRTC ports: Understanding IP addresses and port ranges in WebRTC*

<https://bloggeek.me/webrtc-ports-ip-addresses/> [Stand: 24.11.2022]

Levent-Levi, Tsahi (2020): *How WebRTC Works?*

<https://bloggeek.me/how-webrtc-works/> [Stand: 24.11.2022]

Nakamaru, Koji *unity-webview*

<https://github.com/gree/unity-webview> [Stand: 24.11.2022]

Ruether, Traci (2022): *Streaming Protocols:*

<https://www.wowza.com/blog/streaming-protocols> [Stand: 17.09.2022]

Ruether, Traci (2022): *RTSP: The Real-Time Streaming Protocol Explained (Update)*

<https://www.wowza.com/blog/rtsp-the-real-time-streaming-protocol-explained> [Stand: 17.09.2022]

Ruether, Traci (2021): *SRT: The Secure Reliable Transport Protocol Explained*

<https://www.wowza.com/blog/srt-the-secure-reliable-transport-protocol-explained> [Stand: 17.09.2022]

Ruether, Traci (2021): *What Is WebRTC? (Update)*

<https://www.wowza.com/blog/what-is-webrtc> [Stand: 17.09.2022]

Sochos, Johanna (2020): *Interactivity test: Concept and KPIs (part 2)*

https://www.rohde-schwarz.com/us/solutions/test-and-measurement/mobile-network-testing/stories-insights/article-interactivity-test-concept-and-kpis-part-2-_253245.html [Stand: 17.09.2022]

Zekauskas, Matthew J.; Karp, Anatoly; Shalunov, Stanislav; Boot, Jeff W.; Teitelbaum, Benjamin R. (2015): *datatracker.ietf.org*

<https://datatracker.ietf.org/doc/rfc4656/> [Stand: 17.09.2022]

ABBILDUNGSVERZEICHNIS

Abb. 1: Entwicklung der Mobilfunktechnologie von 1980er bis 2017, Quelle: Understanding 5G: A Practical Guide to Deploying and Operating 5G Networks (2021), S. xvi.	3
Abb. 2: Prognose zur Anzahl der 5G-Anschlüsse weltweit nach Region von 2020 bis 2027 (in Millionen), Quelle: Statista-Dossier Mobilfunkstandard-5g (2021), S. 10.	4
Abb. 3: Verteilung der Mobilfunkanschlüsse in Europa nach Mobilfunkstandard in den Jahren 2017 bis 2021 und Prognose 2025, Quelle: Statista-Dossier Mobilfunkstandard-5g (2021), S. 13.	4
Abb. 4: 5G Hauptanwendungsszenarien nach International Telecommunication Union (ITU) International Mobile Telecommunications (IMT) Vision, Quelle: Understanding 5G: A Practical Guide to Deploying and Operating 5G Networks (2021), S. 23.	5
Abb. 5: Prinzipdarstellung einer mobilen Netzwerkarchitektur, Quelle: Understanding 5G: A Practical Guide to Deploying and Operating 5G Networks (2021), S. 4.	6
Abb. 6: Non-Stand-Alone-Architektur (NSA), Quelle: 3GPP TR 21.915 version 15.0.0 Release 15 (2019), S 11.	7
Abb. 7: Stand-Alone-Architektur (SA), Quelle: 3GPP TR 21.915 version 15.0.0 Release 15 (2019), S 12. 7	
Abb. 8: Bandbreite (Bandwidth) vs Durchsatz (Throughput), Quelle: Design and optimization for 5G wireless communications (2020), S. 241.	8
Abb. 9: Leistungsanforderungen für Szenarien mit hoher Datenübertragungsgeschwindigkeit und Nutzerdichte, Quelle: 3GPP TS 22.261 version 17.10.0 Release 17 (2022), S. 48.	9
Abb. 10: Gewichtete Schlüsselfähigkeiten in verschiedenen Nutzungsszenarien, Quelle: VIAVI Solutions 5G Architecture and Specification, Primärquelle: ITU-R M.2083-0 (2015), S. 14, 15.	11
Abb. 11: Network-Slicing-Architektur, Quelle: Design and optimization for 5G wireless communications (2020), S. 204.	11
Abb. 12: 3GPP-Architektur zur Realisierung von Edge-Anwendungen, Quelle: https://www.3gpp.org/news-events/3gpp-news/edge-sa6 [Stand: 20.11.2022].	13
Abb. 13: 3GPP-Edge-Architektur, Quelle: 3GPP SA6 EDGEAPP Architecture for enabling Edge Applications (2021), S. 10.	13
Abb. 14: 1. DNS-basierte Erkennung, 2. Edge-Enabler-Client-Erkennung, 3. Edge-Enabler-Client-Erkennung - Management und Orchestrierung auf Grundlage von ETSI ISG MEC, Quelle: ETSI White Paper #36 (2020), S. 11-12, (leicht modifiziert).	14
Abb. 15: Edge Node Sharing, Quelle: GSMA Operator Platform Telco Edge Requirements Version 1.0 (2021), S. 33.	15
Abb. 16: 3GPP <i>Inter-Edge-Data-Network</i> EDGE-9 und <i>Inter-Edge-Data-Network</i> EDGE-9, Quelle: 3GPP TS 23.558 version 17.5.0 Release 17 (2022), S 27.	15

Abb. 17: *Standards Development Organisation (SDO)* übergreifendes *Mapping* der *Operator-Platform(OP)* Architektur Quelle: ETSI White Paper No. 49 (2022), S. 13..... 16

Abb. 18: Gesamtbild eines verbundenen MEC-Systems, Quelle: ETSI White Paper No. 49 (2022), S. 19. 16

Abb. 19: Kennzahlen für Dienste mit hoher Datenübertragungsgeschwindigkeit und niedriger Latenzzeit, Quelle: 3GPP TS 22.261 version 17.10.0 Release 17 (2022), S. 58..... 17

Abb. 20: Betriebsarten des I²C-Bussystems, Quelle: Raspberry Pi - das technische Handbuch (2019), S. 267..... 18

Abb. 21: Übersicht: USB - Universal Serial Bus, Quelle: Computertechnik-Fibel (2020), S. 290-291, (leicht modifiziert). 19

Abb. 22: Transmission Control Protocol (TCP) im Vergleich zum User Datagram Protocol (UDP), Quelle: <https://www.wowza.com/blog/streaming-protocols>, [Stand: 17.09.2022]. 20

Abb. 23: Übersicht von Streaming-Protokollen, Quelle: <https://www.wowza.com/blog/streaming-protocols>, [Stand: 17.09.2022]. 21

Abb. 24: WebRTC(Web Real Time Communication)-Protokoll-Stack, Quelle: High-performance browser networking (2013), S. 317. 22

Abb. 25: Herangehensweisen, um den Netzwerkdurchsatz zu erhöhen, Quelle: Design and optimization for 5G wireless communications (2020), S. 241. 23

Abb. 26: Herangehensweisen, um Latenz zu verringern, Quelle: Design and optimization for 5G wireless communications (2020), S. 305..... 25

Abb. 27: mMTC Designansätze, Quelle: Design and optimization for 5G wireless communications (2020), S. 350. 26

Abb. 28: Durchschnittliche Latenz der Mobilfunkstandards in Österreich vom 1. Quartal 2017 bis zum 4. Quartal 2022, Quelle: <https://de-statista-com.elibrary.campus02.at/statistik/daten/studie/1301773/umfrage/latenz-der-mobilfunkstandards-in-oesterreich/> [Stand: 30.01.2023]. 27

Abb. 29: Durchschnittliche Download-Geschwindigkeiten der Mobilfunkstandards in Österreich vom 1. Quartal 2017 bis zum 4. Quartal 2022, Quelle: <https://de-statista-com.elibrary.campus02.at/statistik/daten/studie/1301770/umfrage/download-rate-nach-mobilfunkstandard-in-oesterreich/> [Stand: 30.01.2023]. 28

Abb. 30: Durchschnittliche Upload-Geschwindigkeiten der Mobilfunkstandards in Österreich vom 1. Quartal 2017 bis zum 4. Quartal 2022, Quelle: <https://de-statista-com.elibrary.campus02.at/statistik/daten/studie/1301762/umfrage/upload-rate-nach-mobilfunkstandard-in-oesterreich/> [Stand: 30.01.2023]. 28

Abb. 31: *Round-Trip Time* (RTT, Paketumlaufzeit), Quelle: <https://dl.cdn-anritsu.com/en-en/test-measurement/files/Application-Notes/Application-Note/mt1000a-ef7101.pdf> [Stand: 16.09.2022]. 29

Abb. 32: *One-Way Latency* (OWL), Quelle: <https://dl.cdn-anritsu.com/en-en/test-measurement/files/Application-Notes/Application-Note/mt1000a-ef7101.pdf> [Stand: 16.09.2022]. 29

Abb. 33: Ping-Schema und Konsolenfenster mit Ping-Befehl und Ausgabe, Quelle: <https://dl.cdn-anritsu.com/en-en/test-measurement/files/Application-Notes/Application-Note/mt1000a-ef7101.pdf> [Stand: 16.09.2022]. 29

Abb. 34: Liste unterschiedlicher Lastparameter, Quelle: <https://www.5gtechnologyworld.com/measurements-show-5g-improves-latency-in-public-networks/> [Stand: 17.09.2022]. 30

Abb. 35: Latenz-Box-Plots pro Technologie für den Vergleich von RTT mit OWL-Ergebnissen für ‚Low‘-, ‚Medium‘- und ‚High‘-Lastparameter, Quelle: <https://www.5gtechnologyworld.com/measurements-show-5g-improves-latency-in-public-networks/>, [Stand: 17.09.2022]. 31

Abb. 36: Übersicht über Leistungs- und Basisparameter, Quelle: 5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation (2021), S. 4-6..... 32

Abb. 37: Testgruppen mit Basisparametervariation, Quelle: 5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation (2021), S. 6. 33

Abb. 38: Performance-Testing Konzept mit Interaktion zwischen dem Testsystem und zu prüfendem System, Quelle: 5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation (2021), S. 7. 33

Abb. 39: Beispielschema einer 5G Industrieanlage, Quelle: 5G-ACIA White Paper Performance Testing of 5G Systems for Industrial Automation (2021), S. 8. 34

Abb. 40: Schema des Versuchsaufbaus, Quelle: Eigene Darstellung. 35

Abb. 41: *Sixfab Raspberry Pi 5G Development Kit*, Quelle: <https://sixfab.com/product/raspberry-pi-5g-development-kit-5g-hat/> [Stand: 19.05.2022]. 37

Abb. 42: *Waveshare SIM8200EA-M2 5G HAT for Raspberry Pi*, Quelle: <https://www.waveshare.com/product/iot-communication/long-range-wireless/4g-gsm-gprs/sim8200ea-m2-5g-hat.htm> [Stand: 24.05.2022]. 37

Abb. 43: ServoCity-SPT200-Direct-Drive-Schwenk-/Neigesystem, Quelle: <https://www.servocity.com/spt200-pan-tilt-kit/> [Stand: 15.06.2022]. 38

Abb. 44: Lynxmotion-Lynx-B-Schwenk-/Neigesystem, Quelle: <http://www.lynxmotion.com/p-287-lynx-b-pan-and-tilt-kit-black-anodized.aspx> [Stand: 15.06.2022]. 38

Abb. 45: ELP-USB4K02AF-KL100W SONY-IMX415-4K-UHD-USB-Kamera mit *Auto Focus*, Quelle: <http://www.elpcctv.com/elp-128-sony-imx415-sensor-night-vision-4k-usb-camera-with-white-led-p-169.html> [Stand: 15.06.2022]. 39

Abb. 46: Arducam-B0280-12MP-IMX477-USB-Kameramodul mit 6 mm-CS-Mount-Objektiv, Quelle: <https://www.uctronics.com/download/Amazon/B0280.pdf> [Stand: 15.06.2022]. 39

Abb. 47: Servomotoren mit PCA9685-Servo-Controller-Board und Raspberry Pi steuern, Quelle: https://tutorials-raspberrypi.de/mehrere-servo-motoren-steuern-raspberry-pi-pca9685/ [Stand: 18.06.2022].	40
Abb. 48: <i>PCA9685 16 Channel 12 Bit PWM Servo Driver</i> , Quelle: http://wiki.sunfounder.cc/index.php?title=PCA9685_16_Channel_12_Bit_PWM_Servo_Driver [Stand: 18.06.2022].	40
Abb. 49: Servomotoren mit Pololu Micro-Maestro-USB-Servo-Controller-Board und Raspberry Pi steuern, Quelle: https://www.aranacorp.com/en/command-a-serial-controller-maestro-with-raspberry-pi/ [Stand: 20.06.2022].	41
Abb. 50: Pololu Micro Maestro USB Servo Controller, Quelle: https://www.pololu.com/product/1350 [Stand: 20.06.2022].	41
Abb. 51: 3-DoF (Rotationsbewegung) und 6-DoF (Rotations- und Translationsbewegung) mit einem VR-Headset, Quelle: https://virtualspeech.com/blog/degrees-of-freedom-vr [Stand: 13.07.2022].	42
Abb. 52: VR-Brille Oculus Quest 2, Quelle: https://business.oculus.com/products/specs/ [Stand: 13.07.2022].	42
Abb. 53: VR-Brille Vive Pro, Quelle: https://www.vive.com/de/product/vive-pro/ [Stand: 13.07.2022].	42
Abb. 54: Zuweisung von GPU-Speicher in der <i>Raspberry Pi Software Configuration</i> , Quelle: Eigene Darstellung.	45
Abb. 55: Optionale Angaben zur Erstellung des privaten Schlüssels und des selbstsignierten Zertifikats, Quelle: Eigene Darstellung.	47
Abb. 56: Ausgabe des <code>lsusb</code> -Befehls im Pi-OS-Terminal, Quelle: Eigene Darstellung.	48
Abb. 57: Registrierung eines USB-Geräts und dazugehörige Rückgabe im Pi-OS-Terminal, Quelle: Eigene Darstellung.	48
Abb. 58: UV4L-Streaming-Server-Webanwendung, Quelle: Eigene Darstellung.	49
Abb. 59: UV4L-WebRTC-Stream-Weboberfläche und gestartete Videoübertragung, Quelle: Eigene Darstellung.	49
Abb. 60: Rückgabe der hinzugefügten Firewall-Regeln im Pi-OS-Terminal, Quelle: Eigene Darstellung.	51
Abb. 61: Aktivierung von UFW mit Rückgabe im Pi-OS-Terminal, Quelle: Eigene Darstellung.	51
Abb. 62: Statusabfrage mit Nummerierung von UFW mit Rückgabe in Pi-OS-Terminal, Quelle: Eigene Darstellung.	52
Abb. 63: Erstellung einer Sub-Domain, Quelle: Eigene Darstellung.	52
Abb. 64: Web-Interface Nokia FastMile 5G Gateway 3.1 Routerkonfiguration, Quelle: Eigene Darstellung, https://www.nokia.com/networks/products/fastmile-5g-gateways/ [Stand: 24.11.2022].	54
Abb. 65: OpenXR Application Interface, Quelle: https://www.khronos.org/openxr/ [Stand: 24.11.2022].	55

Abb. 66: WebRTC-Signaling Server und WebRTC offer/answer Architektur, Quelle:
<https://www.wowza.com/blog/webrtc-signaling-servers>, <https://bloggeek.me/webrtcglossary/jsep/>,
 [Stand: 24.11.2022] (leicht modifiziert)..... 56

Abb. 67: WebRTC-Verbindungsherstellung, Quelle: <https://docs.freedomrobotics.ai/docs/how-webrtc-works>, [Stand: 24.11.2022]..... 57

Abb. 68: Unity-WebRTC-Paket, Quelle: Eigene Darstellung. 58

Abb. 69: Unity-WebRTC-UI, Quelle: Eigene Darstellung. 58

Abb. 70: Unity-XR-Stereo-Rig, Quelle: Eigene Darstellung. 59

Abb. 71: Unity XR-Plugin-Framework-Struktur, Quelle:
<https://docs.unity3d.com/Manual/XRPluginArchitecture.html>, [Stand: 25.11.2022]..... 60

Abb. 72: Verteilung der Interpupillary Distance der männlichen und weiblichen Testpersonen absolut und
 prozentual, Quelle: Eigene Darstellung. 63

Abb. 73: Stereoskopisches Kamerasetup auf motorisiertem Kamerastativ, Quelle: Eigene Darstellung. . 64

Abb. 74: Übersicht perfSONAR-Umfang, Quelle: https://docs.perfsonar.net/intro_about.html
 [Stand: 16.11.2022]. 65

Abb. 75: Netzwerktestendgeräte: 1. Computer zu Raspberry Pi 4, 2. Computer zu Computer, Quelle:
 Eigene Darstellung, Flaticon.com(Raspberry Pi). 65

Abb. 76: Platzierung von Testknoten und lokales Testen, Quelle:
https://www.perfsonar.net/deployment_local.html, https://www.perfsonar.net/deployment_local.html
 [Stand: 17.11.2022] (leicht modifiziert)..... 67

Abb. 77: TWAMP-Latenzzeit-Messung mittels perfSONAR-Tools im CentOS 7-Terminal, Quelle: Eigene
 Darstellung. 68

Abb. 78: TWAMP-Latenzzeit-Messung mittels perfSONAR-pScheduler im CentOS 7-Terminal, Quelle:
 Eigene Darstellung. 69

Abb. 79: TWAMP-RTT-Messung mittels perfSONAR-pScheduler im CentOS 7 Terminal, Quelle: Eigene
 Darstellung. 70

Abb. 80: perfSONAR-pSConfig-Workflow, Quelle: https://docs.perfsonar.net/psconfig_intro.html
 [Stand: 17.11.2022]. 70

Abb. 81: perfSONAR-pSConfig Gruppentypen, Quelle:
https://docs.perfsonar.net/psconfig_templates_intro.html [Stand: 17.11.2022] (leicht modifiziert)..... 71

Abb. 82: perfSONAR-Schema der Ableitung von Tasks aus dem pSConfig-File, Quelle:
https://docs.perfsonar.net/psconfig_templates_intro.html [Stand: 17.11.2022] (leicht modifiziert)..... 71

Abb. 83: TWAMP-Paket-Padding für symmetrische Übertragung, Quelle:
<https://datatracker.ietf.org/doc/rfc6038/> [Stand: 17.11.2022]. 72

Abb. 84: perfSONAR-Firewall-Regeln, Quelle: https://www.perfsonar.net/deployment_security.html [Stand: 17.11.2022]. 73

Abb. 85: perfSONAR-MaDDash-Graph eines Messverlaufs, Quelle: eigene Darstellung 74

Abb. 86: Stereoskopische Videoübertragung mit Wiedergabe der linken Kamera, Quelle: Eigene Darstellung. 76

Abb. 87: Stereoskopische Videoübertragung mit Wiedergabe der rechten Kamera, Quelle: Eigene Darstellung. 76

Abb. 88: Unity-Lernpfad, Quelle: <https://learn.unity.com/pathways>, [Stand: 25.11.2022]..... 77

Abb. 89: Messaufbau, Quelle: Eigene Darstellung. 78

Abb. 90: Parameterkonfiguration der durchgeführten Netzwerktests, Quelle: Eigene Darstellung. 79

Abb. 91: Messdatenauszug im perfSONAR-MaDDash-Graph, Quelle: Eigene Darstellung. 80

Abb. 92: Schema der FH CAMPUS 02 5G Infrastruktur, Quelle: Eigene Darstellung, Flaticon.com (Cisco, BBU). 81

Abb. 93: Schema zur Schaffung eines Zugangs für Edge-Computing-Anwendungen und Ähnliches in der FH CAMPUS 02 5G-Infrastruktur, Quelle: Eigene Darstellung, Flaticon.com (BBU)..... 81

TABELLENVERZEICHNIS

Tab. 1: Datenübertragungsgeschwindigkeit 5G-Modem Quectel M502Q-AE und SIMcom SIM8200EA-M2, Quelle: Eigene Darstellung	36
Tab. 2: Statistische Kennwerte der Interpupillary Distance, Quelle: 2012 ANTHROPOMETRIC SURVEY OF U.S. ARMY PERSONNEL: METHODS AND SUMMARY STATISTICS, S. 149-150 (leicht modifiziert).	61
Tab. 3: Häufigkeitstabelle Interpupillary Distance, Quelle: 2012 ANTHROPOMETRIC SURVEY OF U.S. ARMY PERSONNEL: METHODS AND SUMMARY STATISTICS, S. 150 (leicht modifiziert).	62

QUELLCODEVERZEICHNIS

Quellcode 1: Hinzufügen der externen Paketquelle von UV4L im Pi-OS-Terminal, Quelle: Eigene Darstellung.	44
Quellcode 2: Aktualisierung der Paketliste und Installation neuerer Paketversionen im Pi-OS-Terminal, Quelle: Eigene Darstellung.	44
Quellcode 3: Installation des UV4L-Raspicam-Treibers im Pi-OS-Terminal, Quelle: Eigene Darstellung.	45
Quellcode 4: Installation von UV4L-Raspicam-Extras im Pi-OS-Terminal, Quelle: Eigene Darstellung.	45
Quellcode 5: Aufrufen der raspi-config im Pi-OS-Terminal, Quelle: Eigene Darstellung.	45
Quellcode 6: Installation optionaler UV4L-Pakete im Pi-OS-Terminal, Quelle: Eigene Darstellung.	46
Quellcode 7: Installation des WebRTC-Pakets für Raspberry Pi 2, 3 und 4 im Pi-OS-Terminal, Quelle: Eigene Darstellung.	46
Quellcode 8: Installation des Demo-Pakets, einer externen Webanwendung, im Pi-OS-Terminal, Quelle: Eigene Darstellung.	46
Quellcode 9: Erstellung eines privaten Schlüssels und eines selbstsignierten Zertifikats im Pi-OS-Terminal, Quelle: Eigene Darstellung.	46
Quellcode 10: Installation des XMPP-Pakets im Pi-OS-Terminal, Quelle: Eigene Darstellung.	47
Quellcode 11: Isusb Befehl in Pi OS Terminal, Quelle: Eigene Darstellung.	47
Quellcode 12: Registrierung eines USB-Geräts im Pi-OS-Terminal, Quelle: Eigene Darstellung.	48
Quellcode 13: Befehl zum Aufrufen der Treiberfunktionen und Konfigurationen in Pi OS Terminal, Quelle: Eigene Darstellung.	50
Quellcode 14: Installation von <i>Uncomplicated Firewall</i> (UFW) im Pi-OS-Terminal, Quelle: Eigene Darstellung.	50
Quellcode 15: Port-Freischaltung mit limitierter Verbindungsanzahl UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.	50
Quellcode 16: Port-Freischaltung UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.	51
Quellcode 17: Installation von UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.	51
Quellcode 18: Aktivierung von UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.	51
Quellcode 19: Statusabfrage von UFW im Pi-OS-Terminal, Quelle: Eigene Darstellung.	51
Quellcode 20: Erstellung und anschließendes Öffnen eines neuen Ordners im Pi-OS-Terminal, Quelle: Eigene Darstellung.	53
Quellcode 21: Erstellung und Öffnen eines neuen Dokuments im Pi-OS-Terminal, Quelle: Eigene Darstellung.	53
Quellcode 22: Dokumentinhalt für Auto-IP-Update im Pi OS Terminal, Quelle: Eigene Darstellung.	53

Quellcode 23: Umwandlung des Dokuments in eine ausführbare Datei im Pi-OS-Terminal, Quelle: Eigene Darstellung. 53

Quellcode 24: Öffnen der Cron-Tabelle im Pi-OS-Terminal, Quelle: Eigene Darstellung. 53

Quellcode 25: Hinzufügen des Auto-IP-Update-Skripts zur Cron-Tabelle im Pi-OS-Terminal, Quelle: Eigene Darstellung. 54

Quellcode 26: Call-Request UV4L Signaling-Protokoll, Quelle: Eigene Darstellung. 57

Quellcode 27: Installation der Adafruit-CircuitPython-Servokit Bibliothek im Pi-OS-Terminal, Quelle: Eigene Darstellung. 60

Quellcode 28: I²C-Schnittstellendetektion im Pi-OS-Terminal, Quelle: Eigene Darstellung. 61

Quellcode 29: Netzwerkmanager mit Systemstart ausführen im CentOS 7-Terminal , Quelle: Eigene Darstellung. 66

Quellcode 30: Ermitteln des WLAN-Chipsets im CentOS 7-Terminal, Quelle: Eigene Darstellung. 66

Quellcode 31: Herstellung der Unterstützung von exFAT im CentOS 7-Terminal, Quelle: Eigene Darstellung. 66

Quellcode 32: Prüfen des JSON-Files und anschließende Veröffentlichung im CentOS 7-Terminal, Quelle: Eigene Darstellung..... 72

Quellcode 33: Beziehen des pSConfig-Files im CentOS 7-Terminal, Quelle: Eigene Darstellung..... 72

Quellcode 34: Aufrufen aller geplanten Tasks des pScheduler im CentOS 7-Terminal, Quelle: Eigene Darstellung. 72

Quellcode 35: Testgeräte Schreibrechte für Archive erteilen im CentOS 7-Terminal, Quelle: Eigene Darstellung. 73

Quellcode 36: Neustart von Datenbank und MaDDash im CentOS 7-Terminal, Quelle: Eigene Darstellung. 73

Quellcode 37: Beziehen des pSConfig-Files im CentOS 7-Terminal, Quelle: Eigene Darstellung..... 73