

Schlussbericht vom 31.07.2023

zu IGF-Vorhaben Nr. 21639 N

Thema

Integriertes Testbed für die ubiquitäre, durchgängige, echtzeitfähige und zuverlässige Kommunikation für Industrie 4.0 - TestKomm 4.0

Berichtszeitraum

01.04.2021 - 31.03.2023

Forschungsvereinigung

Forschungsvereinigung Elektrotechnik beim ZVEI e.V.

Forschungseinrichtung(en)

Hahn-Schickard-Gesellschaft für angewandte Forschung e.V. (FE 1)

Hochschule Offenburg (FE 2)

Gefördert durch:

TestKomm 4.0

Integriertes Testbed für die ubiquitäre, durchgängige, echtzeitfähige und zuverlässige Kommunikation für Industrie 4.0

Abschlussbericht

Kurzfassung:

Im Zeitalter des Internet of Things (IoT) sind praktisch alle Anwendungsfelder der Automatisierungstechnik von der zunehmenden Vernetzung betroffen. Vor dem Hintergrund der potenziellen Umsatzmöglichkeiten und auch unter dem Druck konkurrierender Lösungen aus den Bereichen der schmalbandigen Low Power Wide Area-(LPWA)-Netzwerktechnologien unter Nutzung lizenzfreier Frequenzen, wie LoRa, SIGFOX, Weightless oder MIOTY, hat das 3rd Generation Partnership Project (3GPP) als Standardisierungsgremium für die zellulare Mobilkommunikation diesen Markt ebenfalls entdeckt. In der Folge hat 3GPP für die vierte Mobilfunkgeneration (4G) LTE ab der Rel. 13 verschiedene Alternativen entwickelt, deren Bandbreiten und Datenraten gegenüber der Standard-LTE-Variante reduziert sind und die somit eine höhere Nutzerdichte, geringere Kosten und eine geringere Energieaufnahme versprechen. Diese neuen Narrowband-IoT-(NB-IoT)-Varianten sind seit Rel. 15 Teil des 5G Standards als integraler Bestandteil der massiven Maschinenkommunikation (mMTC). Somit stehen sich zunehmend mehrere potentielle Kommunikationstechnologien gegenüber, die sich in Ihren Eigenschaften und deren Nutzbarkeit für bestimmte Anwendungsszenarien deutlich unterscheiden.

In dieser Situation der rapiden Entwicklungen und Verfügbarkeit der Kommunikationstechnologien stellen sich zahlreiche Fragen hinsichtlich deren Nutzbarkeit, die vor allem für die deutsche industrielle Automatisierungstechnik von grundlegender Bedeutung sind. Für die Beantwortung dieser Fragen wurde im Rahmen des TestKomm 4.0-Projekts neben den theoretischen Untersuchungen eine Testumgebung aufgebaut, auf der repräsentative Testfälle reproduzierbar durchgeführt und nachvollziehbar gemessen werden konnten. Diese Testumgebung integriert unterschiedliche Abstraktionsebenen von der Simulation über eine Emulation bis hin zu Feldtests. Im Ergebnis werden sowohl Hersteller- und Technologie neutrale Ergebnisse bereitgestellt als auch eine Testumgebung aufgebaut und bereitgestellt, die es Unternehmen erlaubt, ihre eigenen Produkte zu validieren und zu optimieren.

Das Ziel des Forschungsvorhabens ist erreicht worden.

Berichtsumfang: 153 S., 69 Abb., 46 Tab., 5 List., 75 Lit.

Laufzeit: 01.04.2021 - 31.03.2023

Forschungsstellen:

Hahn-Schickard-Gesellschaft für
angewandte Forschung e.V. - HSG (FE1)
Dr. Mark Weyer

Hochschule Offenburg - HSO (FE2)
Prof. Dr. Axel Sikora

Bearbeiter und Verfasser:

Prof. Dr. Axel Sikora (HSO)
Manuel Schappacher (HSO)
Fabian Sowieja (HSO)
Jubin Elayanithottathil (HSO)

Dr. Mark Weyer (HSG)
Kristian Stern (HSG)
Nikolas Zeitler (HSG)

Inhaltsverzeichnis

Kurzfassung	II
1. Executive Summary	1
2. Einleitung	2
2.1. Problem und Aufgabenstellung	2
2.2. Lösungsansatz	3
2.3. Projektablauf	3
2.4. Zusammenarbeit mit Dritten	4
2.5. Verwendung und Veröffentlichung der Ergebnisse	4
3. Stand der Technik	5
3.1. Wireless Industrial IoT	5
3.2. Low Power Wide Area Networks	6
3.2.1. LoRaWAN	7
3.2.2. MIOTY	9
3.3. Zelluläre Netzwerke	10
3.3.1. 4G - LTE	11
3.3.2. 5G- NR	12
3.3.3. NB-IoT	13
3.4. Vergleich drahtloser Technologien	14
3.4.1. Überblick	14
3.4.2. Leistungsparameter	15
3.5. Tests und Messungen von drahtlosen Technologien	16
3.5.1. Abstraktionsebenen und vorhandene Lösungen	17
3.5.2. Testbeschreibungssprache	19
3.5.3. TTCN-3 Test Automation Frameworks	19
4. Anwendungsfälle	21
4.1. Charakterisierung von drahtlosen Kommunikationsanwendungen	21
4.2. Fabrikautomation	24
4.2.1. FA1 - Panel zur Robotersteuerung	24
4.2.2. FA2 - Produktionszelle	25
4.3. Prozessautomation	26
4.3.1. PA1 - Kläranlage	26
4.3.2. PA2 - Zustandsüberwachung einer Prozessanlage	27
4.4. Maschine-zu-Maschine Kommunikation	28
4.4.1. MM1 - Fahrerlose Transportsysteme	28
4.4.2. MM2 - Warenverfolgung	30
5. Anforderungserhebung	30
5.1. Drahtlose Kommunikationslösungen	31

5.2. Testsysteme	32
5.2.1. Zentrale Teststeuerung	33
5.2.2. Feldtests	35
5.2.3. Emulator	35
5.2.4. Simulator	36
6. Aufbau Simulationsumgebung	37
6.1. Simulationsumgebung	37
6.2. Beschreibungssprache für Simulationen	39
6.3. TestKomm 4.0 Schnittstelle	42
6.4. Frontend für die Ausführung von Simulationen	43
7. Aufbau der Emulations- und Feldtestumgebungen	45
7.1. Zentrales Testsystem	45
7.1.1. Architektur	45
7.1.2. Implementierung der Testfälle	47
7.1.3. Konfigurationsparameter	48
7.1.4. Interaktion mit Testgeräten	50
7.1.5. Anbindung der Simulationsumgebung	51
7.1.6. Anbindung der Emulationsumgebung	52
7.1.7. Interaktion mit Messgeräten	53
7.1.8. Visualisierung und Zusatzskripte	54
7.1.9. Testsystem-Interface	55
7.2. Automated Physical Testbed	57
7.2.1. Hardware Architektur	57
7.2.2. Hardware Aufbau	58
7.2.3. Software-Architektur	58
7.2.4. Steuerung des APTBs	60
7.2.5. Mehrwegeausbreitung und Verzögerungselemente	63
8. Erstellung der Auswertesoftware	67
8.1. Datenbank	67
8.2. Systemarchitektur	69
8.3. Visualisierungsanwendung	69
9. Aufbau und Beschreibung von Testfällen	73
9.1. Testphasen	73
9.2. Beschreibung von Testfällen	73
9.2.1. Testfälle	74
9.2.2. Parametrierung	74
9.3. Testaufbau	75
9.3.1. Eingesetzte Geräte	77
9.3.2. Feldtestumgebung	77
9.3.3. Emulationsumgebung	79

9.3.4. Simulationsumgebung	81
9.4. Testsystematik	83
9.4.1. Technologievergleich	83
9.4.2. Abstraktionsebenen	83
9.4.3. Parametervariationen	84
9.4.4. Optimierung	87
9.5. Testausführung	87
10. Testergebnisse	89
10.1. Technologievergleich	89
10.2. Abstraktionsvergleich	90
10.3. Parametervariationen	92
10.3.1. NB-IoT	92
10.3.2. LTE	108
10.3.3. LoRa	113
10.3.4. MIOTY	120
10.4. Optimierungspotentiale	126
11. Zusammenfassung	132
11.1. Fazit	132
11.2. Nutzen für KMU, innovativer Beitrag und industrielle Anwendungsmöglichkeiten	133
11.3. Mögliche zukünftige Arbeiten	135
A. Verwendung der Zuwendung	136
B. Notwendigkeit und Angemessenheit der geleisteten Arbeiten	137
C. Plan zum Ergebnistransfer in die Wirtschaft	138
D. Abkürzungsverzeichnis	141
E. Abbildungsverzeichnis	145
F. Tabellenverzeichnis	147
G. Literatur	148

1. Executive Summary

In times of the Internet of Things (IoT), nearly all fields of application in automation industry are affected by increasing networking. Besides a number of solutions from the fields of narrowband Low Power Wide Area network (LPWAN) technologies using license-free frequencies, such as LoRa, SIGFOX, Weightless or MIOTY, the 3rd Generation Partnership Project (3GPP) as the standardization body for cellular mobile communications has also discovered this market. As a result, 3GPP has developed various alternatives for fourth-generation cellular (4G) LTE from Rel. 13 onward, with reduced bandwidths and data rates compared to the standard LTE variant, promising higher user density, lower costs and lower power consumption. These new narrowband IoT (NB-IoT) variants have been part of the 5G standard since Rel. 15 as an integral part of the Massive Machine Type Communications profile (mMTC). Thus, several communication technologies are facing each other, having their own characteristics and differ regarding usability for specific application scenarios.

Due to rapid developments and the availability of different communication technologies, numerous questions arise regarding their usability, being of high importance especially for German industrial automation technology. Therefore, the project TestKomm 4.0 has built a test environment that helps — in addition to the theoretical investigations that have been made — to answer these questions. This testbed allows to reproducibly execute, trace and measure representative test cases by integrating different levels of abstraction starting from simulation via emulation to field tests. Finally, results that are nearly independent from vendor and technology are provided as well as a test environment that allows companies to validate and optimize their own products.

2. Einleitung

Im Zeitalter des Internet of Things (IoT) sind praktisch alle Anwendungsfelder der Automatisierungstechnik von dem zunehmenden Trend der Vernetzung betroffen [1], was diesen zu einem so genannten „Megatrend“ werden lässt. Die maßgebliche Motivation für zunehmende Vernetzung von Maschinen und Anlagen im Sinne der Machine-to-Machine-(M2M)-Kommunikation besteht darin, dass in den räumlich verteilten und vernetzten Anwendungen Effizienz gesteigert und Kosten gespart werden können, so dass neue Anwendungen, Geschäfts- oder Dienstleistungsmodelle überhaupt erst möglich werden. Diese Aussage gilt insbesondere auch für die industrielle Automation.

Vor dem Hintergrund der potenziellen Umsatzmöglichkeiten und auch unter dem Druck konkurrierender Lösungen aus den Bereichen der schmalbandigen low power wide area network (LPWAN)-Netzwerktechnologien unter Nutzung lizenzfreier Frequenzen, wie LoRa®, bzw. LoRaWAN [2], SIGFOX [3] oder MIOTY [4, 5], hat das 3rd Generation Partnership Project (3GPP) [6] als relevantes Standardisierungsgremium für die zellulare Mobilkommunikation diesen Markt ebenfalls entdeckt. In der Folge hat 3GPP für die vierte Mobilfunkgeneration (4G) LTE ab der Release 13 (Rel. 13) verschiedene Alternativen entwickelt, deren Bandbreiten und Datenraten gegenüber der Standard-LTE-Variante reduziert sind und die somit eine höhere Nutzerdichte, geringere Kosten und eine geringere Energieaufnahme versprechen [7]. Diese neuen Narrowband-IoT-(NB-IoT)-Varianten sind seit Rel. 15 Teil des 5G-Standards als integraler Bestandteil der massive machine type communication (mMTC)). Im Weiteren werden die LPWAN- und die NB-IoT-Technologien als Narrow Band Wireless Area Networks (NB-WWAN)-Technologien bezeichnet.

2.1. Problem und Aufgabenstellung

Vor dem Projektstart wurden bereits weltweit die obigen Erweiterungen der 4G-Netze installiert, die in modernen Netzen in Bezug auf die Basisstationen (evolved NodeB, eNB) rein in Software abgebildet werden können [8]. In dieser Situation der rapiden Entwicklungen stellten sich zahlreiche Fragen und Herausforderungen, die vor allem für die vielen Systementwickler und Anwender der deutschen industriellen Automatisierungstechnik von grundlegender Bedeutung sind:

- Welche Leistungsfähigkeit kann mit welchen der zur Verfügung stehenden NB-WWAN-Lösungen unter idealen und unter realen Umgebungsbedingungen erreicht werden? Von besonderer Bedeutung waren hier technische Systemparameter wie Latenzzeiten, Jitter, Zuverlässigkeit, Energiebedarf und Durchgängigkeit der Datenübertragung, aber auch kommerzielle Rahmendaten, wie die damaligen und zukünftig potenziell erreichbaren Investitions- und Betriebskosten.
- Mit welchen Parametern erreichen die jeweiligen Technologien jeweils welche Leistungsfähigkeit? Diese Fragestellung war insbesondere für die NB-IoT-Technologien mit ihrem Variantenreichtum von großer Bedeutung.

- Insbesondere musste die Frage beantwortet werden, ob (und unter welchen Randbedingungen) mit diesen Ansätzen auch eine zuverlässige und hart echtzeitfähige Funkkommunikation im Industriebereich unterstützt werden kann.
- In der Folge stellte sich die Frage, welche Anwendungen mit diesen Leistungsdaten damals und in der Zukunft sinnvoll unterstützt werden können.
- In welcher Weise können LPWAN-Technologien mit zellularen Mobilkommunikationstechnologien konkurrieren, bzw. diese sinnvoll ergänzen?
- Mit welchen Anpassungen könnten die verfügbaren NB-WWAN-Technologien so erweitert und optimiert werden, dass sie für die in der industriellen Automatisierungstechnik relevanten Anwendungen möglichst effizient eingesetzt werden können? Hierbei sollten vor allem Algorithmen und Verfahren zum Netzwerkmanagement, zum Abgleich zwischen Control- und Data-Plane sowie zur Frequenzzuweisung untersucht und ggfls. erweitert werden.
- Wie kann die Leistungsfähigkeit einer Technologie, eines Produkts oder einer Installation durch Simulationsmodelle effizient vorhergesagt werden?
- Welchen Einfluss könnten und sollten diese Erfahrungen auf die zukünftige Standardisierung sowohl des 3GPP als auch der LPWAN-Technologien haben?

2.2. Lösungsansatz

Für die Beantwortung obiger Fragen wurde im TestKomm 4.0-Projekt neben den theoretischen Untersuchungen eine Testumgebung aufgebaut, auf der repräsentative Testfälle für NB-WWAN-Technologien reproduzierbar durchgeführt und nachvollziehbar getestet werden können. Diese Testumgebung integriert unterschiedliche Abstraktionsebenen von der Simulation über eine Emulation bis hin zu Feldtests und kann mit Hilfe von einheitlichen Testcase-Beschreibungen in TTCN-3 angesteuert werden. Passive Kanaleigenschaften und aktive Störer können aktiviert werden. Es werden hersteller- und technologieneutrale Vergleichsergebnisse bereitgestellt. Zudem erlaubt es die Testumgebung Unternehmen, ihre innovativen Produkte zu validieren und zu optimieren.

2.3. Projektlauf

Bereits vor Beginn der Arbeiten zu TestKomm 4.0 verfügten die Forschungseinrichtungen über tiefes Know-How im Bereich der Kommunikationstechnik und der damit verbundenen Methodik zum Testen solcher Systeme. Diese Kenntnisse wurden innerhalb des Projekts intensiv genutzt und weiter vertieft. Zudem wurden neue Werkzeuge geschaffen, die für künftige Test- und Evaluierungsaktivitäten von großem Nutzen sein werden. Zu Beginn des Projekts wurden die technischen Parameter und Anforderungen erarbeitet. Hierzu wurden in Absprache mit dem Projekt begleitenden Ausschuss (PA) die zu testenden und zu analysierenden Kommunikationsprotokolle und die damit verbundenen

Anwendungsfälle definiert. Diese Anforderungen flossen sowohl in die Parametrierung und Anpassung der Simulationsmodelle als auch in die Definition der Testfälle mit ein. Zudem waren die Anforderungen maßgeblich ausschlaggebend für die Umsetzung der Emulationsumgebung und die Durchführung der Realtests. Mit Hilfe der im Projekt angepassten Simulationsmodelle, der aufgebauten Emulationsumgebung und einem Satz an Real-Testfällen wurden die festgelegten Technologien anhand der von den Anwendungsfällen abgeleiteten Parameter evaluiert. Basierend auf den Ergebnissen konnten wiederum iterativ Parameteranpassungen durchgeführt werden, um anwendungsspezifisch optimierte Konfigurationen zu erarbeiten.

Die Ergebnisse wurden fortlaufend an die Mitglieder des PA und weitere Interessierte verteilt, um diesen die Möglichkeit zu geben, die Ergebnisse für ihre internen Anwendungsfälle zu nutzen und ggfls. weitere Testfälle zu definieren. Hieraus entstand ein durchgängiger Prozess, durch den eine sukzessive Annäherung an die Projektanforderungen möglich wurde.

2.4. Zusammenarbeit mit Dritten

Die Arbeiten zu TestKomm 4.0 waren durch enge Kooperationen z.B. mit Unternehmen, Herstellern und anderen Projekten geprägt. Insbesondere mit dem PA herrschte ein reger Austausch. So wurden in diesem Rahmen mehrere Sachleistungen zur Verfügung gestellt, wie z.B.

- NB-IoT Referenzplattformen inkl. Zubehör durch die Firma Commsolid GmbH [9]
- Mehrere Entwicklungskits wie Multi Technologies Low Power Wireless Nodes for LoRa and IEEE802.15.4 durch die Fa. senTec Elektronik GmbH [10]
- Diverse NB-IoT Evaluation Kits durch die u-blox AG [11]
- MIOTY Geräte durch die Fa. STACKFORCE GmbH [12]

Zudem unterstützte der PA das Projekt in Form von Dienstleistungen, überwiegend im Bereich der Entwicklungsunterstützung und Messdurchführung. Weitere Kontakte fanden im Rahmen der regelmäßigen Projekttreffen des Projektbegleitenden Ausschusses statt.

2.5. Verwendung und Veröffentlichung der Ergebnisse

Der Projektfortschritt sowie die Ergebnisse wurden insbesondere auf Treffen mit dem PA vorgestellt und diskutiert. Die Ergebnisse wurden bereits Mitgliedern des PA und weiteren Interessierten zur Verfügung gestellt und werden dort genutzt. Im Rahmen der Arbeiten zu TestKomm 4.0 sind zudem eine Reihe von wissenschaftlichen wie nichtwissenschaftlichen Veröffentlichungen entstanden. Eine Aufstellung der einzelnen Maßnahmen und Veröffentlichungen findet sich in Anhang C. Es ist geplant, sowohl die Ergebnisse als auch die getätigten Entwicklungen in entsprechenden Folgeprojekten einzusetzen und zu erweitern.

3. Stand der Technik

3.1. Wireless Industrial IoT

Mit dem zunehmenden Trend zur Digitalisierung und drahtlosen Vernetzung werden räumlich verteilte drahtlose Netzwerke (engl. Spatially Distributed Wireless Networks (SDWN)) für viele Anwendungsbereiche attraktiv. Um eine nahtlose drahtlose Konnektivität in vielen internet-of-things (IoT) basierenden Industrie 4.0-Anwendungsfällen wie Fabrikautomatisierung, Prozessautomatisierung, Logistik und M2M zu ermöglichen, spielen SDWN wie Short-Range Wireless Networks (SRWN), NB-WWAN und 5G-Netzwerke eine wichtige Rolle. Die Motivation für die zunehmende Vernetzung von Maschinen und Anlagen im Sinne der Industrial Internet of Things (IIoT)-Kommunikation ist, dass in räumlich verteilten und vernetzten Anwendungen die Effizienz gesteigert und die Kosten gesenkt werden können. Die Systemanforderungen in diesen Anwendungen sind in Bezug auf Latenz, Zuverlässigkeit, Batterielebensdauer, Periodizität, Reichweite, Bandbreite, Datenrate, Mobilität etc. vielfältig. So sind beispielsweise Anwendungen aus der Fabrikautomation durch eine hohe Dichte von Knoten in einer Produktionsanlage (Produktionszelle) gekennzeichnet, während die Gesamtzahl der Geräte relativ gering ist. Gleichzeitig sind sehr schnelle Reaktionen (bis zu 10 ms oder sogar 1 ms) erforderlich [13]. Im Gegensatz dazu, ist die jeweilige Datenmenge und die logische Tiefe der Verknüpfungen eher gering. Prozessautomatisierungsanwendungen sind räumlich viel weiter verteilt, möglicherweise über mehrere Quadratkilometer einer chemischen oder biologischen Prozessanlage. Die Reaktionen sind weniger zeitkritisch, aber von größerer logischer Tiefe als bei Anwendungen der Fabrikautomation, da z.B. manche Gefahrensituationen erst in der Kombination zahlreicher Messwerte erkannt werden können. Dies impliziert auch eine hohe Anzahl von räumlich verteilten Endgeräten. Die Datenmenge pro Sensor ist in der Regel ebenfalls gering, kann aber in der Kumulation signifikant werden. Anwendungen der klassischen M2M-Kommunikation, z.B. zur Infrastrukturüberwachung, sind räumlich am weitesten verteilt. Hier kann es vorkommen, dass nur ein einziger Knoten in einem System installiert ist. Allerdings gilt die Skalierbarkeit für diesen Anwendungsbereich als sehr hoch, da mehrere tausend Geräte angeschlossen werden können.

Die meisten 3G- und 4G-Dienste können die vielfältigen Kommunikationsanforderungen wie hohe Datenrate, hohe Zuverlässigkeit, hohe Abdeckung, geringe Latenz usw. nicht erfüllen [14], [15].

Basierend auf den Systemanforderungen dieser Anwendungsfälle müssen verschiedene Technologien bewertet werden, um eine geeignete Funklösung auszuwählen. Um die verschiedenen Arten von Netzwerktechnologien zu testen und zu vergleichen und um sicherzustellen, dass sie die unterschiedlichen Anforderungen wie Zuverlässigkeit, niedrige Datenrate, geringe Bandbreite, große Reichweite und lange Batterielebensdauer erfüllen, spielen systematische Funktionstests und Leistungsmessungen eine wichtige Rolle. Hierfür existieren verschiedene Methoden die meistens jedoch entweder technologiespezifisch sind oder auf verschiedenen Abstraktionsebenen basieren, die die Systemumgebungen modellieren oder prototypisieren, wodurch die Methoden nicht

einheitlich sind.

Nachfolgend werden die in diesem Projekt relevanten Technologien vorgestellt und anhand ihrer theoretischen Leistungsdaten gegenübergestellt.

3.2. Low Power Wide Area Networks

Ein LPWAN ist ein drahtloses Netzwerk, das eine Kommunikation über große Entfernungen ermöglicht, um Geräte mit geringer Bandbreite und batteriebetriebene Geräte, wie z.B. Sensoren, miteinander zu verbinden. LPWANs versprechen ein neues Niveau von Verbindungsbudgets bei niedriger Ausgangsleistung und Kosten, die eine weitreichende und stabile lokale Kommunikation bei IoT-Einsätzen ermöglichen. Bei LPWAN handelt es sich nicht um eine einzelne Technologie, sondern um eine gemeinsame Bezeichnung für verschiedene Technologien für Weitverkehrsnetze mit geringem Stromverbrauch, die aus unterschiedlichen Bausteinen bestehen [16].

Im Allgemeinen können LPWANs in unlizenzieren und lizenzierten Frequenzbändern betrieben werden. Die im Projekt betrachteten LPWANs LoRaWAN und MIOTY nutzen in Europa das nichtlizenzierte 863-870MHz Frequenzband, welches prinzipiell von jedem genutzt werden kann. Um gegenseitige Störungen zu vermeiden, werden jedoch regional einschränkende Maßnahmen getroffen [17]. Diese müssen zwingend beachtet werden, da sie einen signifikanten Einfluss auf die Nutzbarkeit einer Technologie haben können [18].

In der EU beispielsweise beziehen sich diese Einschränkungen auf die maximal erlaubte Sendeleistung, sowie die erlaubte Nutzungszeit eines Frequenzbandes. Letzteres wird über den sogenannten Duty-Cycle limitiert. Entscheidend hierbei ist die Dauer, über die ein Paket über die Luft gesendet wird. Bei einem Duty-Cycle von 1% z.B. darf eine Anwendung, die Pakete mit einer Sendedauer von einer Sekunde sendet, nur alle 100s ein neues Paket übertragen. In Abb. 1 ist der Einfluss des Duty-Cycles auf Anwendungen dargestellt. Beispielsweise werden durch diese Regulierungen direkte Sendewiederholungen unterbunden. Um eine bidirektionale Übertragung möglich zu machen, werden meist verschiedene Subbänder für Uplink (UL) und Downlink (DL) verwendet.

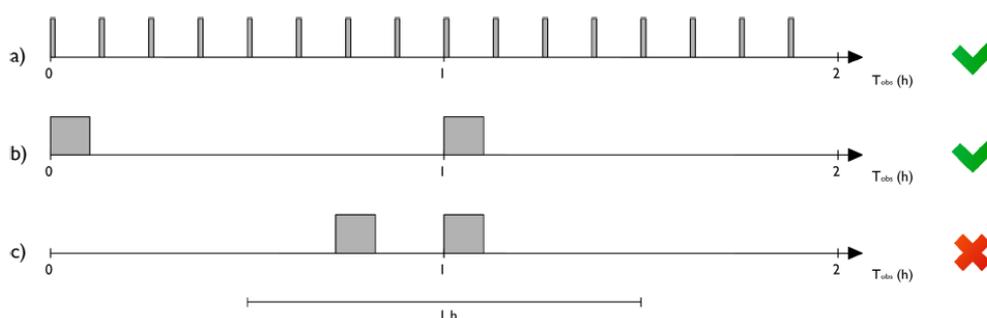


Abbildung 1: Einfluss des Duty-Cycle auf Anwendungen in unlizenzieren Bändern

Für Endgeräte im europäischen 863-870MHz Band ist die Leistung auf 25 mW Effektiv

Strahlungsleistung (ERP) limitiert und es ist ein Duty-Cycle von 1% vorgesehen [17]. Durch Limitierungen in der Sendeleistung wird gleichzeitig auch die Reichweite verringert und durch die Limitierung des Duty-Cycles wird der Durchsatz, aber auch die Flexibilität einer Anwendung eingeschränkt, die beispielsweise nicht beliebig oft Sendewiederholungen durchführen kann. Trotz diesen Einschränkungen kommt es in diesen Frequenzbändern zu vermehrten Störungen, da dennoch nicht reguliert werden kann, wann welcher Teilnehmer sendet. Somit kann eine Störbeeinflussung nicht ausgeschlossen werden.

3.2.1. LoRaWAN

LoRa/LoRaWAN ist eine proprietäre Technologie für LPWAN und basiert auf Chirp-Spread-Spektrum (CSS) Modulation. LoRa beschreibt hierbei die physische Schicht (Layer 1 im ISO/OSI Modell). LoRaWAN ist das Protokoll der höheren Schichten und ist „nominell“ ein offener Standard, der von der LoRa Alliance verwaltet wird. Allerdings sind entsprechende Chips, die die LoRa-Modulation implementieren, nur von Semtech erhältlich. LoRaWAN unterstützt private und öffentliche Netzwerkstrukturen. Die LoRa-basierte LPWAN-Technologie ist weit verbreitet, da viele Netzbetreiber, Netzserver, Outdoor-Gateways und Endgeräte im Handel erhältlich sind.

Es stehen Bandbreiten von 125 kHz, 250 kHz und 500 kHz zur Verfügung, wobei in Europa, außer für die FSK Modulation, nur eine Bandbreite von 125 kHz verwendet werden kann, wodurch die Datenrate auf 5.7 kbit s^{-1} begrenzt ist [19].

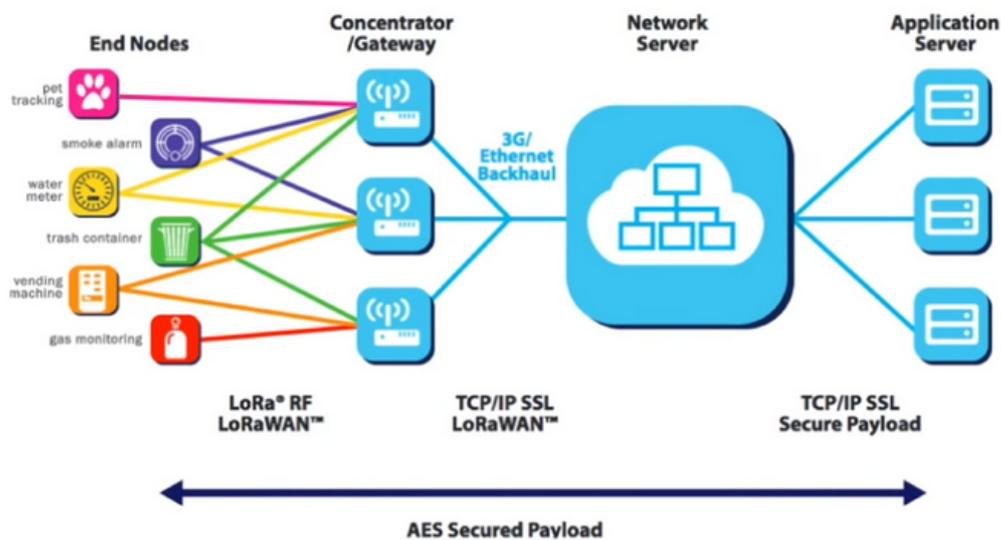


Abbildung 2: LoRaWAN Netzwerkkonstruktion [20]

In Abb. 2 ist die LoRaWAN Netzwerkkonstruktion dargestellt. Es verwendet eine sternförmige Netzwerktopologie, in der die LoRa-Gateways als transiente Brücke fungieren, die Nachrichten zwischen den Endgeräten und dem zentralen Netzwerk im Backend aus-

tauschen. Pro Zelle können bei LoRa bis zu 50.000 Endgeräte [21] unterstützt werden, wobei solch eine hohe Zahl vor allem aufgrund von regionalen Regulierungen deutlich geringer ausfällt bzw. die Leistung deutlich limitiert [18].

Der Netzwerkservers implementiert das LoRaWAN Protokoll und ist unter anderem für die Verwaltung der Gateways, die Authentifizierung der Geräte und für die sogenannte adaptive Datenratensteuerung verantwortlich. Die sichere Datenübertragung wird durch symmetrische kryptografische Verfahren sowie Sitzungsschlüssel garantiert. Jedes LoRa-Endgerät besitzt spezifische Netzwerk- und Applikationsschlüssel, die bei der Herstellung festgelegt werden, was durch die Registrierung und Aktivierung eines Geräts in einem Netzwerk geschieht.

Der Netzwerkservers sendet seine Daten weiter an einen Applikationsservers, der die Anwendungsdaten entschlüsselt. Hier können eigene Anwendungen, wie beispielsweise eine Sensordatenvisualisierung, integriert werden.

Innerhalb der EU wird von LoRa der Frequenzbereich von 868–868.6 MHz genutzt. Damit ist die Sendeleistung der Endgeräte auf 25 mW begrenzt und es gilt ein Duty-Cycle von 1% [17]. Für den Downlink nutzt es das Frequenzband von 869.4–869.65 MHz und ist daher im Downlink auf 500 mW und 10% Duty-Cycle begrenzt.

Die Datenrate und die Stabilität der Datenübertragung wird über den Spreizfaktor (SF) eingestellt. Ein hoher Spreizfaktor erlaubt die Übertragung über größere Strecken, da eine größere Signalspreizung die Empfangsempfindlichkeit erhöht. Es bedingt aber einen geringeren Durchsatz, eine längere Übertragungszeit und einen höheren Energiebedarf, aufgrund der längeren Signalübertragungsdauer. In Tab. 1 ist der Zusammenhang zwischen Spreizfaktor, Datenrate, Duty-Cycle und die daraus resultierende maximale Anzahl an Nachrichten pro Stunde dargestellt. Weiterhin ist der Zusammenhang zwischen dem Spreizfaktor und der Empfindlichkeit dargestellt. Die Datenrate ist hierbei für jeweils ein Paket berechnet. Über mehrere Pakete gerechnet wird sie durch den Duty-Cycle begrenzt. Die Werte für die Übertragungszeiten beziehen sich auf eine Datengröße von 50 Bytes, eine Bandbreite von 125 kHz und eine Coderate von 1.

Tabelle 1: Zusammenhang zwischen Spreizfaktor, Datenrate, Übertragungsdauer, Duty-Cycle [22, 22]

Parameter	SF7	SF8	SF9	SF10	SF11	SF12
Empfindlichkeit [dBm]	-123	-126	-129	-132	-134,5	-137
Max. paketgröße [Bytes]	222	222	115	51	51	51
Zeit in Luft [ms]	118,0	215,6	390,1	698,4	1478,7	2793,5
Min. Sendeintervall [s] (DutyCycle)	11,8	21,6	39,0	69,8	127,9	279,4
Datenrate [bps]	5468,75	3125	1757,8	876,56	5371	293

3.2.2. MIOTY

MIOTY zählt ebenfalls zu den LPWAN, die zum Aufbau privater IoT-Netzwerke verwendet werden [5]. Die Technologie wurde vom Fraunhofer IIS entwickelt und nutzt Telegram Splitting Multiple Access (TSMA), basierend auf dem ETSI-Standard TS 103357 [23] für LPWAN-Systeme. Hierbei werden Pakete in kleinere Pakete aufgeteilt und als sogenannte packet bursts nach einem bestimmten TSMA-Muster in einem Frequenz-Zeit Schema über die Luft gesendet. Durch definierte Übertragungspausen zwischen den Bursts sowie den unterschiedlichen Frequenz-Zeit-Mustern sollen so Paketverluste durch Interferenzen verringert werden [23, 24].

In Abb. 3 ist die Netzwerkarchitektur von Mioty dargestellt. Es wird eine sternbasierte Topologie zwischen Endgeräten und der MIOTY (MIOTY) Basisstation verwendet. Die Basisstation empfängt Daten der registrierten Endgeräte und leitet sie über eine Backhaul-Verbindung an ein Mioty-Backend weiter. Im Backend sind ein Service Center, das unter anderem für das Netzwerkmanagement für mehrere Basisstationen zuständig ist, und das Application-Center, das für das Dekodieren von anwendungsbezogenen Daten zuständig ist, definiert [25].

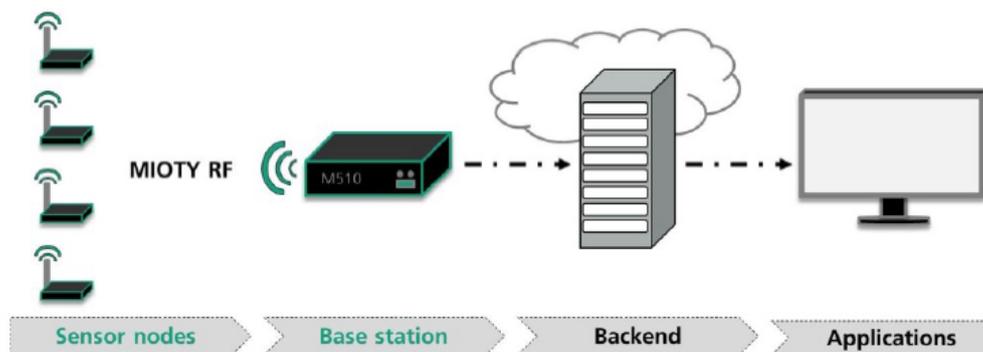


Abbildung 3: Mioty Netzwerkarchitektur [25]

MIOTY basiert auf der Ultra Narrow Band (UNB)-Technologie mit sehr schmaler Signalbandbreite (2 kHz), um eine Datenkommunikation über große Entfernungen zwischen tausenden von IoT-Geräten und einer Basisstation zu ermöglichen. MIOTY arbeitet mit Standard-Transceivern. Das Grundkonzept dieser Technologie ist das Telegramm-Splitting, das eine kompakte Telegrammübertragung in viele gleichgroße Funkbursts aufteilt. Bei Mehrfachzugriff werden die Funkbursts über Zeit und Frequenz verteilt. Für eine korrekte Dekodierung müssen nur 50% der Funkbursts kollisionsfrei sein. Dadurch wird die Kollisionswahrscheinlichkeit der Telegramme verringert und die Toleranz gegenüber Störungen erhöht [24]. Es handelt sich bei MIOTY um keine Open Source Plattform und es gibt aktuell vergleichsweise wenige Produkte für MIOTY, da es noch eine sehr neue Technologie ist. Allerdings ist hervorzuheben, dass viele der gängigen Transceiver-ICs für den Einsatz in MIOTY-Netzen genutzt werden können [24].

Das Fraunhofer Institut IIS bietet Basisstationen an, die an die eigenen Bedürfnisse angepasst werden können. Außerdem können mittlerweile Software-Lizenzen über

weitere MIOTY-Alliance-Mitglieder für die Entwicklung eigener Endgeräte erworben werden [5].

MIOTY nutzt den Frequenzbereich von 868–868.6 MHz. Damit ist die Sendeleistung der Endgeräte auf 25 mW begrenzt und es gilt ein Duty-Cycle von 1%. Für den Downlink nutzt es das Frequenzband von 869.4–869.65 MHz und ist daher im Downlink auf 500 mW und 10% Duty-Cycle begrenzt [23].

In Mioty sind zwei verschiedene Modi mit jeweils drei unterschiedlichen Bandbreiten definiert. Neben einem Stromsparprofil (Ultra Low Power (ULP)) ist ein Reichweitenprofil (Extended Range (ER)) definiert. Je nach Modus werden unterschiedliche TSMA-Muster verwendet. Die wichtigsten Eigenschaften der Sendemodi sind in Tab. 2 dargestellt.

Tabelle 2: Übersicht wichtiger Mioty-Parameter [23]

Parameter	UL-ULP	UL-ER	DL
Symbolrate [Sym- b/s]	2380,371	396,729	2380,371
MCL [dB]	153	164	161
Burst Dauer [ms]	15,12	90,74	11.76–21.43
Bandbreite [Hz]		Narrow: 396,729 Standard: 2380,371 Wide: 28564,453	

Es werden unidirektionale und bidirektionale Endgeräte definiert. Nur die bidirektionalen Endgeräten können Nachrichten im Downlink empfangen. Bevor eine Kommunikation zwischen Endpunkt und Basisstation stattfinden kann, muss sich dieser bei der Basisstation registrieren. Bidirektionale Endpunkte können diese Registrierung über die Luft vornehmen, während unidirektionale Endpunkten bereits manuell in der Basisstation registriert sein müssen.

3.3. Zelluläre Netzwerke

Zelluläre Netzwerke werden klassischerweise von Mobilfunkbetreibern bereitgestellt. Diese besitzen Lizenzen, um in bestimmten Funkbereichen senden zu dürfen, müssen gleichzeitig aber gewisse Kriterien bezüglich des Netzausbaus befolgen [26]. Diese Kriterien beziehen sich größtenteils auf die Bedarfe der Endnutzer, sodass das Kernziel der zellulären Netzwerke in der Vergangenheit die Erhöhung der Datenraten und die Bereitstellung nutzerzentrierter Services war.

Die Spezifikation der zellulären Dienste wird durch das 3GPP durchgeführt [6]. Seit Release 13 der 3GPP-Spezifikation werden auch Applikationen im Bereich des IoT in der Spezifikation adressiert. Unter anderem werden schmalbandige Lösungen spezifiziert, mit ähnlichen Leistungsdaten wie die klassischen LPWANs. Diese werden von 3GPP unter dem Oberbegriff des cellular IoT (cIoT) standardisiert werden.

Im Vergleich zu den LPWANs kommunizieren zelluläre Netzwerke über ein lizenziertes

Funkspektrum. Die Schmalbandversionen der Mobilfunktechnologien verfügen über eine geringere Bandbreite und vereinfachte Knoten- und Netzverwaltungsmechanismen wie NB-IoT und LTE-M. Die am besten geeignete zelluläre IoT-Technologie für LPWAN-Anwendungsfälle ist Narrowband IoT (NB-IoT). Die Schmalbandversionen der Mobilfunktechnologie zeichnen sich durch eine geringere Bandbreite und vereinfachte Knoten- und Netzverwaltungsmechanismen aus.

In diesem Projekt wird NB-IoT als *cl*oT Lösung und LTE als Breitbandtechnologie untersucht. Hierbei werden insbesondere private zelluläre Netzwerke betrachtet. Diese Netzwerke werden nicht mehr von Netzbetreibern verwaltet, sondern privat unter einer lizenzierten Funkfrequenz [27]. Dieses neue Nutzungsmodell ist Teil des 5G-Standard. Da NB-IoT seit Release 15 nun auch Teil des 5G Standards ist, spielt diese Technologie insbesondere auch in privaten Netzwerken eine sehr wichtige Rolle.

3.3.1. 4G - LTE

Long Term Evolution (LTE) wurde von der 3GPP als Nachfolger der 3G (UMTS) Standards seit Release 8 spezifiziert und ist als Teil des öffentlichen Mobilfunknetzes weit verfügbar. Eine Hauptänderung gegenüber der alten Mobilfunkstandards ist, dass das Netzwerk komplett paketbasiert aufgebaut ist und die Architektur deutlich geändert wurde, was eine Interoperabilität mit anderen IP basierten Netzwerken vereinfacht. Eine vereinfachte Architektur eines 4G Netzwerkes ist in Abb. 4 dargestellt. Endgeräte werden im Mobilfunkbereich mit User Equipment (UE) bezeichnet. Das Endgerät kommuniziert über ein Radio Access Network (RAN) mit der evolved Node B (eNB), die die Daten in das Kernnetzwerk (evolved packet core (EPC)) weiterleitet.

LTE ist eine Mobilfunktechnologie, die eine Hochgeschwindigkeitsdatenübertragung für mobile Geräte ermöglicht. Sie unterstützt auch IoT-Geräte, doch ihr hoher Stromverbrauch und die begrenzte Netzkapazität schränken ihre Eignung für industrielle IoT-Anwendungsfälle ein. Vielmehr ist sie für niedrig latente Anwendungen relevant.

In diesem Projekt war diese Technologie aufgrund der hohen Bandbreite, hohen Übertragungsraten sowie der hohen Verbreitung und der damit verbundenen sehr guten Verfügbarkeit relevant. Im Gegensatz zu den anderen Technologien werden jedoch andere Anwendungsfälle adressiert.

Gegenüber den LPWA-Lösungen ist LTE komplexer, da es deutlich andere Anforderungen erfüllen und insbesondere sehr flexibel auf Nutzeranforderungen reagieren können muss. Aus diesem Grund sind zahlreiche adaptive Maßnahmen in LTE implementiert und es sind viele Netzwerkparameter für die Parametrierung des RAN verfügbar [28]. In öffentlichen Netzwerken werden diese Netzwerkparameter von den Netzbetreibern individuell angepasst. Hierin liegen die Möglichkeiten und die Herausforderungen, die sich beim Betrieb von privaten zellulären Netzwerken ergeben.

In LTE wird Orthogonal Frequency Division Multiple Access (OFDMA) als Übertragungstechnik eingesetzt. Hierzu werden die Daten über viele orthogonale Hilfsträger übertragen. Die Anzahl der Hilfsträger ist als Netzwerkparameter einstellbar und bestimmt unter anderem die Menge an Daten, die pro Zeiteinheit gesendet werden können.

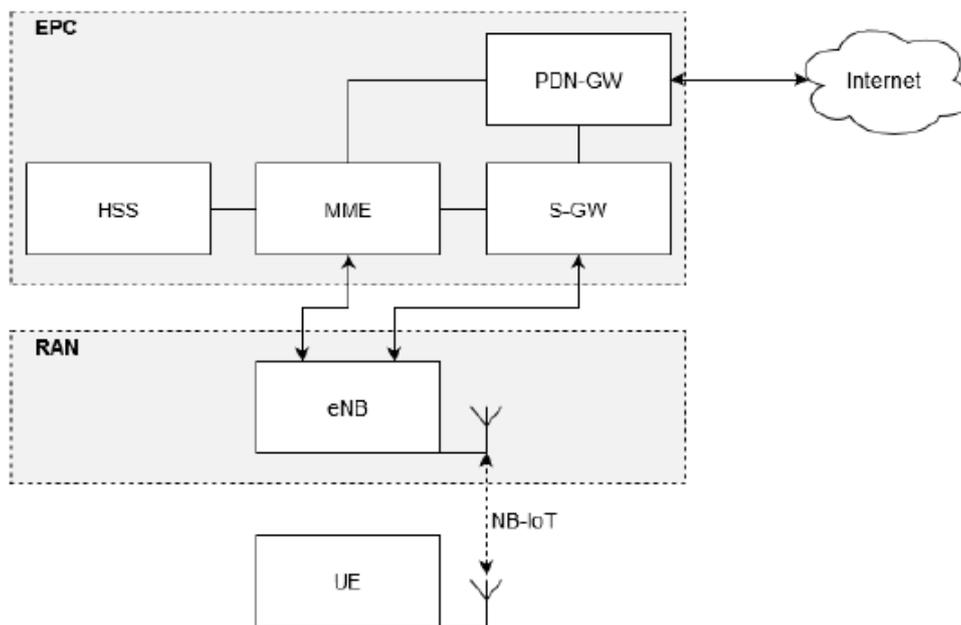


Abbildung 4: Vereinfachte 4G Netzwerkarchitektur

3.3.2. 5G- NR

Die fünfte Mobilfunkgeneration 5G ist seit Release 15 spezifiziert und wird seither stetig erweitert. Die fünfte Mobilfunkgeneration markiert einen weiteren Meilenstein in der Entwicklung der Mobilfunktechnologien und bietet zahlreiche neue Konzepte und potentielle Anwendungsfälle, die mit vorigen Mobilfunklösungen nicht umsetzbar waren. Die Anforderungen an 5G orientieren sich an der ITU-Vision IMT-2020 und sind in der 3GPP-Spezifikation TS 22.261 festgehalten [29].

Es lassen sich drei Hauptanwendungsgebiete identifizieren. Das erste ist Ultra-Reliable and Low-latency Communications (URLLC), bei dem sehr geringe Latenzen von bis zu 1ms und eine 99,99% Paketerfolgsrate bei Daten von bis zu 32B erreicht werden sollen. Das zweite ist eMBB (embb), das höhere Datenraten von bis zu 20Gbps, höhere Kapazitäten und Datenvolumen ermöglicht. Das dritte Anwendungsgebiet ist mMTC, bei dem eine hohe Anzahl von IoT-Geräten mit einer Dichte von bis zu 1 Million pro Quadratkilometer und Unterstützung von mehrjährigen Jahren Batteriebetrieb unterstützt wird [30].

Wie auch in LTE, wird in 5G OFDMA als Übertragungstechnik verwendet, allerdings mit unterschiedlichen Numerologien. Dadurch können auch niedriglatente Anwendungen, durch eine kürzere Framedauer (mini-slots), unterstützt werden [30].

Aus Sicht industrieller Anwendungen sind insbesondere die folgenden charakteristischen Eigenschaften von 5G interessant, die allerdings teilweise noch nicht komplett umgesetzt sind [31].

- Network Slicing: Ein Netzwerk kann in logische Abschnitte mit unterschiedlichen

Konfigurationen aufgeteilt werden.

- Quality of Service (QoS): Bereitstellung einer bestimmter Service-Güte auf Paketbasis.
- TSN-Unterstützung: Unterstützung niedrig latenter Kommunikation über den TSN-Standard .
- Bereitstellung zahlreicher Services wodurch auch externe Systeme mit dem internen 5G Netzwerk Daten austauschen können.
- Campusnetze: Privater Betrieb eines 5G Netzes.

3.3.3. NB-IoT

NB-IoT ist Teil des LTE und wurde in Release 15 des 3GPP in die 5G-Spezifikation übernommen. Zur Komplexitätsreduktion wurde die Funkschnittstelle im Vergleich zu 4G komplett überarbeitet und zudem weitere Optionen zur Einsparung von Energie eingeführt [32]. NB-IoT wurde von der 3GPP aufgrund des stetig breiter werdenden Spektrums von Anwendungsfällen für Machine Type Communication (MTC) eingeführt. Es ist vorgesehen für IoT-Geräte im Ultra-Low-End-Bereich. Einige Kernpunkte von NB-IoT sind nachfolgend aufgelistet [33]:

- Hohe Reichweite: 20 dB MCL gegenüber GSM.
- Hohe Batterielaufzeit: Bis zu über 10 Jahren.
- Garantierte Latenz: Bis zu 10 s.
- Viele Verbindungen pro Zelle: Über 50 tausend pro Zelle.
- Kostengünstige Endgeräte: Geräte für weniger als 5 USD.
- Guter Netzausbau durch die Nutzung der bestehenden LTE-Infrastruktur.
- Hohe Sicherheit durch die Nutzung von LTE-Sicherheitsmechanismen.

NB-IoT verwendet mit Quadrature Phase Shift Keying (QPSK) und Binary Phase-Shift-Keying (BPSK) robuste Modulationsverfahren. Als Übertragungsverfahren wird OFDMA im Downlink und SC-FDMA im Uplink verwendet, da dieser den Energiebedarf deutlich reduziert. Zur Übertragung werden zwölf 15 kHz-Hilfsträger verwendet, um die zur Verfügung stehende Bandbreite von 180kHz auszunutzen [33]. Dies entspricht einem RB in LTE. In 3GPP Release 14 werden drei Leistungsklassen mit 14 dBm, 20 dBm und 23 dBm Ausgangsleistung spezifiziert. Im Gegensatz zu unlicenzierten Lösungen ist die Leistung prinzipiell nicht beschränkt und mit bis zu 23 dBm auch deutlich höher als die anderen betrachteten LPWAs. Dennoch ist die Leistung in den Endgeräten begrenzt, um dennoch kostengünstig und energiesparend sein zu können.

Ein Ziel von NB-IoT ist es, die Leistungsübertragungsbilanz im Gegensatz zu GSM um 20 dB zu erhöhen. Hierfür werden drei verschiedene Modi definiert, die eine Leistungsübertragungsbilanz von 144, 154 und 164 dB ermöglichen. Dazu tragen neben der erwähnten robusten Modulationsart auch die Verwendung von Sendewiederholungen auf der Media-Access-Control-(MAC)-Schicht bei. Der Prozess wird als automatic repeat request (HARQ) bezeichnet. Pakete müssen innerhalb einer gewissen Zeit bestätigt werden, ansonsten wird automatisch die Nachricht erneut gesendet. Die Anzahl möglicher Sendewiederholungen wird hierbei dynamisch angepasst. Bei schlechteren Bedingungen wird die Anzahl der Sendewiederholungen automatisch höher gesetzt, bis auf einen Wert von 128. In diesem Fall reduziert sich die Datenrate jedoch auf 20 bit/s [32].

Um den Energiebedarf im Vergleich zu 4G und anderen Mobilfunkstandards zu reduzieren, wurden viele Bestandteile des LTE-Protokolls entfernt bzw. vereinfacht. Zudem wurden die Energiesparmodi Power Save Mode (PSM) und extended Discontinuous Reception (eDRX) eingeführt. Nur unter Verwendung dieser Energiesparmodi ist es möglich, Batterielebensdauern von bis zu 10 Jahren zu erreichen. In [34] ist eine Lebensdauerberechnung anhand von empirischen Daten für NB-IoT dargestellt. Aus dieser Untersuchung ergibt sich eine Lebensdauer von über 10 Jahren, bei einer täglichen Uplink-Nachricht mit 200 B Nutzerdaten. Bei stündlichen Nachrichten reduziert sich die Lebensdauer deutlich, auf einen Bereich von 0,7–2 Jahre [34, 35].

Neben öffentlichen Netzwerken, die von den Mobilfunkanbietern ausgebaut werden, gibt es die Möglichkeit, private LTE/NB-IoT Netzwerke zu betreiben. Dadurch kann ein lokales LTE/NB-IoT-Netzwerk verwendet werden, welches nur für das Unternehmen und dessen IoT-Geräte nutzbar ist. Dies hat den Vorteil, dass das Netzwerk an die eigenen Bedürfnisse angepasst werden kann, die Daten immer im Unternehmen bleiben und die Netzwerklast rein von der eigenen Anwendung bestimmt ist. Gleichzeitig nimmt der administrative Teil für das Netzwerk zu, welches komplett selbst verwaltet werden muss, die Installationskosten sind deutlich höher und es sind auch hier zusätzliche Wartungs- und Supportkosten zu erwarten. Lösungen in diesem Bereich verwenden ein Software Defined Radio (SDR) und eine Implementierung des Kernnetzwerkes rein in Software. Somit sind auch einfach andere Standards wie 4G oder 5G mit derselben Hardware möglich. Um private Mobilfunknetzwerke betreiben zu können, ist eine Frequenzvergabe durch die Bundesnetzagentur notwendig. Diese beschränkt sich aktuell auf Versuchsfunklizenzen sowie Frequenzen im Bereich von 3,7–3,8 GHz-Bereich. Dieser Bereich wird von 4G und 5G ausschließlich für den Betrieb von Campusnetzwerk-Lösungen benutzt.

3.4. Vergleich drahtloser Technologien

3.4.1. Überblick

Dieser Abschnitt bietet einen Überblick über die wichtigsten Leistungskennzahlen zu den untersuchten LPWAN und Mobilfunklösungen.

Ein Überblick über die theoretische Analyse der verschiedenen Technologien, die in dieser Arbeit betrachtet werden, ist in Tab. 3 zu finden. Es ist offensichtlich, dass trotz

der vielen Anforderungen wie Reichweite, Energieverbrauch, Flexibilität der Nutzlast usw. alle konkurrierenden Technologien auf die eine oder andere Weise versuchen, die Anforderungen zu erfüllen, was diesen LPWAN-Bereich sehr wettbewerbsfähig macht. Die Auswahl der besten Lösung für ein bestimmtes Problem ist eine anspruchsvolle Aufgabe.

Tabelle 3: Theoretischer Vergleich von Leistungsparameter für unterschiedliche drahtlose Technologien

Parameter	LoRa	Mioty	NB-IoT	LTE
Spektrum	unlizenziert	unlizenziert	lizenziert	lizenziert
Modulation	CSS	GMSK	OFDMA/SC-FDMA	OFDMA
MCL	151 dB	154 dB	144–164 dB	140 dB
Bandbreite	125 kHz	2 kHz	180 kHz	1.4–20 MHz
Datenrate (UL)	<27 kbit s ⁻¹	<470 kbit s ⁻¹	<60 kbit s ⁻¹	<150 Mbit s ⁻¹
Datenrate (DL)	<27 kbit s ⁻¹	<470 kbit s ⁻¹	<25 kbit s ⁻¹	<1000 Mbit s ⁻¹
Max. Nutzdatengröße	51–222 B	250 B	1358 B (UDP)	1358 B (UDP)
Max. Anzahl Pakete pro Tag	Duty-Cycle: 1%	Duty-Cycle: 1%	unbegrenzt	unbegrenzt
Offener Standard	ja	nein	teilweise	teilweise
Verbreitung	weit	kaum	weit	weit

3.4.2. Leistungsparameter

Für das Testen von drahtlosen Kommunikationstechnologien müssen Parameter erfasst werden, über die das Systemverhalten beschrieben werden kann. Folgende Leistungsparameter sind von besonderer Bedeutung bei der Betrachtung drahtloser Kommunikationstechnologien:

- **Signalstärke:** Über die Signalstrecke von Sender und Empfänger wirken sich zahlreiche Effekte auf die Signalübertragung aus. Zur Charakterisierung der Verbindungseigenschaften wird die Empfangsleistung anhand von Statistiken wie Received Signal Strength Indication (RSSI) und Signal to Noise Ratio (SNR) ermittelt.
- **Paketerfolgsrate:** Der erfolgreiche Empfang von Paketen ist eine der wichtigsten und elementarsten Anforderungen für Anwendungen. Es gibt viele Faktoren, die

zu Paketverlusten führen, z.B. schwache Signalqualität, Kollisionen, Störungen usw. Da die LPWAN-Anwendungsfälle meist in einer räumlich verteilten Umgebung stattfinden, ist es wichtig, diese Metrik zu analysieren. Zu diesem Zweck wird die Anzahl der gesendeten und empfangenen Pakete gemessen.

- **Latenz:** Anwendungsfälle haben unterschiedliche Anforderungen an die Übertragungsdauer der Daten. Daher wird die Latenz als Leistungsparameter erfasst.
- **Nutzdatenlänge:** Die Länge der Nutzdaten spielt für Anwendungen eine zentrale Rolle und muss daher berücksichtigt werden. Die Anzahl der gesendeten Daten hat einen direkten Einfluss auf die Übertragungsdauer und den Energiebedarf.
- **Koexistenz:** Da die meisten LPWAN-Technologien in unlicenzierten Frequenzbändern arbeiten, besteht die Gefahr von Störungen durch koexistierende Technologien. Es ist wichtig, die Netzstabilität zu bewerten und das Koexistenzverhalten systematisch zu analysieren.

3.5. Tests und Messungen von drahtlosen Technologien

Räumlich verteilte NB-WWAN-Systeme sind die Voraussetzung für viele IoT- und Industrie 4.0-Anwendungsfälle. Um eine zuverlässige Leistung dieser Systeme zu gewährleisten, spielen Funktionstests in verschiedenen Phasen des Systementwicklungsprozesses eine wichtige Rolle.

Die Funktionsprüfung solcher Netzwerke ist jedoch mit vielen Herausforderungen verbunden, denn die drahtlosen Kommunikationsgeräte sind in der Regel ressourcenbeschränkt und werden von komplexen Protokollen und Mechanismen der Kommunikationsschicht unterstützt.

In den meisten Fällen müssen diese Geräte auch zusammenarbeiten, und sie befinden sich in der Regel in einer unbeaufsichtigten Umgebung, was hohe Anforderungen an die Zuverlässigkeit mit sich bringt. Es können einige Firmware-spezifische Fehler auftreten, wie z.B. eine ungültige Interpretation/Implementierung von Normen, Technologien oder Spezifikationen, allgemeine Fehler in der Firmware usw.

Es gibt auch hardwarebezogene Fehler, z. B. aufgrund von Ressourcenbeschränkungen, Zeitproblemen, Hardwareproblemen usw. Ein weiteres großes Problem bei drahtlosen Netzwerken ist, dass sie in der Regel in komplexen Topologien betrieben werden, was viele Probleme z.B. hinsichtlich der Signalausbreitung mit sich bringt. Außerdem sind viele Störeffekte, wie Interferenzen oder Mehrwegeeffekte, bei der drahtlosen Kommunikation zu beobachten. Beim Mehrwegeempfang kommen die elektromagnetischen Wellen des Senders über unterschiedliche Pfade beim Empfänger an. Durch unterschiedliche Signallaufzeiten kann es zu Intersymbolinterferenzen kommen, aber auch bei einer deutlich höheren Signallaufzeit als die Symboldauer können Verschlechterungen der Signalstärke beobachtet werden, die je nach Funkumgebung über Wahrscheinlichkeitsverteilungen oder empirische Modelle modelliert werden können [36].

Aufgrund der Bedeutung von Leistungsmessungen für den systematischen Vergleich konkurrierender Technologien wurden verschiedene Test- und Verifizierungsansätze

entwickelt, die spezifische Merkmale von drahtlosen Netzwerken berücksichtigen und auf verschiedenen Testplattformen durchgeführt werden.

3.5.1. Abstraktionsebenen und vorhandene Lösungen

Die üblicherweise verwendeten Testplattformen für drahtlose Kommunikationslösungen basieren auf verschiedenen Abstraktionsebenen wie Simulation, emulierten Labortestumgebungen und Feldtests.

Diese drei Ebenen unterscheiden sich deutlich, sodass es nur Lösungen für jeweils eine der Ebenen gibt, aber keine einheitliche Testlösung. Eines der Ziele dieses Projektes war die Entwicklung einer einheitlichen Plattform zur Ausführung von Simulationen, Emulationen und Feldtests.

Nachfolgend werden die Ebenen sowie die existierenden Testplattformen auf den Ebenen beschrieben.

Simulationen Die Simulation kann als Nachahmung des Betriebs eines realen Systems im Laufe der Zeit definiert werden. Sie wird verwendet, um das Verhalten des Protokolls und seiner Implementierung, d.h. der Firmware über der Ebene der Hardware-Abstraktionsschicht, zu testen. Laut einer 2016 durchgeführten Studie über Simulationstools und Testumgebungen für drahtlose Sensornetze verwendeten etwa 53% der Autoren Simulationen in ihren Forschungsarbeiten, so dass Simulationen das Forschungswerkzeug der Wahl für die meisten Forschungsarbeiten im Bereich der drahtlosen Sensornetze sind.

Die Netzsimulatoren verwenden häufig mathematische Beschreibungen von Netzfunktionen, um Eingabedaten zu berechnen und das Verhalten und die Merkmale des Netzes zu erzeugen. Es gibt Simulatoren auf Systemebene, die für die grobe Bewertung der Netzwerkleistung hilfreich sind und sehr abstrakte Modelle verwenden, die Hardware, Netzwerke, drahtlose Kanäle oder zeitbezogene Verhaltensweisen nicht berücksichtigen. Auf der nächsten Ebene verwenden Netzwerksimulatoren die detaillierten Modelle des Netzwerkverhaltens und der Protokolle. Die Hardware und der drahtlose Kanal sind in diesen Simulatoren jedoch immer noch abstrahiert. Da es dem Simulationsansatz an Realitätsnähe mangelt (beim Testen von verteilten Netzwerklösungen), haben Testbeds in der Vergangenheit immer mehr Aufmerksamkeit erlangt

Ereignisgesteuerte Simulatoren sind die Plattform der Wahl für eine effiziente Ausführung von Szenarien mit Modellen von Kommunikationsknoten [37]. Die populärsten ereignisgesteuerten Netzwerksimulatoren sind das quelloffene NS-3 [38] und das OMNet++ [39] Netzwerksimulations-Framework sowie der kommerzielle Riverbed modeler [40] (früher OPNET modeler). Auf Berechnungen basierende MATLAB-Modelle werden ebenfalls häufig für Simulationen verwendet.

Emulationen Emulation ist die Nachahmung des Verhaltens bestimmten Systems, Geräts oder einer bestimmten Umgebung mit Hilfe eines anderen Systems. Die Emulation mit Testbeds besteht aus echter Hardware, Netzwerkkomponenten und Software. Wie

beim Simulationsansatz können Emulatoren vordefinierte emulierte Netzbedingungen und Verkehrsdynamiken für verschiedene Anwendungsfälle erzeugen. Die Hardwarebasierte Emulation erreicht eindeutig den größten Realismus auf der physikalischen Ebene. Praktische Erwägungen wie einfache Entwicklung, Kontrolle und Wiederholbarkeit von Experimenten haben dazu geführt, dass die Emulation mit Testbeds die vorherrschende experimentelle Technik ist. Einige der Emulatoren werden mit bestehenden Netzsimulatoren kombiniert, die über umfangreiche Modelle und Bibliotheken verfügen. Im folgenden Abschnitt wird ein Überblick über die vorhandenen Testbeds gegeben.

Feldtests Feldtests sind Tests in der Außen- und Innenumgebung, in der das reale Hardwaregerät verwendet wird. Obwohl sie bei der Bewertung der Zuverlässigkeit von drahtlosen Netzwerken eine wichtige Rolle spielen, haben sie auch Nachteile. Der größte Nachteil besteht in der Schwierigkeit, Tests aufgrund der unkontrollierbaren Natur der Testumgebung reproduzieren zu können. Darüber hinaus sind Feldtests zeitaufwändig und somit kostenintensiv.

Testplattformen Geeignete Testumgebungen sind für das Testen verteilter drahtloser Netzwerke unerlässlich und gewinnen im Zusammenhang mit IoT-Technologien immer mehr an Bedeutung.

In der Literatur finden sich viele Arbeiten, die sich mit der Analyse verschiedener Test- und Leistungsbewertungsplattformen für drahtlose Systeme befassen [41, 42]. H. Hellbrück et. al. [41] haben eine Zusammenfassung bestehender europäischer und einiger anderer populärer WSN-Testbeds zusammengestellt.

Verschiedene Testbeds, die für die Erforschung drahtloser Kommunikationstechnologien zur Verfügung stehen, sind darauf ausgerichtet, verschiedene Anforderungen wie Heterogenität, Skalierbarkeit, Portabilität, Verbund, Flexibilität, Mobilität, Interaktivität, Fehlersuche, Wiederverwendung von Software, Wiederholbarkeit und Gleichzeitigkeit zu erfüllen [42].

Kürzlich wurden in der Literatur viele Testumgebungen beschrieben. H. Hellbrück et. al [43] haben viele WSN-Testbeds wie DES-Testbed, FRONTS, Kansei, MIRAGE (Intel), Mote Lab, NetEye, Senslab, TutorNet, TWIST, VineLAB, w-iLab.t Testbed und Wisebed aufgelistet. Wisebed [43] ist ein großes, heterogenes WSN-Testbed, dessen Ressourcen über das Internet verfügbar sind, um gemeinsam mit anderen Testbeds Experimente durchzuführen. Diese Föderation von Testbeds basiert auf dem Konzept der Virtualisierung von Testbeds und virtuellen Verbindungen zwischen ihnen. Es bietet eine Vielzahl von Schnittstellen für Endbenutzer und Anwendungen. One Lab [44] ist ein Zusammenschluss von Testbed-Besitzern, der einen einfachen Zugang zu Testbeds ermöglicht und verschiedene IoT-Testbeds wie FIT IoT-Lab, III IoT-Lab, w-iLab.t unterstützt. FIT IoT Lab [45] ist ein wissenschaftliches Testbed und eine Erweiterung des Senslab-Testbeds. Es bietet eine sehr große Infrastruktureinrichtung, die für das Testen kleiner drahtloser Sensorgeräte und heterogener kommunizierender Objekte geeignet ist. Es verfügt über 2700 Knoten, die über sechs verschiedene Standorte in Frankreich

verteilt sind und unterschiedliche Prozessorarchitekturen und Funkchips unterstützen [45]. Santander testbed [46] ist eine IoT-Testeinrichtung für Smart-City-Anwendungsfälle mit einer dreistufigen Architektur aus IoT-Knoten, Repeatern und Gateways. Einige der anderen Prüfstände für drahtlose Systeme sind FIT UPMAC Lab, FIT r2Lab, NITOS, CoteXlab. Die Testumgebungen FUSECO, NITOS und FLEX (Fire LTE testbeds for open Experimentation) unterstützen ebenfalls das Testen von 4G/5G-Drahtlosnetzwerken und Industrie 4.0-Anwendungen. Die meisten der bestehenden Testbeds verwenden drahtlose Schnittstellen in Außen- und Innenumgebungen. Das ivESK-Team hat im Vorfeld des Projekts ein flexibles automatisiertes physikalisches Testbed (APTb) für drahtlose Netze entwickelt, das eine drahtgebundene Verbindung von HF-Elementen wie Dämpfungsgliedern, Verteilern, Schaltern und einer Mikrocontroller-Einheit mit Webserver zum Testen und Prüfen von drahtlosen Netzen verwendet [37]. Es ermöglicht die Variation der Pfadcharakteristik durch Veränderung der Dämpfungsglieder und Schalter. Der größte Vorteil von APTb ist die Möglichkeit einer isolierten, gut kontrollierten, wiederholbaren Testumgebung unter verschiedenen Bedingungen, um statistische Analysen und sogar Regressionstests durchzuführen. Allerdings fehlt es noch an einem nahtlosen und automatisierten Testablauf.

3.5.2. Testbeschreibungssprache

Test Description Language ist eine Sprache für die Spezifikation von Testbeschreibungen und die Darstellung von Testausführungsergebnissen. Es besteht eine methodische Lücke zwischen dem hochrangigen Ausdruck dessen, was getestet werden muss, d. h. den in Prosa oder in der Test Purpose Language (TPL) beschriebenen Testzwecken, und der komplexen Codierung der ausführbaren Tests in der Testing and Test Control Notation Version 3 (TTCN3) [47]. TDL bietet die Möglichkeit diese Lücke zu füllen.

TTCN3 ist eine Standard-Skriptsprache, die vom European Telecommunications Standard Institute (ETSI) entwickelt wurde. Die TTCN-3-Spezifikation wird üblicherweise für Konformitätstests von Kommunikationssystemen verwendet [47].

Specification and Description Language (SDL) ist eine Spezifikationssprache, die auf die eindeutige Spezifikation und Beschreibung des Verhaltens von reaktiven und verteilten Systemen abzielt. Sie wird von der ITU-T als ITU-Empfehlung Z.100 entwickelt und gepflegt. Aufgrund ihrer Eignung für Echtzeit-, Stimulus-Response-Systeme wurde SDL in der Telekommunikationsbranche ausgiebig genutzt.

3.5.3. TTCN-3 Test Automation Frameworks

Es gibt verschiedene Testautomatisierungs-Frameworks für eingebettete Systeme. Für SDWNs werden jedoch die TTCN-3-basierten Testautomatisierungs-Frameworks als geeignete Lösung angesehen. Die verfügbaren TTCN-3-basierten Testautomatisierungs-Frameworks sind Eclipse Titan [48] und Ttworkbench [49]

Obwohl ETSI die TTCN-3 Sprachsyntax formalisiert und vor mehr als einem Jahrzehnt öffentlich zugänglich gemacht hat, gibt es nur wenige Open-Source-Compiler, die kontinuierliche Wartung, Entwicklung und ergänzende Tools bieten können. Einer der besten

unter ihnen ist wahrscheinlich das bekannte Ericsson Titan TTCN Projekt, das ein Team von Entwicklern kürzlich auf github unter der Eclipse Public License (EPL) veröffentlicht hat.

TTworkbench ist eine IDE mit vollem Funktionsumfang für jede Art von Testautomatisierungsprojekt. Dieses leistungsstarke, benutzerfreundliche Werkzeug ermöglicht das Testen von Softwareprodukten und Dienstleistungen unabhängig von Technologie, Betriebssystem oder Implementierungsbereich. Basierend auf der standardisierten Testsprache TTCN-3 ermöglicht es sowohl Testautomatisierung als auch unbegrenzte Erweiterbarkeit.

4. Anwendungsfälle

Dieses Kapitel deckt Teile von AP 1 ab und beschreibt industrielle Anwendungsfälle für drahtlose Kommunikationstechnologien. Dieses Arbeitspaket wurde von Hahn-Schickard und Institut für eingebettete Systeme und Kommunikationstechnik (ivESK) in gleichen Teilen und in Abstimmung mit dem PA bearbeitet.

Die in diesem Projekt relevanten Technologien wurden im Kontext industrieller Anwendungen untersucht und bewertet. Es wurden daher unterschiedliche industrielle Anwendungsfälle betrachtet und deren Anforderungen an eine drahtlose Kommunikationslösungen dargestellt. Über Testkampagnen wurde die Erfüllbarkeit der Anwendungsfälle durch die Technologien überprüft. Das im Projekt entwickelte Testsystem ist aber nicht auf das Testen dieser spezifischen Anwendungsfälle beschränkt. Die Anwendungsfälle dienen vielmehr als repräsentative Anwendungen aus unterschiedlichen Bereichen der industriellen Automatisierungstechnik und sollen sowohl die Tauglichkeit der Technologien für konkrete Beispiele untersuchen und validieren, aber auch die Anwendbarkeit des entwickelten Testaufbaus darstellen.

Die ausgewählten Anwendungsfälle orientieren sich an typischen industriellen Anwendungsszenarien im Bereich der industriellen Automatisierung. Hierbei wurde für dieses Projekt zwischen den Anwendungsbereichen der Fabrikautomation, der Prozessautomation und der Maschine-zu-Maschine (M2M) Kommunikation unterschieden. Diese Anwendungsbereiche unterscheiden sich deutlich in ihren Anforderungen, insbesondere bezüglich Latenz, Gerätedichte und Umgebungsbedingungen. In den nachfolgenden Abschnitten wird auf die einzelnen Bereiche genauer eingegangen.

Zur Identifizierung und Zuordnung der Anwendungsfälle in diese Anwendungsbereiche wurden unterschiedliche Studien und Projekte betrachtet. In [50] [51] wurden mehrere industrielle Anwendungsfälle für drahtlose Systeme identifiziert und spezifiziert. Darüber hinaus wurden im Taktilus-Projekt weitere Anwendungsfälle in der Fabrik- und Prozessautomation definiert [52].

Insgesamt wurden im Projekt zwei Anwendungsfälle jeweils in den Bereichen der Fabrik- und Prozessautomation sowie der M2M Kommunikation ausgewählt und charakterisiert. Jedes Szenario beschreibt die Prozesse und Interaktionen industrieller Automatisierungsgeräte (z.B. Sensoren, Aktoren, SPS) und wie Daten zwischen ihnen übertragen werden. Darüber hinaus wird die Umgebung, in der die Systeme betrieben werden, beschrieben.

In diesem Abschnitt werden diese Anwendungsfälle vorgestellt, charakterisiert und schließlich deren Anforderungen an die drahtlosen Kommunikationslösungen verglichen.

4.1. Charakterisierung von drahtlosen Kommunikationsanwendungen

Für drahtlose Systeme ist es von entscheidender Bedeutung, nicht nur das Kommunikationsmuster einer Anwendung zu charakterisieren, sondern auch die Umgebung, in der ein drahtloses System eingesetzt werden soll.

Anwendungen im IoT-Bereich folgen bestimmten Kommunikationsmustern. Eine Übersicht zu gängigen Mustern ist in Tab. 4 dargestellt.

In diesem Projekt spielten Gerätezustände eine wichtigere Rolle als die Unterscheidung zwischen periodischer oder aperiodischer Datenübertragung. Entsprechend könnten für dieses Projekt die aperiodischen und periodische Kommunikationsarten (vgl. Tab. 4) zusammengefasst werden. Es wurden stattdessen bestimmte Gerätezustände betrachtet, aus denen heraus eine Datenübertragung getriggert werden kann. Dies kann durch einen geeigneten Testaufbau bewerkstelligt werden, beispielsweise durch geschicktes Einstellen der Übertragungsintervalle. Zudem sind die kombinierten Muster als Zusammensetzung der einzelnen Kommunikationsrichtungen zu verstehen.

Tabelle 4: Übersicht zu relevanten Kommunikationsmustern

Name	Szenario	Relevanz
Aperiodischer Uplink (AU)	IoT-Geräte senden häufig Daten aufgrund von Ereignissen. Im Vergleich zur periodischen Übertragung kann sich das Funkmodem in verschiedenen Zuständen befinden.	Auswirkung verschiedener ED/BS-Zustände, Übertragungszeit und Energiebedarf
Periodischer Uplink (PU)	Ein IoT-Gerät sendet Daten in einem periodischen Intervall. Dies wird häufig in Systemen mit höheren Sicherheitsanforderungen verwendet.	Übertragungszeit, Aktualisierungszeit, Durchsatz, Energiebedarf
Burst Uplink (BU)	IoT-Geräte aggregieren häufig Daten über einen längeren Zeitraum, um Ressourcen in Bezug auf Energie und Frequenznutzung zu sparen.	Auswirkungen der Datenaggregation, Vergleich des Energiebedarfs mit einer einzelnen Datenübertragung, Durchsatz, Übertragungszeit
Aperiodischer Downlink (AD)	Aperiodische Downlink-Nachrichten können zur Aktualisierung des Zustands eines IoT-Geräts verwendet werden.	Auswirkungen von Downlink-Nachrichten: Übertragungszeit, aktueller Bedarf
Uplink triggert Downlink (UTD)	Ein IoT-Gerät sendet in einem regelmäßigen Intervall Daten, und die BS antwortet sofort, z. B. um die Daten zu bestätigen, einen angeforderten Wert oder andere Daten zurückzusenden. Dies ist häufig erforderlich und nützlich, da viele Technologien kurz nach dem Senden in einen Ruhezustand übergehen.	Auswirkungen des Downlink-Fensters, Reaktionszeit, Energiebedarf
Downlink triggert Uplink (DTU)	Eine Anwendung könnte Geräteparameter ändern und eine Statusaktualisierung oder eine Bestätigung durch das IoT-Gerät erwarten.	Auswirkungen auf Downlink-ACK, Reaktionszeit, Energiebedarf
Firmware Update (FU)	IoT-Sensoren werden oft über viele Jahre hinweg verwendet. Manuelle Aktualisierungen sind zeit- und kostenaufwändig. Daher sind Aktualisierungen über die Luft ein wichtiger Anwendungsfall.	Auswirkungen einer Aktualisierung, Zeit, Durchsatz, Energiebedarf

Die Ausbreitung von elektromagnetischen Wellen hängt von verschiedenen Parametern ab, die eine genaue Modellierung der Umgebung erfordern, um sie vollständig zu beschreiben. Es gibt jedoch einige wesentliche Effekte bei der drahtlosen Ausbreitung, die durch Dämpfung, Interferenz, Mehrwegeausbreitung und Mehrwegeschwund verursacht werden. Diese Effekte entstehen aufgrund von Störungen zwischen drahtlosen Geräten, den Eigenschaften und Dimensionen der Umgebung sowie der Größe des Raums und anderen Faktoren.

Für die Kategorisierung der Umgebungsbedingungen wird das folgende, in [2] vorgeschlagene Schema verwendet:

- 0 - Sichtverbindung bzw. Line of Sight (LOS)
- 1 - leichte Beeinträchtigung der Sichtverbindung (Personen, Bäume, Säulen, Balken in einer Fabrik,...)
- 2 - mittlere bis vollständige Beeinträchtigung der Sichtverbindung (Non Line of Sight (NLOS)), z. B. in einer Produktionszelle oder Roboterzelle
- 3 - keine Sichtverbindung; sehr hoher Metallanteil, z. B. in einer verfahrenstechnischen Anwendung

4.2. Fabrikautomation

Die Fabrikautomation ist eine Disziplin innerhalb der Automatisierungstechnik, die sich mit der Automatisierung von Fabriken durch technische Automaten beschäftigt. Im Mittelpunkt stehen die Prozesskette der automatisierten Produktion und die dafür notwendigen Technologien wie Montage- und Handhabungstechnik, Robotik, Bildverarbeitung, Identifikationssysteme, Sensoren und Aktoren, Antriebstechnik, Steuerungstechnik, zugehörige Software, Hydraulik und Pneumatik sowie relevante Sicherheitssysteme.

Anwendungen aus der Fabrikautomation zeichnen sich durch eine hohe Dichte von Knoten in einer Produktionsanlage (Produktionszelle) aus, während die Gesamtzahl der Geräte relativ gering ist. Gleichzeitig sind sehr schnelle Reaktionen (bis zu 10 μ s oder sogar 1 μ s) erforderlich. Die jeweilige Datenmenge und die logische Tiefe der Verbindungen sind eher gering.

4.2.1. FA1 - Panel zur Robotersteuerung

Eines der Anwendungsbeispiele innerhalb des Projekts, welches aus dem Projekt Taktillus übernommen wurde [52], definiert und beschreibt ein Roboterbedienfeld, das die Daten des Sicherheitsschalters „Not-Halt- und Freigabeschalter“ an die Sicherheitssteuerung überträgt. Die Robotersteuerung ist ein physischer und beispielhafter Platzhalter für ein beliebiges IoT-Automatisierungsgerät. Die Sicherheitssteuerung könnte auch eine Standardsteuerung oder eine andere Systemeinheit mit Ethernet-Schnittstelle darstellen. Ein Bedienpanel kann bis zu zehn Sicherheitssteuerungen verwalten. Das IoT-Gerät kann in einem Polling-Modus arbeiten, in dem das Gerät auf Anfragen einer Steuerung

antwortet, und in einem Publish-Modus, in dem das IoT-Gerät in einem konfigurierbaren Zeitintervall sendet. Eine Sicherheits-Steuerung kann zwei oder mehr Verbindungen zu zwei oder mehr Roboter-Panels haben, wenn sie sich alle immer in der Nähe des Roboters befinden.

Die Charakterisierung dieses Anwendungsfalls ist in Tab. 5 zusammengefasst.

Tabelle 5: Charakterisierung von FA1

Kommunikations-Muster	<ul style="list-style-type: none"> • Aperiodischer Downlink • Periodischer Uplink
Umgebung	<ul style="list-style-type: none"> • Einsatzumgebung: Abgeschlossene Fabrikumgebung (50x50m) • Funkstrecke: LOS mit leichten Einschränkungen • Dynamik: Leicht variierende Kanalbedingungen (aufgrund der Laufgeschwindigkeit) • Aktive Einflüsse: kaum Interferenz zu erwarten
Relevante Technologien	<ul style="list-style-type: none"> • LTE

4.2.2. FA2 - Produktionszelle

Dieses Anwendungsprofil beschreibt die Anforderungen und Bedingungen, die an Maschinen oder an diskrete Fertigungsprozesse in Produktionszellen gestellt werden. Die Prozesse folgen einem Zyklus, der letztlich auch den Produktausstoß bestimmt. Deshalb muss das drahtlose Kommunikationssystem das Kommunikationsvolumen dieses Zyklus zuverlässig bewältigen können. Die zu übertragenden Mess- und Steuerungsdaten sind kurz. Die Anzahl der drahtlosen Geräte und damit der logischen Verbindungen ist überschaubar. Mobile drahtlose Geräte sind z.B. als Sensoren möglich. Die Ausdehnung des Netzes entspricht der Größe der Maschine. [50]

Die Charakterisierung dieses Anwendungsfalls ist in Tab. 6 zusammengefasst.

Tabelle 6: Charakterisierung von FA2

Kommunikations-Muster	<ul style="list-style-type: none"> • Periodischer Uplink • Periodischer Uplink triggert Downlink
Umgebung	<ul style="list-style-type: none"> • Einsatzumgebung: Abgeschlossene Fabrikumgebung (200x300m) • Funkstrecke: LOS mit leichten bis schweren Sichteinschränkungen • Dynamik: Variierende Kanalbedingungen aufgrund sich bewegender Sensorik/Aktorik • Aktive Einflüsse: kaum Interferenz zu erwarten
Relevante Technologien	<ul style="list-style-type: none"> • LTE

4.3. Prozessautomation

Prozessautomation bezieht sich auf chemische und verfahrenstechnische Prozesse. Anwendungen der Prozessautomation finden sich z.B. in der Öl- und Gasförderung, in der Stromerzeugung oder in Schmelzbetrieben wie Eisen-, Stahl- und Tempergießereien. Häufig werden verschiedenste Sensoren (z.B. zur Überwachung der Temperatur oder der Durchflussmenge) und entsprechende Aktoren (z.B. Ventile) eingesetzt. Funkkommunikation ist eine interessante Lösung, da teilweise raue Umgebungsbedingungen und große Entfernungen von bis zu mehreren Kilometern überbrückt werden müssen. Je nach Geschwindigkeit der zu steuernden Prozesse bestehen typischerweise moderate Anforderungen an Latenzzeit und Zuverlässigkeit.

Anwendungen der Prozessautomation sind räumlich weit verteilt, möglicherweise über mehrere Quadratkilometer einer chemischen oder biologischen Prozessanlage. Die Reaktionen sind weniger zeitkritisch, aber von größerer logischer Tiefe als in der Fabrikautomation, da z.B. manche Gefahrensituationen erst in der Kombination vieler Messwerte erkannt werden können. Dies impliziert auch eine hohe Anzahl von räumlich verteilten Endgeräten. Auch die Datenmenge pro Sensor ist in der Regel gering, kann aber in der Kumulation erheblich werden.

4.3.1. PA1 - Kläranlage

In Kläranlagen werden schon seit langem drahtlose Lösungen eingesetzt. Die meisten Kläranlagen erstrecken sich über weite Flächen mit Vorklärstrecken, Rund- und Längsräume. Diese physikalische Infrastruktur wurde erst in den letzten Jahren mit Automatisierung aufgerüstet. In den meisten Fällen wurden beim Bau der Anlagen nur Stromleitungen, aber keine Datenleitungen installiert. Hier sind Funklösungen oft die

einzigste nachträglich realisierbare Möglichkeit der Datenvernetzung, da andere Übertragungstechnologien wie Datenlichtschranken, Kabeltrommeln oder Schleifringe konzeptbedingte Nachteile aufweisen. So haben sich in der letzten Klärungsstufe drahtlose Anwendungen für den Einbau in rotierende Rundsauger gegenüber korrosionsanfälligen Schleifringen durchgesetzt, auch weil es kaum möglich ist, eine Datenleitung an der vertikalen Hauptwelle eines Rundsaugers nachzurüsten. Heutzutage werden Schlammspiegelmessgeräte mit einem Feldbusanschluss durchgängig auf Kesselbrücken installiert. Die Datenanbindung an die zentrale Leitwarte der Kläranlage wird über PROFIBUS und PROFINET mittels einer Funkübertragung an die rotierende Eimerbrücke realisiert. Aus Sicht der Anwendung sind die Anforderungen an die Übertragungsstrecke gering. Die Übertragungszeiten können im Bereich von einer Sekunde liegen. [50]

Die Charakterisierung dieses Anwendungsfalls ist in Tab. 7 zusammengefasst.

Tabelle 7: Charakterisierung von PA1

Kommunikations-Muster	<ul style="list-style-type: none"> • Aperiodischer Uplink (APU) • Periodischer Downlink (PD) • DTU • FU
Umgebung	<ul style="list-style-type: none"> • Einsatzumgebung: Außenanlage (200x200m) • Funkstrecke: mittlere bis schweren Sichteinschränkungen (2) • Dynamik: Statischer Kanal • Aktive Einflüsse: kaum Interferenz zu erwarten • NB-IoT
Relevante Technologien	<ul style="list-style-type: none"> • LoRa (LoRa) • MIOTY

4.3.2. PA2 - Zustandsüberwachung einer Prozessanlage

Sensoren und Aktoren sind die Basis jeder automatisierten Produktion. Ihre korrekte und präzise Funktion ist entscheidend für den wirtschaftlichen Betrieb einer Anlage. Ausfälle können zu Produktionsstopps führen, Kosten verursachen und unter Umständen zu körperlichen Schäden führen. Um den Zustand der Prozessanlage zu erfassen, muss eine große Anzahl von Sensoren und Aktoren in ein Netzwerk eingebunden werden, wobei hohe Anforderungen an die Verfügbarkeit und Zuverlässigkeit der Übertragung gestellt werden. Die Charakterisierung dieses Anwendungsfalls ist in Tab. 8 zusammengefasst.

Tabelle 8: Charakterisierung von PA2

Kommunikations-Muster	<ul style="list-style-type: none"> • Aperiodischer Uplink • Periodischer Uplink • Downlink triggert Uplink • Firmwareupdate
Umgebung	<ul style="list-style-type: none"> • Einsatzumgebung: Außenanlage (1000x500m) • Funkstrecke: LOS mit leichten bis schweren Sichteinschränkungen bis hin zu NLOS (0-3) • Dynamik: Statischer Kanal • Aktive Einflüsse: Hohes Potential an Interferenz zu erwarten aufgrund vieler Geräte in der Umgebung sowie metallischen Gegenständen
Relevante Technologien	<ul style="list-style-type: none"> • NB-IoT • LoRa • MIOTY

4.4. Maschine-zu-Maschine Kommunikation

Anwendungen der klassischen M2M-Kommunikation, z.B. zur Infrastrukturüberwachung, sind räumlich am weitesten verteilt. Hier kann es vorkommen, dass nur ein einziger Knoten in einem System installiert ist. Die Skalierbarkeit für diesen Anwendungsbereich wird jedoch als sehr hoch eingeschätzt, da mehrere tausend Geräte angeschlossen werden können.

4.4.1. MM1 - Fahrerlose Transportsysteme

Fahrerlose Transportsysteme (FTS) als Teil Automatisierte Transportsysteme (ATS) werden im Bereich der industriellen Automatisierung zum Transport von Lasten und Werkstücken auf vorher festgelegten Routen eingesetzt. Die Strecken sind in Segmente unterteilt, die durch Bodenmarkierungen gekennzeichnet sind, die vom FTS identifiziert werden können. Eine übergeordnete Fahrzeugsteuerung empfängt Transportaufträge von einer Anlagensteuerung und generiert daraus Fahrbefehle für die einzelnen Fahrzeuge. Diese werden lokal im Fahrzeug durch eine Fahrzeugsteuerung umgesetzt. Um die entsprechenden Fahrbefehle an die jeweiligen Fahrzeuge zu senden, muss die übergeordnete Steuerung über ein stets aktuelles Prozessabbild verfügen. Mit den Informationen über das Prozessabbild wird auch der Verkehr geregelt, um Kollisionen

und Blockaden zu vermeiden. Das Prozessabbild wird ereignisorientiert aktualisiert, indem jedes Fahrzeug bei Erreichen einer Etagenmarkierung eine Statusmeldung sendet. Zusätzlich werden bei längeren ereignislosen Fahrten oder bei längerem Stillstand Statusmeldungen z. B. im Minutentakt an die übergeordnete Steuerung gesendet. In der Regel erhält das Fahrzeug als Antwort auf seine Übertragung einen aktuellen Fahrbefehl. Da die zulässigen Fahrgeschwindigkeiten gering und die Strecken teilweise lang sind, erübrigt sich eine periodische Übertragung. Die zu übertragenden Fahrbefehle und Statusmeldungen umfassen typischerweise nur wenige Bytes, z.B. acht Bytes. Andererseits kann es eine große Anzahl von Fahrzeugen geben. Die aperiodischen Kommunikationsanforderungen von nicht synchronisierten logischen Verbindungen können zu Kollisionen bei der drahtlosen Kommunikation führen. Diese müssen innerhalb bestimmter Zeiten aufgelöst werden.[50]

Die Charakterisierung dieses Anwendungsfalls ist in Tab. 9 zusammengefasst.

Tabelle 9: Charakterisierung von MM1

Kommunikations-Muster	<ul style="list-style-type: none"> • Uplink triggert Downlink (wenn FTS Bodenmarkierung erreicht) • Periodischer Uplink (wenn längere Zeit keine Marker) • Downlink triggert Uplink (Kommandos von der zentralen Steuerung)
Umgebung	<ul style="list-style-type: none"> • Einsatzumgebung: geschlossene Innenanlage (1000x500m), Außenanlage (3000x2000m) • Funkstrecke: LOS mit leichten bis schweren Sichteinschränkungen bis hin zu NLOS (0-3) • Dynamik: Deutliche Änderungen des Funkkanals aufgrund der Bewegung • Aktive Einflüsse: Hohes Potential an Interferenz von anderen Geräten und der Umgebung
Relevante Technologien	<ul style="list-style-type: none"> • LTE • NB-IoT • LoRa • MIOTY

4.4.2. MM2 - Warenverfolgung

Drahtlose Kommunikationslösungen können eingesetzt werden, um eine lückenlose Warenverfolgung in der Logistik zu ermöglichen. Durch die Verwendung von drahtlosen Sensoren und Gateways können Warenströme in Echtzeit überwacht und Engpässe schnell identifiziert werden, was eine effizientere Lieferkette und höhere Transparenz ermöglicht. Die Charakterisierung dieses Anwendungsfalls ist in Tab. 10 zusammengefasst.

Tabelle 10: Charakterisierung von MM2

Kommunikations-Muster	<ul style="list-style-type: none"> • Aperiodischer Uplink (Events) • Periodischer Uplink (Status updates) • Downlink triggert Uplink (Kommandos von der zentralen Steuerung) • Firmwareupdate
Umgebung	<ul style="list-style-type: none"> • Einsatzumgebung: Außenanlage (10000x10000m) • Funkstrecke: LOS mit leichten bis schweren Sichteinschränkungen bis hin zu NLOS (0-3) • Dynamik: Deutliche Änderungen des Funkkanals aufgrund der Bewegung möglich • Aktive Einflüsse: Hohes Potential an Interferenz von anderen Geräten und der Umgebung
Relevante Technologien	<ul style="list-style-type: none"> • NB-IoT • LoRa • MIOTY

5. Anforderungserhebung

Dieses Kapitel deckt Teile von AP 1 ab und beschreibt die Anforderungen an die unterschiedlichen Teilsysteme. Dieses Arbeitspaket wurde von Hahn-Schickard und ivESK in gleichen Teilen und in enger Abstimmung bearbeitet, indem die unterschiedlichen Teilkomponenten identifiziert und deren Anforderungen aufgestellt wurden.

Die zum Projektzeitpunkt existierenden Testlösungen erlaubten es nicht (vgl. Abschnitt 3.5), technologieunabhängige Tests in einer einheitlichen Umgebung durchzuführen. Die meisten Testplattformen haben eine ähnliche zugrunde liegende Architektur, aber die Methoden zur Beschreibung, Konfiguration, Ausführung, Überwachung und

Steuerung von Tests sind unterschiedlich.

Im Rahmen dieses Projekts wurde deshalb eine einheitliche Testumgebung entworfen, die es ermöglicht, auf allen Ebenen dieselben Testfälle, Testszenarien und Testparameter zu verwenden. Zudem bietet sie einen einheitlichen Testausführungsprozess sowie Testverfahren und Methoden zur Visualisierung der Ergebnisse an.

Die Anforderungen an die einzelnen Teilsysteme bestehend aus zentraler Teststeuerung, Feldtestumgebung, Emulator und Simulator sind nachfolgend dargestellt.

5.1. Drahtlose Kommunikationslösungen

In Abschnitt 4 wurden die ausgewählten Anwendungsfälle beschrieben und charakterisiert. Diese stehen repräsentativ für bestimmte Anwendungsgebiete der industriellen Automatisierungstechnik. Das entwickelte Testsystem ist aber nicht auf das Testen dieser Anwendungsfälle beschränkt. Vielmehr dienen die Anwendungsfälle in diesem Projekt als Anhaltspunkt und sollen die Anwendbarkeit des Testsystems darstellen sowie Technologien in einem Anwendungsfall-spezifischen Kontext betrachten.

Die Anwendungsfälle stellen unterschiedliche Herausforderungen an drahtlose Kommunikationslösungen. Die Anforderungen der einzelnen Anwendungsfälle an die Systemparameter sind in Tab. 11 dargestellt. Hierfür werden die Kommunikationsmuster aus Tab. 4 in Abschnitt 4 referenziert. Die Tabelle basiert auf Anwendungsfällen, die in VDE-Studien ermittelt wurden [50] [51], sowie auf Anwendungsfällen des Taktilus-Projektes [52].

Tabelle 11: Anforderungen der Anwendungsfälle and die Kommunikationslösungen

Systemparameter	FA1	FA2	PA1	PA2	MM1	MM2
Latenz	10 ms	100 ms	1000 ms	10 000 ms	100–300 ms	60 000 ms
Fehlerrate	1×10^{-6}	1×10^{-6}	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-4}
Paketgröße	16–32 B	2 B	4 B	<5000 B	250 B	64 B
Periodizität	5–20 ms ⁻¹	10 ms ⁻¹	500 ms ⁻¹	1–24 h ⁻¹	500 ms ⁻¹	1–24 h ⁻¹
Reichweite	5–50 m	10 m	<200 m	1000 m	500–3000 m	<5000 m
Verbundene Geräte	1–10	1–10	10	>1000	20	>1000
Einsatzumgebung	innen, (0-1)	innen (1-2)	außen (0-2)	außen (0-3)	außen (0-3)	außen (0-3)
Mobilität	<1 m/s	<5 m/s	0 m/s	0 m/s	2.5 m/s	3–30 m/s
Batterielebensdauer	unwichtig	unwichtig	> 1 yr	10 yr	1 d	10 yr
Kommunikations-Muster	PU AD	PU P. UTD	APU PD DTU FU	APU PD BU DTU FU	PU A. UTD A. DTU	AU FU A. PU A. DTU

5.2. Testsysteme

Ein Hauptziel dieses Projektes lag in der Umsetzung einer einheitlichen Testumgebung, die es erlaubt Tests auf unterschiedlichen Abstraktionsebenen durchzuführen. Zu diesem Zweck wurden Komponenten entwickelt, die eine Testausführung auf Simulations-, Emulation- und Feldtestebene ermöglichen. Die zentrale Teststeuerung wird durch ein zentrales Testsystem durchgeführt. Die Anforderungen an die jeweiligen Teilsysteme und Ebenen sind im Nachfolgenden dargestellt.

5.2.1. Zentrale Teststeuerung

Die Zentrale Teststeuerung soll verwendet werden, um Testfälle zu beschreiben und diese auf den unterschiedlichen Abstraktionsebenen auszuführen. Sie soll Nachrichten mit einem Device Under Test (DUT) austauschen können, mit dem in diesem Projekt entwickelten Netzwerkemulator interagieren können sowie Simulationen steuern können. Im Vorfeld der Entwicklung des TTCN3 basierten Testsystems wurden hierfür die in Tab. 12 beschriebenen Anforderungen aufgestellt.

Tabelle 12: Anforderungen an das zentrale Testsystem

ID	Anforderung
TS-RF1	Das Testsystem muss eine gemeinsame Schnittstelle für verschiedene Geräte verwenden.
TS-RF2	Das Testsystem sollte nicht von einer bestimmten Hardware-Schnittstelle abhängig sein.
TS-RF3	Das Testsystem muss in der Lage sein, Simulationen auf einem NS-3-Server zu starten, APTB-Netze zu steuern und mit den zu testenden Implementierungen zu arbeiten.
TS-RF4	Das Testsystem muss ein Verfahren zur Beschreibung und Parametrierung von Netzen auf den verschiedenen Ebenen festlegen.
TS-RF5	Das Testsystem sollte verschiedene Messgeräte unterstützen.
TS-RF6	Das Testsystem muss die gleichzeitige Ausführung des Testverhaltens für verschiedene Knoten unterstützen.
TS-RF7	Verschiedene Testausführungsprozesse müssen aus der Haupttestbeschreibung heraus steuerbar sein.
TS-RF8	Tests für ein bestimmtes Konzept müssen innerhalb ihres Anwendungsbereichs maximal parametrierbar sein.
TS-RF9	Es muss möglich sein, Tests über ein Ethernet-Netzwerk durchzuführen.
TS-RF10	Daten müssen in strukturierter Form erfasst und gespeichert werden.
TS-RF11	Es müssen Events zur strukturierten Datenspeicherung verwendet werden.
TS-RN1	Es muss nicht notwendig sein, die Details des Frameworks zu verstehen, um einen Test zu lesen.
TS-RN2	Das Hinzufügen neuer Geräte darf keine Änderung der Testimplementierung erfordern, da das gleiche Testverhalten verwendet wird.
TS-RN3	Neue Tests müssen einfach zu implementieren sein, ohne dass eine Menge zusätzlicher Code angegeben werden muss.
TS-RN4	Das Testsystem sollte einfach auf einer Client-Plattform zu installieren sein.
TS-RN5	Der Arbeitsablauf beim Schreiben von Tests muss klar definiert sein. Der Rahmen muss einen Weg zur Beschreibung der Tests definieren.
TS-RN6	Der Arbeitsablauf bei der Ausführung von Tests muss leicht verständlich sein, ohne dass man Details des Testsystems verstehen muss.
TS-RN7	Es muss möglich sein, mehrere Testfälle nacheinander oder mehrfach auszuführen.
TS-RN8	Nach der Durchführung eines Tests müssen die Ergebnisse und die Testkonfiguration automatisch in einem zentralen Speicher abgelegt werden.

5.2.2. Feldtests

Feldtests sollen eine Untersuchung von Funktechnologien unter realen Bedingungen erlauben. Hierfür sollen Tests in beliebigen Umgebungen ausführbar sein. In diesem Fall ist es häufig notwendig, Tests auf unterschiedlichen Steuer-PCs durchzuführen und diese zeitlich zu synchronisieren. Die Anforderungen, die sich an die Feldtestdurchführung ergeben, sind in Tab. 13 dargestellt.

Tabelle 13: Anforderungen an Feldtests

ID	Anforderung
FT-RF1	Das Testsystem muss Möglichkeiten zur Synchronisation von Zeitstempeln mit einem zentralen Server anbieten.
FT-RF2	Tests müssen parametrierbar für Sender und Empfänger sein, sodass die Testausführung individuell und auf mehreren PCs durchgeführt werden kann.
FT-RF3	Tests müssen Timeouts berücksichtigen, sodass Tests immer zu Ende laufen, aber auch genug Zeit vorhanden ist um Tests auf unterschiedlichen Controllern zu starten.
FT-RF4	Testdaten sollen nach einem Tests automatisch zusammengeführt und ausgewertet werden können.

5.2.3. Emulator

Am ivESK wird bereits ein Emulator für drahtlose Kommunikationstechnik eingesetzt. Um die Anforderungen des Projektes erfüllen zu können, sind einige Anpassungen und Erweiterungen notwendig. Die Anforderungen an den neuen Netzwerksimulator sind in Tab. 14 zusammengefasst.

Tabelle 14: Anforderungen an den Netzwerkemulator

ID	Anforderung
EM-RF1	Der Emulator muss es ermöglichen verschiedene Topologien aufzubauen.
EM-RF2	Es muss möglich sein, die Netzwerkkonfiguration über ein Human Machine Interface (HMI) zu bedienen.
EM-RF3	Es muss unabhängig von der verwendeten Technologie sein.
EM-RF4	Einzelne HF-Elemente innerhalb des Aufbaus müssen sich automatisch registrieren können.
EM-RF5	Es muss eine API bereitgestellt werden, um HF-Elemente automatisiert steuern zu können.
EM-RF6	Der Emulator soll um eine modulare Komponente zur Emulation von Mehrwegekomponenten erweitert werden.
EM-RF7	Es muss um zusätzliche modulare Verzögerungsglieder erweiterbar sein.
EM-RN1	Tests müssen reproduzierbar durchgeführt werden können.
EM-RN2	Der Aufbau muss einfach zu ändern sein.
EM-RN3	Der Aufbau muss flexibel sein, um einfach Erweiterungen durchführen zu können.
EM-RN4	Der Hardwareaufbau soll mit wenig Aufwand möglich sein, um den Emulator auch an anderen Stellen einfach aufbauen zu können.
EM-RN5	Der Aufbau muss modular sein, um für verschiedene Zwecke mit unterschiedlichem Hardwareaufwand eingesetzt werden zu können.

5.2.4. Simulator

Die Anforderungen an den Simulator sind in Tabelle 15 zusammengefasst.

Tabelle 15: Anforderungen an den Simulator

ID	Anforderung
SI-RF1	Der Simulator muss durch ein Ausführungsskript startbar sein.
SI-RF2	Das Ausführungsskript muss vom Testsystem aufrufbar sein.
SI-RF3	Der Simulator muss das Zusammenspiel mehrerer Geräte simulieren können.
SI-RF4	Der Simulator muss seine Ergebnisse an das Testsystem weiterleiten.

6. Aufbau Simulationsumgebung

Dieses Kapitel deckt die Arbeitsschritte von AP 2 ab. Das Ziel des APs war es, unterschiedliche Funktechnologien mittels Simulationen testen zu können und diese Simulationen in die Umgebung für TestKomm 4.0 einzubinden. Dieses Arbeitspaket wurde maßgeblich von Hahn-Schickard bearbeitet, während ivESK abstimmend mitwirkte, um die Umgebung gut in das Gesamtsystem integrieren zu können.

Dafür werden die einzelnen Schritte bis zur funktionierenden Einbindung in die Umgebung für TestKomm 4.0 erläutert.

Zunächst wird die Simulationsumgebung vorgestellt. Anschließend wird die Beschreibungssprache gezeigt, die von anderen Quellen verwendet werden kann um Tests zu beschreiben, welche die Simulationsumgebung verwenden. Die Testbeschreibung kann dann in der Umgebung für TestKomm 4.0 verwendet werden. Zum Schluss wird eine Webanwendung vorgestellt, welche ebenfalls die Beschreibungssprache für die Ausführung von Simulationen verwendet.

Abschnitt 8 stellt mit dieser Simulationsumgebung erzeugte Ergebnisse vor.

6.1. Simulationsumgebung

Als Simulationsframework wurde ns-3¹ gewählt. Dies ist ein Diskrete-Events-Simulator für Internet-Netzwerke. Es ist ein Open-Source Projekt, welches die Lizenz GNU GPLv2 verwendet. Eine Alternative wäre OMNeT++² gewesen, jedoch existieren wichtige Funktechnologien wie NB-IoT nur für ns-3 und ns-3 lässt sich besser in eine externe Umgebung einbinden.

Für die Entwicklung der Simulationen wurden hauptsächlich jeweils aktuelle Versionen von ns-3 verwendet. Da ns-3 jedoch nicht alle benötigten Funktechnologien beinhaltet, wurden zusätzlich externe Projekte verwendet. Das Projekt NB-IoT³ ist eine der ersten Implementierungen, die auch im offiziellen ns-3 Wiki aufgeführt wird [53]. Es implementiert den NB-IoT Standard aber nicht voll umfänglich und wird für Untersuchungen meist durch spezifische Erweiterungen ergänzt, wie in [54]. Das Projekt LENA-NB⁴ implementiert aktuelle Features von NB-IoT umfangreicher und wurde in unterschiedlichen kürzlich erschienenen wissenschaftlichen Publikationen verwendet [55, 56]. Für die betrachteten Szenarien war dieses Modell allerdings noch nicht ausreichend ausgereift, sodass für die Evaluierung auf das NB-IoT Projekt zurückgegriffen wurde.

Das Ziel der Simulation ist, durch verschiedene Messungen Aussagen über unterschiedliche Funktechnologien in verschiedenen Anwendungsfällen zu treffen. Wichtige durchgeführte Messungen sind Paketverlust, Datenübertragungsrate oder die Latenz vom Sender zum Empfänger. Die Auswertung dieser Messungen erfolgt in der in AP 5 entwickelten Software (siehe Abschnitt 8).

¹www.nsnam.org

²www.omnetpp.org

³<https://github.com/imec-idlab/NB-IoT>

⁴<https://github.com/tudo-cni/ns3-lena-nb>

Um alle diese Funktechnologien fehlerfrei zu simulieren und miteinander zu vergleichen, wurde während des Projekts TestKomm 4.0 ein ns-3-Softwarepaket entwickelt, welches möglichst viele Bestandteile einer Simulation verallgemeinert. Dieses Softwarepaket bietet die folgenden Funktionen an.

- UDP-Applikation um Pakete zu versenden und zu überwachen
- unterschiedliche Kommunikationsmuster (siehe Tabelle 4)
- nach Möglichkeit einheitliche Simulationsparameter trotz unterschiedlicher Funktechnologien
- Positionierung der einzelnen Kunden im Raum
- Messung unterschiedlicher Netzwerkereignisse
- Auswertung der Messungen
- Abspeichern der Messungen in Dateien

Jede in TestKomm 4.0 entwickelte Simulation verwendet dieses Softwarepaket. So wurde nicht nur die Code-Menge deutlich reduziert, sondern auch ein schnelles Einbinden neuer Funktechnologien ermöglicht. Desweiteren können Fehler bei neuen Simulationen vermieden werden, da das Softwarepaket bereits getestet ist.

Das Verhalten und der Aufbau des Netzwerkes lässt sich stark parametrieren, daher wird im folgenden das Netzwerk nur beispielhaft dargestellt. Abbildung 5 zeigt den Aufbau des simulierten Netzwerkes. Das Netzwerk besteht aus einer Vielzahl von Kunden (Clients), einem Server und einem Netzzugangspunkt. Letzterer ist ein Knoten, welcher mit einer Kabelverbindung mit dem Server verbunden ist und mit dem sich die Kunden mittels ihrer Funktechnologie verbinden. Genauer hängt von der jeweiligen Funktechnologie ab. Beispielsweise handelt es sich im Fall von NB-IoT und LTE um eine Basisstation mit angeschlossenem Evolved Packet Core. Betrachten wir beispielhaft das Kommunikationsmuster Uplink (siehe Tab. 4), so versenden die Clients als Source zu zufälligen Zeitpunkten Daten an einen Server (Sink). Der Netzzugangspunkt empfängt die Datenpakete der Clients und leitet diese zum Server weiter. Der Inhalt der Daten spielt für die Simulation keine Rolle, lediglich ihr Umfang sowie die Sendezeitpunkte.

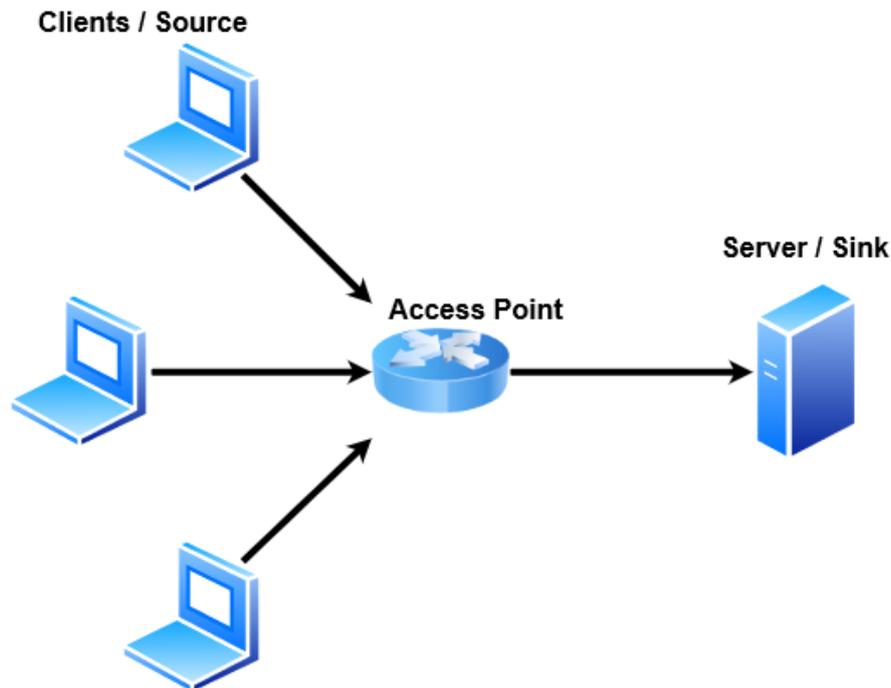


Abbildung 5: Netzwerkaufbau der simulierten allgemeinen Star Topology.

Auf diesem Netzwerk können eine Vielzahl von Verhaltensweisen der Funktechnologien simuliert werden. Sollte ein deutlich komplizierteres Netzwerk mit mehreren Netzzugangspunkten oder mehreren Servern simuliert werden müssen, muss dieses jedoch neu entwickelt werden. Solche Netzwerke testen dann aber weniger die Funktionsweise der Funktechnologie im allgemeinen, sondern viel mehr spezielle Anwendungsfälle.

6.2. Beschreibungssprache für Simulationen

Damit das in Abbildung 5 vorgestellte Netzwerk effektiv und für möglichst viele unterschiedliche Anwendungsfälle eingesetzt werden kann, wurde in TestKomm 4.0 eine Beschreibungssprache für Tests auf der Simulationsebene entwickelt. Diese ist zunächst möglichst einfach gehalten und kann in Zukunft weiter entwickelt werden, dann wird jedoch auch ihre Verwendung komplizierter. Der aktuelle Umfang genügt jedoch für alle in TestKomm 4.0 durchgeführten Tests.

Es genügt üblicherweise nicht, eine Funktechnologie mit einer einzelnen Parameterkombination zu testen. Vielmehr müssen Funktechnologien mit einer Vielzahl unterschiedlicher Parameterkombinationen getestet werden. Beispielsweise soll es möglich sein, dasselbe Netzwerk mit unterschiedlich großen Datenpaketen zu testen. Statt einer einzelnen Simulation muss dann eine ganze Reihe von Simulationen durchgeführt werden. Es ist jedoch nicht praktikabel, jede mögliche Parameterkombination auszuführen, denn deren Anzahl ist in der Regel zu groß. Somit ist es notwendig, Simulationen intelligent zu

beschreiben. Dafür wurde die in Abbildung 6 als Klassendiagramm dargestellte Beschreibungssprache für Simulationen geschaffen. Eine *Simulation* soll nur noch aus einer Liste aus unterschiedlichen Parametern vom Typ *Value* bestehen. Diese *Values* können unterschiedliche Typen *T* haben. Der einfachste Typ ist der *Single_Value*. Hierbei handelt es sich um einen einfachen Parameter, welcher einen Namen und einen einzelnen Wert hat. Ein solcher Wert hat keinen Einfluss auf die Anzahl der Simulationen, jede Simulation wird jedoch mit diesem gesetzten Wert ausgeführt. Der nächste Typ für *Values* ist das *Array*. Bei diesem werden mehrere Werte einem Parameter hinzugefügt. Es können also beispielsweise Simulationen mit unterschiedlich vielen Kunden beschrieben werden. Der letzte Typ ist die *Sequence*. Hier werden einem Parameter Werte vom Start- bis zum Stoppwert zugeteilt, wobei die *step_size* den Abstand zwischen aufeinanderfolgenden Werten angibt. Die *Sequence* könnte auch durch ein *Array* realisiert werden, jedoch können hiermit Parametern beliebig viele Werte zugewiesen werden. Bei einem *Array* müsste jeder Wert händisch getippt werden. Eine Simulation besteht zusätzlich zu den Parametern unter anderem aus der *repeatNumber*. Diese gibt an, wie oft jede Parameterkombination mit unterschiedlichen Zufallsergebnissen ausgeführt werden soll. Im *path* wird der Pfad zur verwendeten ns-3 Version gesichert. Dies ist relevant, da beispielsweise NB-IoT mit einer anderen ns-3 Version als Wifi getestet werden muss. Zum Schluss muss eine *simulation* gewählt werden. Hierbei handelt es sich dann um den Dateinamen der mit ns-3 programmierten Simulation.

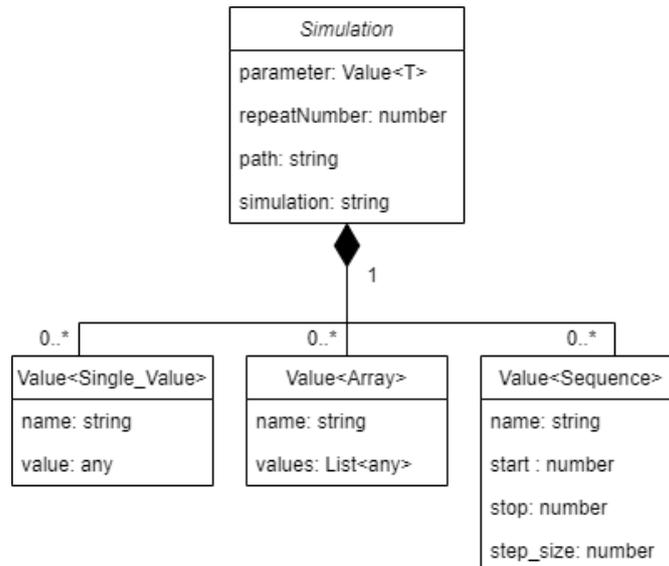


Abbildung 6: Klassendiagramm der Beschreibungssprache

Ein Beispiel, wie eine Instanz der Testbeschreibungssprache aussieht, zeigt Listing 1. Bei der Syntax handelt es sich um ein gewöhnliches JSON-Objekt. Für TestKomm 4.0 wurde ein Code-Generator entwickelt, welcher die Beschreibung liest und einen Test generiert. In diesem Beispiel testet der generierte Test Wifi mit 5, 10 und 15 Kunden

sowie Paketgrößen von 1, 10, 100, 200, 400, 600, 800 und 1000 Byte. Es werden in diesem Beispiel also $3 \cdot 8 \cdot 5 = 120$ einzelne Simulationen ausgeführt. Das Ergebnis wird in Abschnitt 8 präsentiert.

Listing 1: Beispiel einer Testbeschreibung als JSON

```
{
  "parameter": [
    {
      "name": "repeatNumber",
      "type": "string",
      "value": "5",
      "category": "single"
    },
    {
      "name": "simulationTime_s",
      "type": "string",
      "value": "20",
      "category": "single"
    },
    {
      "name": "nClients",
      "type": "string",
      "value": {
        "start": "5",
        "end": "15",
        "stepsize": "5"
      },
      "category": "sequence"
    },
    {
      "name": "maxPacketSize_byte",
      "type": "string",
      "value": ["1","10","100","200","400","600","800","1000"],
      "category": "array"
    },
    {
      "name": "interPacketInterval_ms",
      "type": "string",
      "value": "5",
      "category": "single"
    }
  ],
  "simulationName": "nb-iot",
  "basePath": "/home/testkomm/testkomm/ns3-developement-image/nbiot/"
}
```

6.3. TestKomm 4.0 Schnittstelle

Die in Listing 1 dargestellte Testbeschreibung muss nicht nur von einem Code-Generator generiert werden, es muss auch die Simulation entsprechend des generierten Tests ausgeführt und dann die Ergebnisse abgespeichert werden.

Die dafür notwendigen Akteure sind in Abbildung 7 dargestellt. Hierbei handelt es sich außerdem um das vollständige System für die Einbindung der ns-3-Simulationen in die Umgebung von TestKomm 4.0.

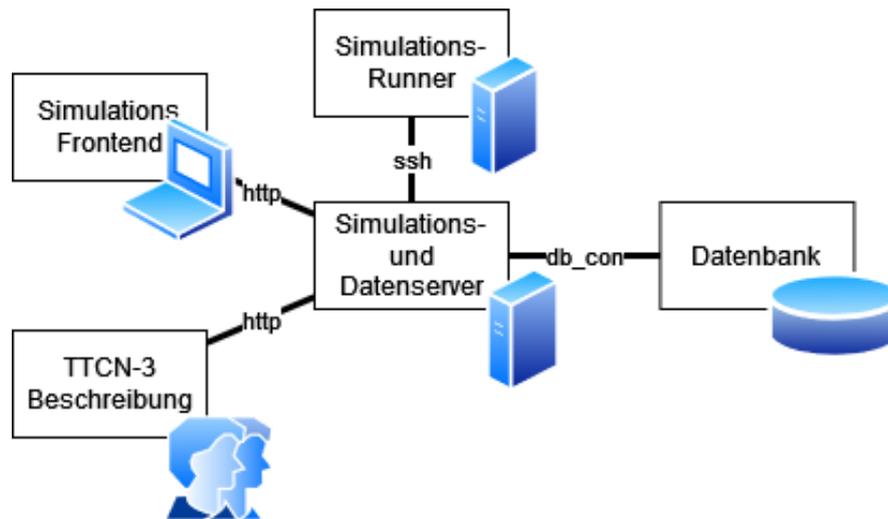


Abbildung 7: Systemansicht der Simulationsumgebung, einschließlich externer TTCN3-Schnittstelle

Die eigentliche ns-3-Simulation, welche mit den unterschiedlichen Parametern ausgeführt wird, befindet sich auf dem Server namens *Simulations-Runner*. Das Ausführen der Simulation auf dem *Simulations-Runner*, die Code-Generierung und das Abspeichern des Ergebnisses in einer Datenbank übernimmt der *Simulations- und Datenserver*. Hierbei handelt es sich um einen NodeJs⁵-Server, welcher mit Typescript⁶ entwickelt wurde. Dieses Backend wurde als gedächtnislose REST-API entwickelt. Eine REST-API stellt Regeln dafür auf, welche Nachrichten sie erwartet und welche Aktionen sie daraufhin ausführt. Der *Simulations- und Datenserver* erwartet eine Simulationsbeschreibung wie in Listing 1, generiert daraus einen Test und führt diesen mittels *Simulations-Runner* aus. Die Ergebnisse werden dann in der *Datenbank* (Database) für spätere Verarbeitungen gesichert. Wer dem *Simulations-Backend* eine solche Beschreibung sendet, spielt keine Rolle. Es ist lediglich wichtig, dass die Anfrage an die korrekte Adresse geschickt wird und die korrekte Form hat. Daher kann das *Simulations-Backend* problemlos durch eine mit TTCN-3 generierte Simulationsbeschreibung innerhalb der Umgebung für TestKomm 4.0 angesprochen werden.

⁵<https://nodejs.org/en/>

⁶<https://www.typescriptlang.org/>

Zusätzlich wurde ein *Simulations-Frontend* entwickelt, welches die Simulationsumgebung unabhängig von TestKomm 4.0 macht.

6.4. Frontend für die Ausführung von Simulationen

Das in Abbildung 7 vorgestellte Simulations-Frontend ermöglicht es, Simulationen mittels grafischer Oberfläche zu beschreiben und auszuführen. Abbildung 8 zeigt, wie dies als Website umgesetzt wurde. Ein genauer Einblick über die Datenhaltung des Projekts TestKomm 4.0 wird in Abschnitt 8 gegeben. Hier sollen lediglich die Eingabefelder gezeigt werden.

Simulation name
wifi

Add Parameter

Name	Variable type	Default	Description	Single value	Array	Sequence	Remove
repeatNumber	String	1	Current Repeat Number [1]	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remove
warmUpTime_s	String	0	warmUpTime [s] [0]	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remove
path	String	./result/	Path to save Results [./result/]	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remove
saveRawData	String	false	Will save EVERY Packet with Timestamp ar	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	Remove

(a) Definition der Parameter

simulationTime_s
20
Enter a single value
 Single value Array Sequence

nClients
5,15,5
Enter value as start,end,stepsize
 Single value Array Sequence

maxPacketSize_byte
1,10,100,200,400,600,800,1000
Enter comma separated values
 Single value Array Sequence

interPacketInterval_ms
5
Enter a single value
 Single value Array Sequence

Start simulation **Update subproject** **Delete subproject** **Results**

(b) Spezifikation der Werte für die Parameter

Abbildung 8: Beschreibung einer Simulation

Zunächst muss eine einzelne Simulation beschrieben werden – Siehe Abbildung 8a. Dafür werden alle von der ns-3-Simulation unterstützten Parameter in eine Liste einge-

tragen. Die einzelnen Parameter erhalten Default-Werte und eine Beschreibung. Das Ergebnis wird dann in einer Datenbank abgespeichert, um wie in Abbildung 8b verwendet zu werden. Hier werden gewünschte Parameterkombinationen in Eingabefelder eingegeben, um daraus eine Simulationsbeschreibung wie in Listing 1 zu erzeugen. Die Eingabefelder werden dafür zunächst mit den in Abbildung 8a eingegebenen Werte gefüllt.

Nachdem alle Werte wie gewünscht eingegeben wurden, kann die Simulationsbeschreibung abgespeichert und ausgeführt werden. Sollte es bereits Ergebnisse geben, können diese ebenfalls betrachtet werden – siehe Abschnitt 8.

7. Aufbau der Emulations- und Feldtestumgebungen

Dieses Kapitel deckt die Arbeitsschritte von AP 3 ab. Ziel dieses APs ist es, die Emulations- und Feldtestumgebung aufzubauen, um die Leistungsfähigkeit der Kommunikationslösungen zu messen.

Dieses Arbeitspaket wurde maßgeblich von ivESK bearbeitet, indem die physische Emulationsumgebung weiterentwickelt wurde, sowie das Gesamtttestsystm entwickelt wurde. Hahn-Schickard unterstützte durch die Entwicklung einer Weboberfläche zur grafischen Steuerung und Konfiguration der Emulationsumgebung.

Die Emulationsumgebung soll es ermöglichen, den HF-Kanal zwischen Knoten so einzustellen, dass realitätsnahe Umgebungen nachgestellt werden können. Dies beinhaltet neben der Kanaldämpfung auch Verzögerungen von Signalen und Signalausbreitungen über mehrere parallele Strecken. Diese Eigenschaften sollen zudem automatisiert und zeitgesteuert eingestellt werden können.

7.1. Zentrales Testsystem

Das Testsystem soll als zentrale Ausführungsplattform fungieren, um Tests systematisch und reproduzierbar in unterschiedlichen Umgebungen durchführen zu können. Neben der Emulationsumgebung (siehe Abschnitt 7.2) kann die Simulationsumgebung (siehe Abschnitt 6.1) benutzt werden sowie unterschiedliche Netzwerkgeräte unterschiedlicher Hersteller. Die Anforderungen an das Testsystem sind in Abschnitt 5.2.1 beschrieben. In diesem Abschnitt sollen die technischen Schritte zur Umsetzung erläutert werden.

7.1.1. Architektur

In Abb. 9 ist die Gesamtarchitektur des Testsystems dargestellt. Sie gibt einen Überblick zu den wichtigsten Komponenten innerhalb des Testsystems und deren Interaktion. In der Abbildung sind mehrere Komponenten dargestellt und deren Interaktion durch Pfeile dargestellt.

Der Kern des Testsystems ist die TTCN3-basierte Testumgebung. Hier werden Tests beschrieben und Daten gesammelt. Dieser Teil basiert auf dem Eclipse-Titan-Framework in Version 8.2.0 [48]. Diese oder eine neuere Titan-Version ist zwingend notwendig, da die Funktionen der objektorientierten Spracherweiterung [57] bei der Gestaltung der Architektur benutzt werden. Zur Interaktion mit den Testgeräten wurden zudem folgende Testports eingesetzt, die bereits in Titan definiert sind:

- *HTTPmsg* [58] zur Interaktion mit den REST-APIs
- *Serial* [59] zur Interaktion mit USB Geräten
- *TCPasp* [60] zur Interaktion mit Standard Commands for Programmable Instruments (SCPI) Geräten
- *UDPasp* [61] zum Aufsetzen eines UDP Servers und interne Kommunikationszwecke

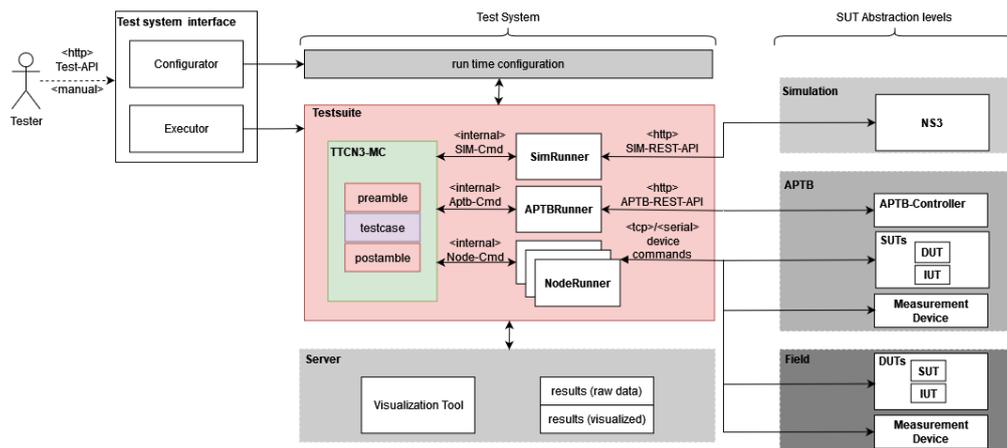


Abbildung 9: Gesamtarchitektur des Testsystems

Zur Interaktion mit externen Komponenten wie dem Simulationsserver, der Emulationsumgebung und Feldtestgeräten, sind die folgenden drei Komponenten definiert:

- *SimRunner*: Konfiguriert und führt Simulationen über eine REST-API aus
- *AptbRunner*: Konfiguriert und steuert die Emulationsumgebung über eine REST-API
- *NodeRunner*: Konfiguriert und steuert Netzwerkgeräte. Diese Komponente kann auf eine Technologie bzw. ein bestimmtes Gerät parametrisiert werden

Zur Umsetzung der Komponenten werden Parallel Test Components (PTCs) eingesetzt. Dies sind eigenständige Komponenten, die innerhalb von Testbeschreibungen erzeugt werden können und parallel eine Funktionalität erfüllen. Der TTCN3 Main Controller (TMC) ist für die Ausführung des Testfalls und die Koordinierung der anderen Testkomponenten zuständig. Über definierte Schnittstellen kann eine Kommunikation zwischen TMC und den PTCs hergestellt werden. Die Umsetzung dieser Kommunikation ist im TTCN3-Standard nicht definiert und wird durch das Titan Framework über eine interne Transmission Control Protocol (TCP)-Kommunikation mit benutzerdefinierten Nachrichtenformaten bewerkstelligt. Sowohl die Funktionalität der Komponenten als auch die Schnittstellen sind in TTCN3 definiert.

Die Beschreibung von Testcases wird auf der Ebene des TMCs vorgenommen. Hier werden alle notwendigen Komponenten, die für einen Testablauf benötigt werden, definiert, beispielsweise *NodeRunner* zur Kommunikation mit einem Endgerät, ein *UdpServer* zum Empfangen der Daten und ein *Writer* zum Erfassen und Abspeichern der generierten Events. Der Ablauf eines Testcases gliedert sich dabei in die drei Schritte: Präambel, Hauptteil, Postambel. Innerhalb des Testcases werden die notwendigen Komponenten erzeugt und konfiguriert. Schließlich wird den Komponenten die auszuführende Funktionalität übermittelt, welche dann eigenständig abgearbeitet wird. Die Komponenten

verfügen über einen Rückkanal, um den Ausführungsteil über den Status der Ausführung zu informieren. Auf diese Weise können auch mehrere Geräte quasi gleichzeitig angesteuert werden.

Um Testcases flexibel parametrieren zu können, wurden verschiedene Datenstrukturen definiert, welche über eine Laufzeit-Konfigurationsdatei einstellbar sind. Um die Einstellmöglichkeiten des Testsystems benutzerfreundlich zu gestalten, wurde zudem die Möglichkeit erstellt sogenannte Profile zu benutzen.

Da TTCN3 sowie das eingesetzte Framework Wissen erfordern, welches für einen Tester nicht zwingend notwendig ist, wurde eine REST-API definiert, die es ermöglicht Tests zu parametrieren und auszuführen, ohne mit dem Testsystem interagieren zu müssen. Die API ermöglicht ein vereinfachtes Nutzen des Testsystem, welches zudem über ein Frontend steuerbar ist.

Innerhalb der Testumgebung sind unterschiedliche Events definiert, die das strukturierte Speichern der Daten ermöglichen. Treten diese Events innerhalb des Testfalles auf, so werden die Daten strukturiert im Dateisystem des Testsystems gespeichert. Der Pfad ergibt sich nach folgendem Schema:

```
project/subproject/testcase/layer/technology/device/testname-timestamp
```

Um frühzeitig Rückmeldung zu Testergebnissen zu bekommen, sind Visualisierungsskripte innerhalb der Testumgebung installiert, die die vorhandenen Daten nach einem erfolgreichen Testdurchlauf untersuchen und plotten. Diese Grafiken werden zusammen mit den Daten abgelegt. Neben den Daten werden zudem die eingestellte Konfiguration sowie die vom Testsystem produzierte Log-Datei mit abgespeichert. Somit sind alle Testausführungen auch zu späteren Zeitpunkten noch nachvollziehbar.

Folgende Prinzipien wurden für die Implementierung der Testfälle berücksichtigt.

1. Testfall-Parameter werden über Laufzeitparameter verfügbar gemacht.
2. Testfälle werden auf der TMC-Komponente durchgeführt.
3. Testfälle, die zusammengehörig sind, werden in einem Modul zusammenfasst.
4. Testfälle werden auf einem hohen Abstraktionsniveau geschrieben, sodass Tests möglichst generisch bleiben.
5. Testfälle werden auf separaten (logischen) Komponenten (PTCs) ausgeführt.

7.1.2. Implementierung der Testfälle

Zur Beschreibung der Testfälle wird ein Beschreibungsschema verwendet, das schließlich im Testsystem und der Simulationsumgebung implementiert wird. Das Beschreibungsschema gliedert sich hierbei in die drei Phasen Präambel, Hauptteil und Postambel auf. Zusätzlich zum Beschreibungsschema wird ein Sequenzdiagramm entworfen, um den Testablauf zwischen den Komponenten zu verdeutlichen. In Abb. 10 ist exemplarisch ein Sequenzdiagramm für einen Uplink Testfall dargestellt.

Schließlich werden die Kommunikationsmuster in dem TTCN3-Framework sowie dem ns-3 Framework implementiert. In beiden Plattformen ist eine skriptbasierte Ausführung

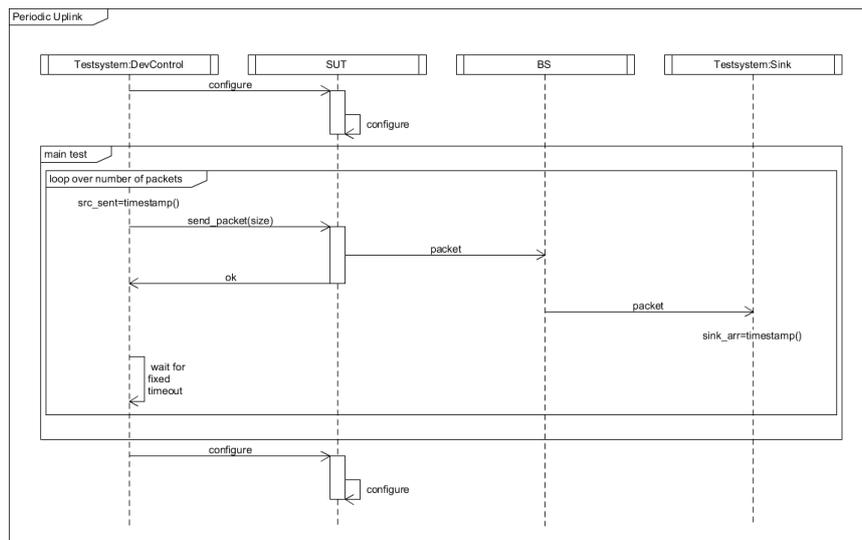


Abbildung 10: Sequenzdiagramm des Uplink Testfalles

möglich, wodurch sich einfach automatisierte Parametervariationen durchführen lassen können. Durch die Verwendung einfacher Kommunikationsmuster und einheitlicher Konfigurationsparameter kann die Beschreibung der Testfälle in beiden Plattformen als einheitlich angesehen werden.

Die Ausführungsschritte zur Implementierung eines Testfalles innerhalb des Testsystems ist in Abb. 11 dargestellt.

7.1.3. Konfigurationsparameter

Die Konfigurierbarkeit von Testfällen stellt ein wichtiges Merkmal des Testsystems dar. In Abb. 12 sind die unterschiedlichen Ebenen von Konfigurationsparametern dargestellt. Jede der dargestellten Ebenen beinhaltet relevante Parameter, die für jeden Testfall klar definiert sein müssen. Die Kombination dieser Parameter ergibt schließlich eine vollständige Testparametrierung.

Um diese Konfigurationsoptionen umzusetzen wurden zwei Dinge definiert:

1. Eine Datenstruktur zur Beschreibung der Konfigurationsoptionen
2. Vordefinierte JSON-Profile je Konfigurationsebene

In Listing 2 ist die höchste Ebene der Konfigurationsdatenstruktur dargestellt. Diese ist generisch gehalten, da damit alle möglichen Fälle abgedeckt werden müssen.

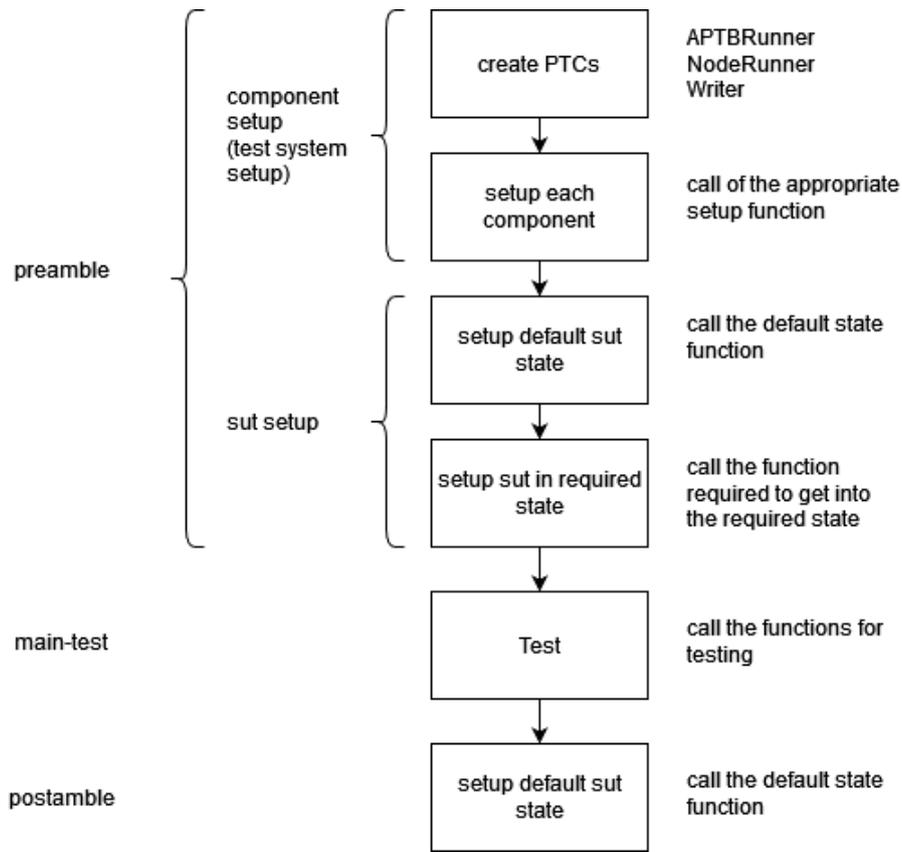


Abbildung 11: Typischer Ablauf eines Testfalles innerhalb des Testsystems

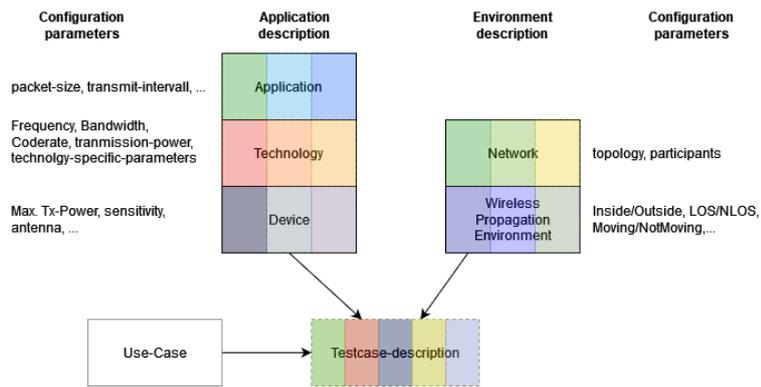


Abbildung 12: Konfigurationsebenen

Listing 2: Datenstruktur zur Beschreibung von Konfigurationsparametern

```

type record TestrunConfig {
  TestDescr descr,
  PacketParameters pck_param,
  DeviceParameter end_device_param,
  DeviceParameter bs_device_param optional,
  NetParameter net_param,
  EdTechParameter ed_tech_param,
  BsTechParameter bs_tech_param optional,
  WirelessParameter env_param
}

```

Diese Konfigurationsparameter sind als separates Modul (*ConfigurationTypes*) definiert, das in den Testfällen eingebunden wird. Jeder Testfall wird dabei mit einer einheitlichen Signatur definiert und nimmt genau einen Parameter vom Typ *TestrunConfig* entgegen. Diese Datenstruktur parametrisiert damit den gesamten Testcase. Aus Sicht eines Nutzers sollen auch mehrere Konfigurationsoptionen hintereinander ausführbar sein. Daher werden auf der Ebene der Konfigurationsdatei die Applikationsparameter (Sendintervall, Paketgröße, usw.) und die Umgebungsparameter als Array angeboten. Der Ausführungsteil (*control*) innerhalb eines Testmoduls sorgt dafür, dass die Testfälle mit den gewünschten Parameterkombinationen aufgerufen werden.

Neben der Parametrierung von Testfällen sind auch einzelne Testfälle auswählbar. Hierfür werden sogenannte *moduleparameter* für jeden Testfall definiert. Nur wenn ein Nutzer explizit einen oder mehrere Testfälle aktiviert, werden diese auch ausgeführt.

7.1.4. Interaktion mit Testgeräten

Testfälle sollen möglichst generisch für unterschiedliche Technologien beschreibbar sein, damit diese unabhängig von der Technologie ausgeführt werden können. Gleichzeitig gibt es je Technologie spezifische Details, die nur für diese Technologie relevant sind. Aus diesem Grund wurden mehrere Interfaces für unterschiedliche Detailgrade verwendet. Für die verwendeten End Devices (EDs) wird die in Abbildung 13 dargestellte Klassenstruktur verwendet. Die dargestellten Funktionen stellen hierbei nur einen Ausschnitt einiger relevanter Funktionalitäten dar.

Das *CommonInterface* erlaubt es, einfache Testfälle, wie das Senden von Bytes, auszuführen. Die anderen Klassen müssen diese Funktionalität implementieren. Will man aber einen speziellen NB-IoT spezifischen Test durchführen, muss man das entsprechende Interface nutzen.

Als Hardwareschnittstelle gibt es die Möglichkeiten, eine serielle Schnittstelle oder eine TCP basierte Schnittstelle zu verwenden. Die Auswahl kann hierbei über die Konfigurationsdatei vorgenommen werden. Die Implementierungen einzelner Geräte greifen auf Methoden zum Senden und Empfangen über diese Schnittstellen zurück. Neben Sende- und Empfangsnachrichten werden im Testport auch die Zeitstempel der Kommunikation erfasst, direkt bevor eine Nachricht über den Port gesendet wird und

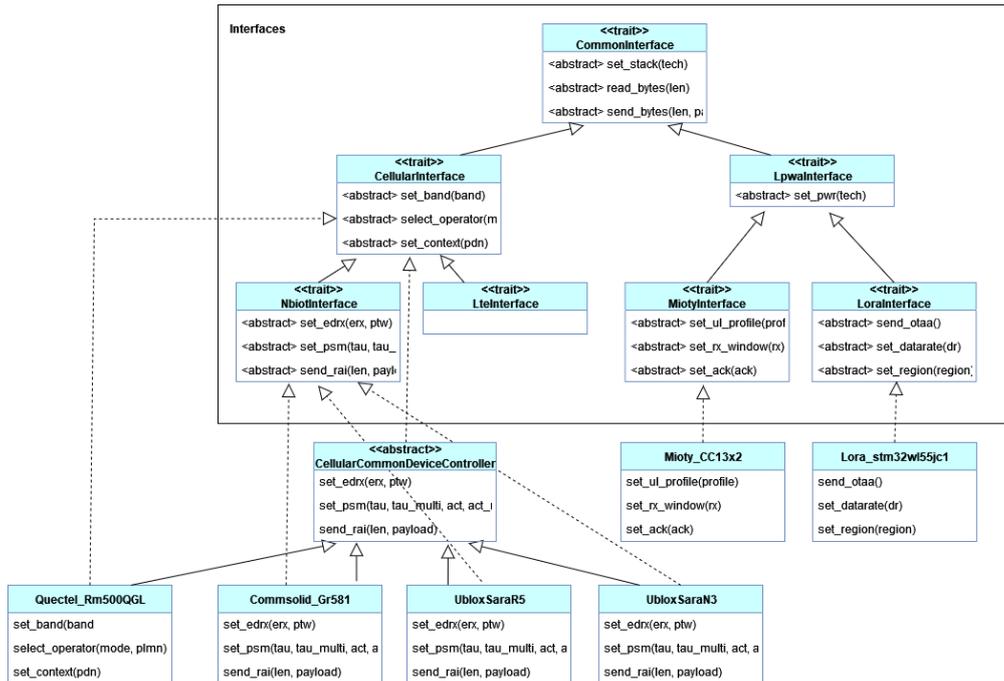


Abbildung 13: Interfaces und Klassen zur Implementierung der DUTs

direkt nachdem eine Nachricht ankommt. Auf Applikationsebene wird schließlich geprüft ob die Nachricht erfolgreich gesendet wurde bzw. welcher Nachrichtentyp empfangen wurde, um ein entsprechendes Event an die *Writer* Komponente zu senden.

7.1.5. Anbindung der Simulationsumgebung

Zur Interaktion mit der Simulationsumgebung wurde die definierte REST-API über TTCN3 eingebunden. Hierfür wurden zunächst die simulationsspezifischen Parameter als TTCN3-Typ eingeführt sowie entsprechende JSON-Dekoder für die Nachrichten definiert (Siehe Beispiel einer Testbeschreibung in Abschnitt 6.1), die zwischen dem Testsystem und dem Simulationsserver ausgetauscht werden.

Für die Interaktion mit dem Simulationsbackend wird der Ablauf wie in Abb. 14 dargestellt in der *SimRunner*-Komponente implementiert.

Für das Senden der Hypertext Transfer Protocol (HTTPS) Nachrichten wurde auf den bereits vorhandenen HTTP-Port [58] zurückgegriffen. Da die Interaktion mit REST-Schnittstellen darüber hinaus einen hohen Stellenwert hat, wurde ein zusätzliches Modul für diese Funktionalität definiert. Dieses wird ebenfalls für die Anbindung der Emulationsumgebung verwendet.

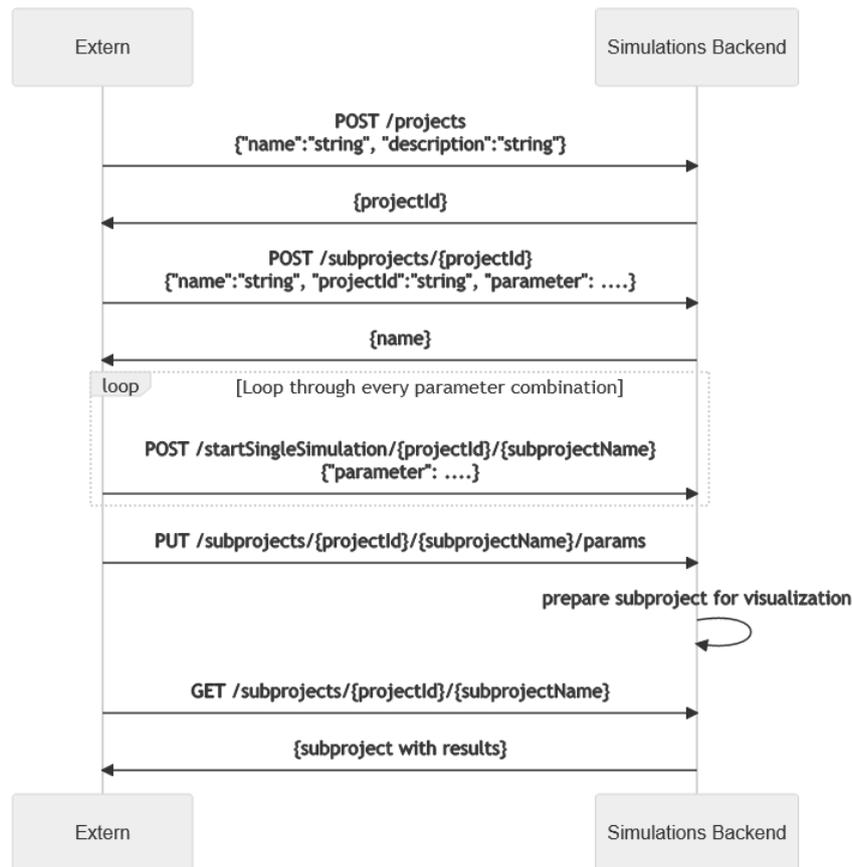


Abbildung 14: Interaktion mit dem Simulationsbackend über die REST-API

7.1.6. Anbindung der Emulationsumgebung

Zur Interaktion mit der Emulationsumgebung bzw. dem Automated Physical Testbed (APT_B) wurde eine definierte REST-API (vgl. Abschnitt 7.2.4) in TTCN3 umgesetzt. Weiterhin wurde eine weitere TTCN3-Komponente als PTC definiert, die für die Steuerung des APT_B zuständig ist. Diese Komponente wird, wie auch die anderen PTCs, über eine definierte Schnittstelle angesprochen. Diese interne Schnittstelle wird innerhalb von Testbeschreibungen benutzt, um das APT_B zu steuern. Folgende Funktionalität wird damit abgedeckt.

- Aktivierung/Deaktivierung von Kanälen
- Konfiguration mehrerer Kanäle
- Konfiguration eines einzelnen Kanals
- Zeitlich gesteuerte unabhängige Änderung der Dämpfung

Die Datenstruktur zur Konfiguration eines APTB-Kanals ist in Listing 3 dargestellt. Ein Kanal entspricht hierbei der Verbindung zwischen zwei Knoten, was intern über eine *channel_id* beschrieben wird.

Listing 3: Beispiel einer Aptb-Kanalparametrierung

```
type record AptbChannel {
  charstring channel_id,
  boolean switch_state optional,
  integer attenuation_db optional,
  integer delay optional,
  IntArr multipath optional
}
```

7.1.7. Interaktion mit Messgeräten

Neben der Automatisierung der DUTs sind auch Messgeräte von hoher Wichtigkeit. Für die Ansteuerung von Messgeräten hat sich SCPI als digitale Schnittstelle durchgesetzt. Zur Erhöhung der Testautomatisierung wurde auf Basis des SCPI ein Interface zur Bedienung des Kommunikationstesters CMW500 in TTCN3 implementiert. Hierfür wurde ein SCPI Controller entwickelt, welcher das SCPI Protokoll beherrscht und eine Klasse für den CMW500 definiert, welche die Funktionalität des CMW500 implementiert. Aufgrund der Fülle des Funktionsumfangs wurde allerdings nicht das volle Funktionsspektrum implementiert, sondern nur die für das Projekt relevanten Funktionen berücksichtigt. Es ist möglich die wichtigsten Netzwerkparameter einzustellen, die Zelle ein und auszuschalten sowie Rauschen auf dem Downlink-Kanal hinzuzufügen. Die separate CMW500-Komponente kann über ein definiertes Port-Interface vom Testfall aus gesteuert werden. Dieses erlaubt das Einstellen von Technologiekonfigurationen und Rauschkonfigurationen über unterschiedliche Nachrichtentypen. In Listing 4 ist ein Ausschnitt aus dem Nachrichtenformat des CMW500-Testports dargestellt.

Listing 4: Nachrichtenformat des CMW Konfigurationsports

```

type port BsCfg_PT message
{
  inout e_BsConfigStatePreset;
  inout e_BsConfigAction;
  inout CellActivate; // activate a cell
  inout CellRoute; // cell connectors + signalling unit
  ...
  inout Nprach; // configure PRACH parameter
  inout CoverageLevels; // coverage extension levels
  inout DIPower; // Downlink power settings
  inout DynamicUIPowerControl; // uplink power control param
  inout UINbiotSchedule; // scheduler parameters uplink
  inout DINbiotSchedule; // scheduler parameters downlink
  inout Psm; // psm parameters
  inout Edrx; // edrx parameters
  inout Npdccch; // downlink control channel
  inout NbiotFeatures; // int.rand+ release assistance
  inout BsStatus; // communicate status to caller
  inout BsState; // communicate basic cell state
} with { extension "internal" }

```

7.1.8. Visualisierung und Zusatzskripte

Bei der Durchführung von Tests ist es wichtig, frühzeitig Rückmeldung zu gemessenen Werten zu bekommen. Folgende Dinge sind für einen Tester entscheidend:

- Testverdict: Hat der Testablauf den Erwartungen entsprochen (Funktionalität)
- Log-Datei: Um im Falle von Ausführungsproblemen die entscheidende Stelle auszumachen (Debugging)
- Plots: Um Rückmeldung zu den untersuchten KPIs zu bekommen (Analyse)

Das Konzept des Testverdicts ist ein zentrales Konzept in TTCN und wird verwendet um die Funktionalität des Tests zu überprüfen. Dabei sind folgende Definitionen für einen Verdict innerhalb dieses Testsystems wichtig:

- *pass*: Der Test ist bestanden. Alles hat wie erwartet funktioniert.
- *inconc*: Ein Timeout ist an einer Stelle innerhalb des Tests aufgetreten, aber die grundsätzliche Funktionalität ist erfüllt.
- *fail*: Es sind fehlerhafte Rückgaben innerhalb des Tests aufgetreten, sodass der Test als nicht bestanden gilt.
- *error*: Fundamentale Vorbedingungen für einen Test sind nicht erfüllt, z.B. ist eine Komponente oder eine IP-Adresse nicht verfügbar.

Im Anschluss an die Ausführung eines Testcases, wird ein Skript aufgerufen, welches den Testverdict an der richtigen Stelle speichert, die Logdateien aller beteiligten Komponenten in eine Datei zusammenfügt und an eine geeignete Stelle abspeichert. Zudem wird ein Visualisierungsskript gestartet, das die Daten an dem definierten Speicherort gemäß den key performance indicators (KPIs) visualisiert und die wichtigsten Statistiken berechnet. Als Beispiel dieser Visualisierung ist in Abb. 15 die Latenz dargestellt.

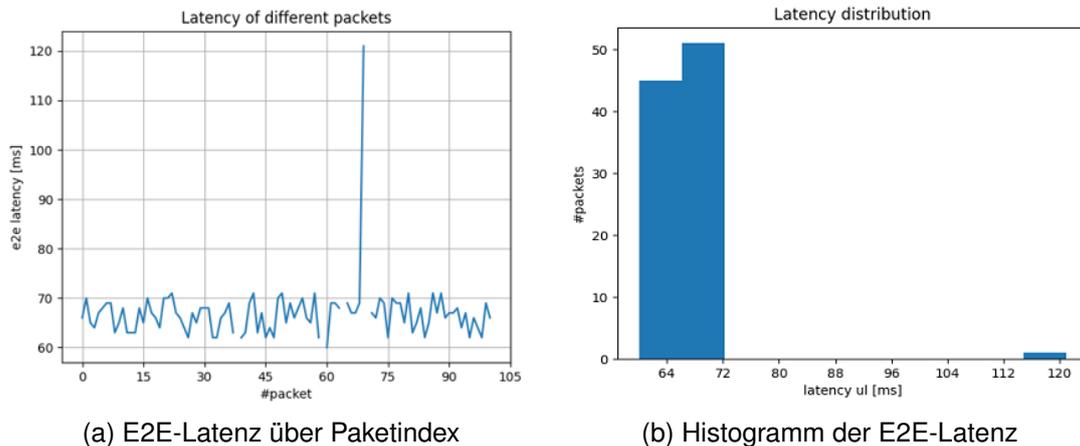


Abbildung 15: Visualisierung der E2E-Latenz eines Testfalls

7.1.9. Testsystem-Interface

Die Nutzung des zentralen Testsystems erfordert Wissen zu deren Benutzung, beispielsweise wie das Format der Konfigurationsdatei aufgebaut ist, wie Testfälle aktiviert werden und wie das Programm über die Kommandozeile gestartet werden kann. Um das System benutzerfreundlicher zu gestalten wurde eine REST-API entwickelt, die die wichtigsten Funktionalitäten des Testsystems abbildet. Dazu zählen:

- Auswahl der Abstraktionsebene (Feld, Emulation, Simulation)
- Auswahl einer Technologie und eines Gerätes
- Auswahl eines oder mehrerer Testfälle
- Auswahl vordefinierter Profile
- Konfiguration des Datenverkehrs und der Umgebungsbedingungen
- Automatisiertes Durchführen von Tests
- Rückmeldung über des Status der durchgeführten Tests

Das entwickelte Frontend ist in Abb. 16 dargestellt. Im Vergleich zum manuellen Bearbeiten der Konfiguration werden im Frontend alle Möglichkeiten angezeigt, sodass

beispielsweise eine fehlerhafte Kombination von Technologie und Testmodul nicht möglich ist.

The screenshot displays the TestSystem Configurator and Execution interface. The browser address bar shows the URL 192.168.1.31:5000. The interface is divided into several sections:

- Abstraction layer:** A dropdown menu is set to "simulation".
- Abstraction layer setting:** Includes a "Configure layer" section with input fields for "distance" (50) and "propagation_loss_model" (COST231), and a "Configure" button. "Run Single" and "Run Batch" buttons are also present.
- Testmodules:** A dropdown menu is set to "TC_protocol_states".
- Testcase:** A dropdown menu is set to "TC_PU_STATIC_N_SUT", with "Add" and "Remove" buttons below it.
- Selected Testcases:** A section for managing selected testcases.
- Technology:** A dropdown menu is set to "lte", with a "Configure" button below it.
- Device:** A dropdown menu is set to "quectel_rm500gl", with a "Configure" button below it.
- Profile:** A dropdown menu is set to "pa3", with a "Configure" button below it.
- Traffic:** Includes input fields for "packet_size", "num_packets", "response_packet_size", and "transmit_int", with a "Configure" button below them.
- Configure all:** A button located to the right of the Traffic section.

Abbildung 16: Frontend zur Konfiguration und Ausführung des Testsystems

7.2. Automated Physical Testbed

Am ivESK wird bereits ein Emulator für Kanaleigenschaften eingesetzt. Er dient zum statischen Aufbau unterschiedlicher Topologien mit einstellbaren Dämpfungen zwischen Knoten. Die Implementierung orientiert sich an den Anforderungen, wie in Abschnitt 5.2.3 beschrieben.

Das APTB wurde hierfür komplett überarbeitet, um neben flexibleren Netzwerkaufbauten auch zusätzliche Elemente wie Mehrwegeausbreitung zu unterstützen.

7.2.1. Hardware Architektur

Die neue APTB-Architektur, die in Abb. 17 dargestellt ist, wurde so konzipiert, dass sie modular und erweiterbar ist und gleichzeitig den Verkabelungsaufwand reduziert und die Steuerung möglichst vieler HF-Elemente ermöglicht. Bei dieser Architektur werden die HF-Elemente in Gruppen unterteilt. Jede Gruppe wird von einem Element Controller (EC) gesteuert, der direkte General Purpose Input Output (GPIO) zu den HF-Elementen hat. Die ECs sind an ein Ethernet-Netzwerk angeschlossen, um mit einem Main Controller zu kommunizieren. Um den Verkabelungsaufwand zu verringern, sind die ECs mit einem Power-over-Ethernet (PoE) Chip ausgestattet, der den Mikrocontroller mit Strom versorgt. Bei dieser Architektur können mehrere PoE-Switches miteinander verbunden werden, um die Anzahl der kontrollierten Geräte zu erhöhen. Als Main Controller (MC) wurde ein Raspberry Pi 3 B+ und als EC ein ESP32-Mikrocontroller mit einer eingebauten PoE-Schnittstelle gewählt.

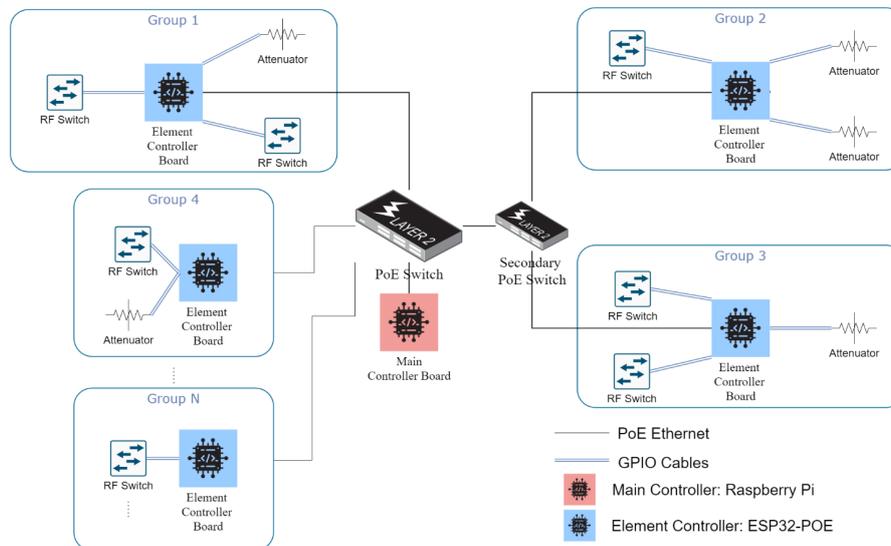


Abbildung 17: Hardwarearchitektur des APTBs

7.2.2. Hardware Aufbau

Alle alten Master- und Slave-Steuerplatinen, Stromversorgungen, invertierende Stromversorgungen, Multiplexer und andere steuerungsbezogene Schaltungen und Komponenten wurden entfernt. Vom alten Aufbau sind nur noch die HF-Elemente und Abschirmboxen übrig.

Das ursprüngliche APTB-System wies eine Reihe von Einschränkungen auf, die den gesamten Schaltkreis kompliziert und schwer zu handhaben machten. Um die Schaltung zu vereinfachen, wurde eine Platine mit den folgenden Funktionen entworfen:

- Breakout-Platine für den Olimex ESP32-P0E: Alle GPIOs und GPIOs, der ESP_EN-Pin, die +5V- und +3,3V-Stromversorgungspins und der Masse-Pin sind über Klemmen leicht zugänglich.
- Ein integriertes Dual-Rail-DCDC-Wandlermodul liefert sowohl -5 VDC als auch 5 VDC mit einer gemeinsamen Masse für das gesamte Board. Damit können sowohl die Schalter als auch das Olimex ESP32-Board über den 5 V-Pin mit Strom versorgt werden.
- Die Steuerung der HF-Dämpfungsglieder erfolgt über die GPIO-Pins als CS/LE-Pins. Alle Datenleitungen und CLK-Leitungen der Dämpfungsglieder verwenden die gleichen GPIOs. Diese GPIOs können über zwei Reihen von Klemmen zugänglich gemacht werden. Dies vereinfacht die Schaltung, da die Anzahl der Komponenten und Drähte reduziert wird.

Für die entworfenen Schaltpläne wurden Platinen entworfen, bestückt und mit den bereits vorhandenen HF-Komponenten verkabelt.

In Abb. 18 ist der Aufbau der APTB Implementierung dargestellt.

7.2.3. Software-Architektur

Die Software-Architektur des APTB wurde so konzipiert, dass sie ein einfaches und intuitives Steuerungssystem für die Element-Controller und RF-Elemente bietet und gleichzeitig dank ihrer modularen Konfiguration die Verwendung und Implementierung verschiedener Protokolle ermöglicht. Abb. 19 zeigt die allgemeine Software-Architektur des APTB.

Die Element-Controller definieren die Konfiguration der von ihnen gesteuerten HF-Elemente und übermitteln diese Informationen während einer Registrierungsphase an den MC. Der MC ist ein Raspberry Pi, auf dem mehrere Docker-Container installiert sind. Der wichtigste ist Node-RED, der die gesamte Kommunikation zwischen MC, ECs und externen Steuerskripten wie TTCN3 abwickelt. Es definiert die gesamte Logik für die Registrierung/Deregistrierung von ECs, die REST-API für die Interaktion mit dem APTB und Datenoperationen auf der MongoDB-Datenbank. Diese Datenbank enthält alle Daten im Zusammenhang mit dem APTB.

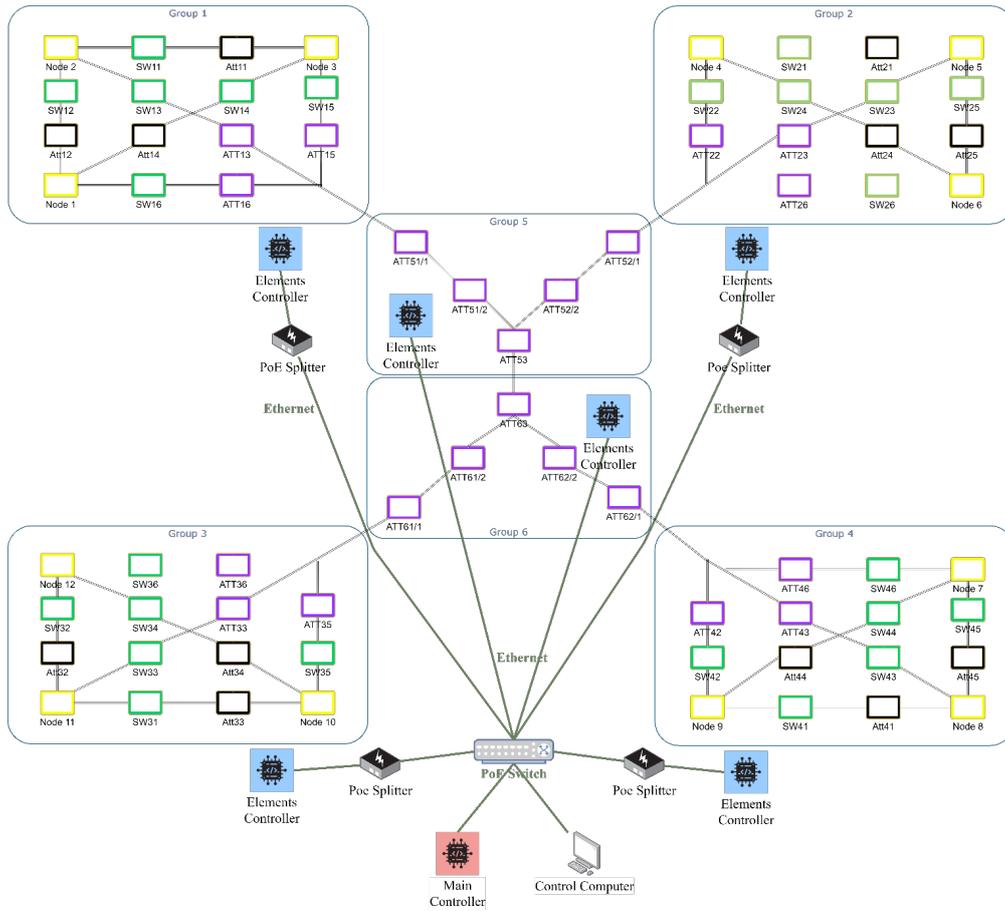


Abbildung 18: Aktueller APTB Aufbau in ivESK

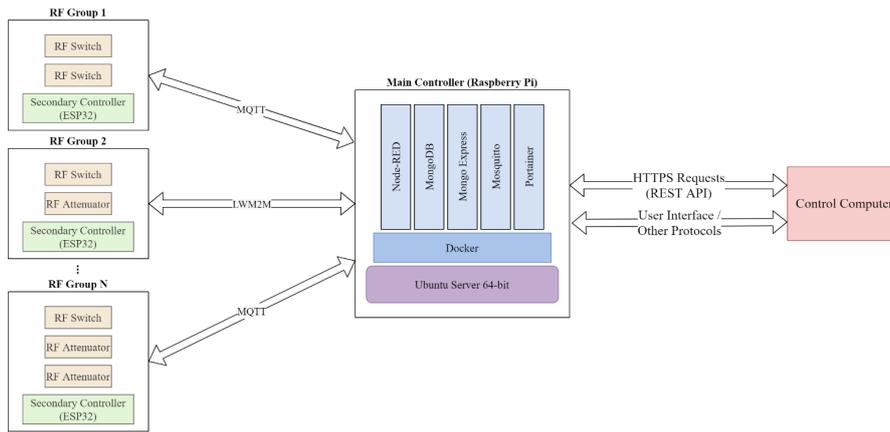


Abbildung 19: Softwarearchitektur des APTBs

Abb. 20 zeigt die verschiedenen Container des MCs im Detail. Ein Steuerrechner, auf dem TTCN3 oder eine Benutzerschnittstelle läuft, interagiert mit dem APTB über Node-RED unter Verwendung der REST-API. Je nach API-Anfrage sendet Node-RED Steuerbefehle an die EC und führt Operationen und Abfragen in der MongoDB-Datenbank aus. Mongo Express wird zur Verwaltung der MongoDB-Datenbank verwendet. Alle Docker-Container des Main Controllers können über Portainer verwaltet werden, ein Container-Management-System, das die Visualisierung der Container, Images, Volumes und Netzwerke ermöglicht.

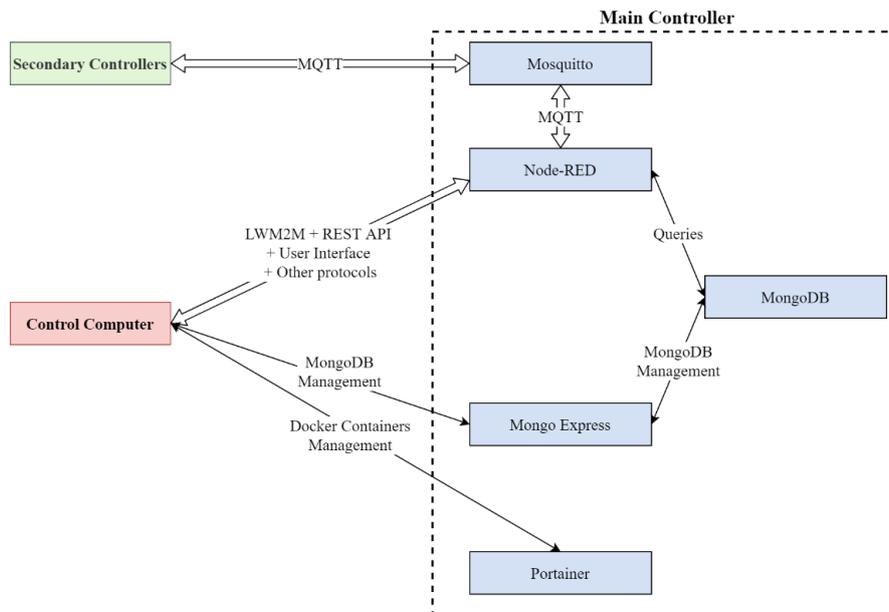


Abbildung 20: Softwarearchitektur des Main Controllers innerhalb des APTBs

Der EC implementiert die Klassen, die zur Steuerung der HF-Elemente erforderlich sind, nämlich die variablen Dämpfungsglieder, Schalter, Mehrwegeelemente und Verzögerungsleitungen. Das Diagramm in Abb. 21 zeigt die Klassen des EC sowie ihre Attribute und Operationen.

7.2.4. Steuerung des APTBs

Der MC kann die angeschlossenen Geräte über unterschiedliche Protokolle verwalten und ansteuern. Dies ermöglicht es, dass auch in Zukunft unterschiedliche Arten von ECs, unter Verwendung unterschiedlicher Protokolle, verwendet werden können. Die beiden Hauptprotokolle, die üblicherweise für die Kommunikation und das Gerätemanagement von eingebetteten IoT-Geräten verwendet werden, sind Message Queuing Telemetry Transport (MQTT) [62] und Lightweight M2M (LwM2M) [63]. Diese beiden Protokolle werden vom MC unterstützt. Um dies zu erreichen, wurde Node-RED, ein flussbasiertes Programmierwerkzeug, das Hardwaregeräte mit Online-Diensten und APIs

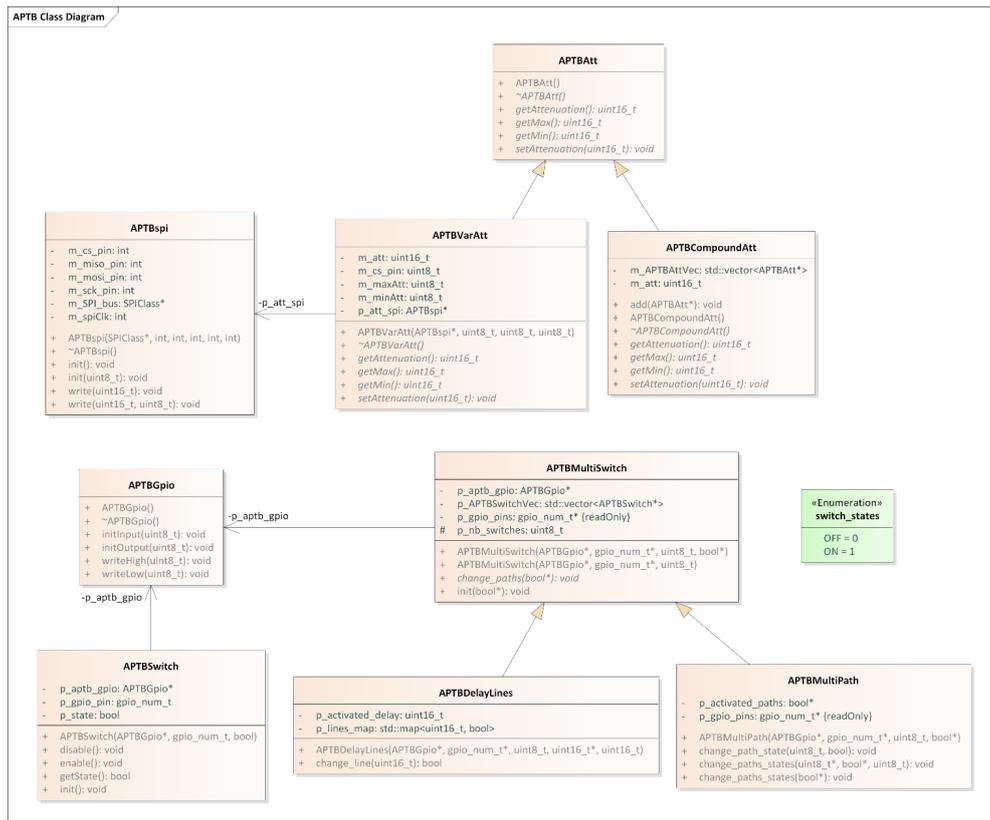


Abbildung 21: Klassendiagramm der Element Controller

verbinden kann, gewählt. Für den MQTT-Broker wurde für diese Arbeit Mosquitto als geeignete Lösung ausgewählt.

Für das Gerätemangement für die ECs und die damit verbundenen HF-Elemente wird MQTT verwendet. Die HF-Elemente sind hierbei jeweils einem bestimmte EC zugeordnet.

folgt einer festgelegten Topic-

Das Management der Geräte erfolgt über MQTT. Hierzu wurde eine entsprechende Topic-Architektur definiert, die die Identifizierung entsprechender ECs und HF-Elemente zueinander ermöglicht. Abb. 22 zeigt ein Beispiel für ein Topic, das verwendet wird, um den Zustand eines HF-Schalters zu ändern. Die HF-Komponenten werden anhand der HF-ID und der EC-ID klar identifiziert. Der MC kann die unterschiedlichen Topics verschiedener ECs auch über eine Wildcard einfach abonnieren, wie in Abb. 23 dargestellt.

APTB / C4:DD:57:AF:AD:4F / updateSwitch / SW1 / request



Elements Controller MAC Address
(Elements Group ID)



Switch ID
in the Elements Group

Abbildung 22: Beispiel für ein MQTT-Topic, das zur Aktualisierung des Zustands eines HF-Schalters verwendet wird

APTB / + / updateSwitch / + / response



The Wildcard "+" allows the MC to
subscribe to the response topics of
all of the Elements Controllers and
for all their RF Switches

Abbildung 23: Beispiel für die Verwendung von MQTT-Wildcards zum gleichzeitigen Abonnieren mehrerer Themen

Die ECs verfügen über eine gespeicherte Konfiguration, die sie selbst sowie die von ihnen gesteuerten RF-Elemente beschreibt. Diese Konfiguration ist in JSON geschrieben, wodurch sie eine gute Strukturierung bekommt und einfach zu parsen ist.

Um den Zustand der APTB-Komponenten zu jedem Zeitpunkt zu speichern, musste eine Datenbank implementiert werden. Aufgrund der Beschaffenheit der Daten und aus Gründen der Funktionalität die dokumentenorientierte Not Only SQL (NoSQL) Datenbank [64] ausgewählt, die JSON-ähnliche Dokumente mit optionalen Schemata verwendet.

Um mit dem APTB zu interagieren und Steuerbefehle zu senden, die von einer externen

Maschine (z.B. einem Computer) kommen, wurde eine REST-API in Node-RED implementiert, die das HTTPS Protokoll verwendet. Bei der Entwicklung der API wurden die Grundprinzipien von REST-APIs eingehalten, nämlich

- Beibehaltung einer Client-Server-Architektur
- Zustandslosigkeit
- Cache-Fähigkeit
- Einheitliche Schnittstelle
- Schichtensystem.

7.2.5. Mehrwegeausbreitung und Verzögerungselemente

Bei Mobilfunk- und Funkanwendungen, bei denen sich die Position der Antenne eines Endgerätes ständig ändert, erfolgt die Ausbreitung der Signale zwischen ihnen durch Streuung, entweder durch Beugung oder Reflexion an Objekten wie Gelände und Gebäuden. Dies wird durch die Behinderung einer direkten Sichtverbindung (LOS) verursacht und führt dazu, dass die Funkwellen aus verschiedenen Richtungen und mit unterschiedlichen Amplituden, Zeitverzögerungen und Phasen beim Empfänger ankommen und miteinander interferieren. Das daraus resultierende Phänomen wird als Mehrwegeausbreitung bezeichnet [36].

Mehrwegeinterferenzen verschlechtern die Leistung drahtloser Kommunikationssysteme erheblich, und es kann nicht viel getan werden, um sie zu beseitigen. Wenn das Mehrwegmedium jedoch gut untersucht wird und man die Ausbreitungsmechanismen und ihre Auswirkungen auf ein System gut kennt, kann man das beste Design für ein System auswählen, um eine gute Ausbreitungsleistung und damit eine bessere Dienstqualität zu erreichen.

Für die Erweiterung des APTB-Aufbaus sollen auch Verzögerungs- und Mehrwegeeffekte betrachtet werden, um Forschern die Möglichkeit zu geben mit der Mehrwegeausbreitung zu experimentieren. Mit Hilfe von Kombinationen aus HF-Splitttern, HF-Schaltern und HF-Wellenleiterkabeln wurde eine neue HF-Komponente entwickelt, über die je nach den gewünschten Ausbreitungsergebnissen Mehrwegekomponenten in verschiedenen Architekturen erstellt werden können.

Die Benutzer können Pfade aktivieren und deaktivieren, um mit verschiedenen Pfadkombinationen zu experimentieren. Sie können auch die Anzahl und die physischen Längen der Pfade ändern, indem sie Kabel mit unterschiedlichen Längen hinzufügen und austauschen. Die Eigenschaften der Ausbreitungsgeschwindigkeit der Kabel sollten berücksichtigt werden, um die gewünschten emulierten Pfadlängen zu berechnen und zu erhalten. Abb. 24 zeigt ein Beispiel für die Architektur eines Mehrwegeelements mit drei möglichen Pfaden.

Zusätzlich zu den Mehrwegeeffekten wurde auch eine Emulation für die verzögerte Ausbreitung von Funksignalen entwickelt. Mit dieser Funktion können die Auswirkungen von

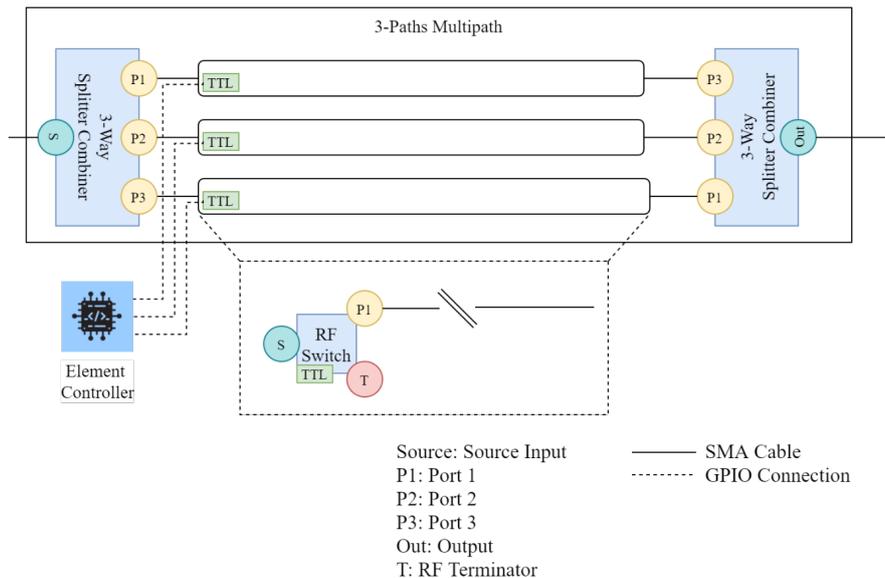


Abbildung 24: Aufbau der Mehrwegeelemente

Verzögerungen nicht nur auf das Signal selbst getestet werden, sondern auch darauf, wie das zu testende Protokoll funktioniert und auf Verzögerungen in der Kommunikation reagiert. Diese Komponente kann mit der gleichen Hardware-Architektur wie der Mehrwege-Emulator aufgebaut werden. Der einzige Unterschied besteht darin, dass jeweils nur ein Pfad aktiviert werden kann.

In diesem Abschnitt sind einzelne Tests dargestellt, die diesen Aufbau verifizieren. Ziel ist es

- Eine Vermessung der beteiligten HF-Komponenten durchzuführen.
- Den Einfluss unterschiedlicher Konfigurationen auf Sinussignale sowie drahtlose Protokolle zu erfassen.

Mit diesem Aufbau können prinzipiell beliebige Verzögerungen auf unterschiedlichen Pfaden eingestellt werden. Entsprechend wurden die Koaxialkabel, HF-Splitter und Schalter auf unterschiedlichen Frequenzen ausgemessen bezüglich der Verzögerung, der Dämpfung und der erzeugten Phasenverschiebung.

Um bestimmte Phasenverschiebungen zu erzeugen, wurden die Koaxialkabel für die relevanten Downlink-Frequenzen der Technologien ausgemessen. Dadurch konnten Extremfälle getestet werden, wie beispielsweise eine Phasenverschiebung von 180° auf einem Mehrwegepfad.

In Abb. 25 ist der SNR für unterschiedliche Kombinationen von Phasenverschiebungen und Technologien dargestellt.

Je Konfiguration wird ein Einfluss auf den SNR deutlich, der zudem deutlich zwischen den Technologien schwankt. Insbesondere bei MIOTY kann für eine Phasenverschiebung

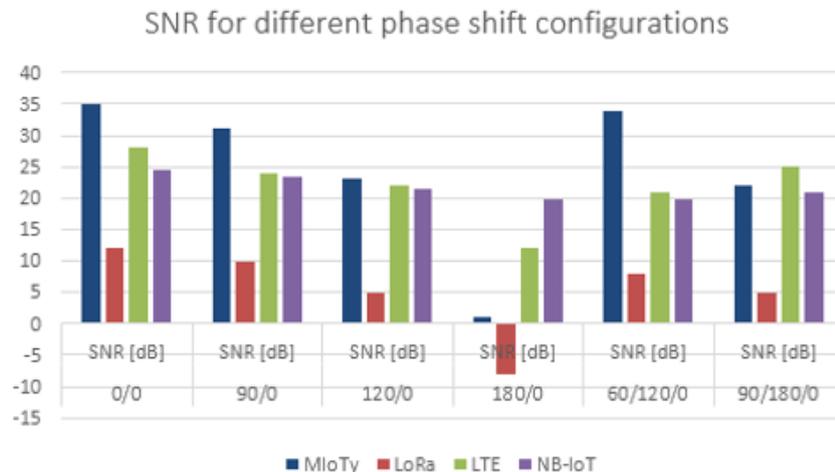


Abbildung 25: Vergleich des SNRs für unterschiedliche Phasenverschiebungen

von 180° ein starker Einbruch des SNRs beobachtet werden. Während dieser Test die Relevanz der Modellierung von Mehrwegekomponenten zeigt, so ist er nicht sehr nah an tatsächlichen Verhältnissen in einer Funkumgebung.

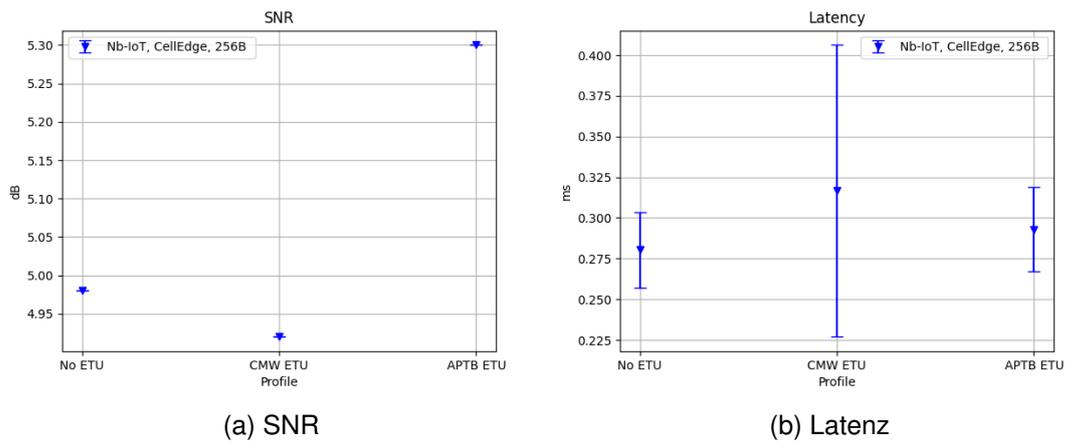
Der Mehrwege-APT-Bau ist jedoch nicht auf solche Tests beschränkt und kann auch für andere Szenarien eingesetzt werden, wie beispielsweise die automatische Umschaltung zwischen Pfadkombinationen. Dies kann beispielsweise für IoT-Szenarien mit langem Sendeintervall interessant sein, da hier Kanalbedingungen zwischen Sendezyklen sehr unterschiedlich sein können, beispielsweise weil der Hauptpfad von einer Maschine geblockt wird oder Wartungsarbeiten durchgeführt werden. Der APT-Bau ermöglicht es, mehrere Pfade aufzubauen und einzelne Pfade zuschaltbar zu machen. Der Verlauf der Signalstärke in Abhängigkeit der Konfiguration ist beispielhaft in Abb. 26 dargestellt. In diesem Test wurden verschiedene Pfadkombinationen der Reihe nach aktiviert und deaktiviert und die Signalstärke im Modem gemessen. Es sind deutliche Schwankungen zwischen den unterschiedlichen Pfadkombinationen zu erkennen.

Da es jedoch oftmals schwer ist, die relevanten Pfade für ein Szenario zu bestimmen, sind von der 3GPP bereits einige Szenarien definiert worden, die prinzipiell auch vereinfacht in unserem Aufbau verwendet werden können. In Abb. 27 ist ein Vergleich zwischen der Nutzung von Dämpfung, des externen ETU-Profiles sowie des internen Referenzprofils des CMW500 dargestellt. Hierbei wurde die Dämpfung jeweils so gewählt, dass ein Reference Signal Received Power (RSRP) von etwa -113 dBm am UE gemessen wurde. Es werden minimale Unterschiede am gemessenen SNR deutlich und insbesondere Unterschiede in der Latenz deutlich. Das interne Referenzprofil erhöht hierbei deutlich die Varianz und den Mittelwert gegenüber dem Aufbau mit nur Dämpfung. Das aufgebaute Profil erzeugt zwar eine leicht erhöhte Latenz, jedoch nicht vergleichbar mit dem Referenzprofil. Der Aufbau konnte auch in weiteren Tests nicht gegenüber dem Referenzprofil verifiziert werden. Ein Grund dafür ist, dass das interne



Abbildung 26: Vergleich des RSRPs für unterschiedliche Pfadkombinationen

Profil Dopplerverschiebungen und Korrelationen zwischen den Pfaden berechnet und somit mehr Details des Kanals beschreibt als über unsere Abstraktion möglich ist.



(a) SNR

(b) Latenz

Abbildung 27: Vergleich der ETU Profile

8. Erstellung der Auswertesoftware

Dieses Kapitel deckt die Arbeitsschritte von AP 5 ab. Das Arbeitspaket wurde Mehrheitlich von Hahn-Schickard bearbeitet, während ivESK abstimmand mitwirkte.

Das bloße Ausführen von Tests und Simulationen ist nicht ausreichend, um eine Aussage zu Funktechnologien und deren Anwendungsfällen zu treffen. Eine ausgiebige Auswertung und Dokumentation der durchgeführten Experimente ist von höchster Wichtigkeit. Die für TestKomm 4.0 entwickelte Auswertesoftware muss Daten aus unterschiedlichen Quellen darstellen und für eine Vielzahl von Projekten eingesetzt werden können. Das Ziel der Auswertungssoftware ist es dann, die Konfiguration der Experimente mitsamt ihrer Ergebnisse zu sichern und eine visuelle Darstellung zu generieren.

Im Folgenden werden die einzelnen technischen Schritte für die Entwicklung einer solchen Umgebung vorgestellt. Zunächst erklärt dieses Kapitel die Datenbank zum Abspeichern der Ergebnisse. Anschließend wird das gesamte System zum Speichern der Ergebnisse in der Datenbank gezeigt. Abschließend wird ein Beispiel gegeben, wie die Ergebnisse visualisiert werden.

8.1. Datenbank

Die vollständige, in TestKomm 4.0 entwickelte Testumgebung sammelt Ergebnisse aus verschiedensten Quellen und ermöglicht es diese auswerten. Dabei ist es nicht nur wichtig diese Daten zu sichern. Aufgrund der großen Menge an Tests müssen die Daten außerdem strukturiert und geordnet werden.

Die in Abbildung 28 vereinfachte Datenbankstruktur zeigt, wie in TestKomm 4.0 unterschiedliche Projekte und deren Ergebnisse strukturiert abgespeichert werden. Zunächst werden Projekte erzeugt. Diese bestehen lediglich aus einem Namen und einer Beschreibung. Die Projekte selbst sammeln alle für das jeweilige Projekt ausgeführten Tests.

Um die Testergebnisse einem Projekt zuzuordnen, gibt es die Unterprojekte. Diese Unterprojekte verfügen ebenfalls über einen Namen, sowie die für die Tests verwendeten Konfigurationen und deren Ergebnisse. Unterprojekte aggregieren Tests und Testergebnisse, die miteinander verglichen werden sollen. Beispielsweise können in einem Unterprojekt Tests aus unterschiedlichen Quellen gesammelt werden, die eine gemeinsame Grundkonfiguration haben und bei denen jeweils die selben Parameter variiert wurden. Dadurch lassen sich die Ergebnisse aus einer Simulation mit denen vom APTB und aus Feldtests vergleichen. Dies ermöglicht eine einfache Überprüfung von Hypothesen, die zunächst in Simulationen und anschließend im APTB und in Feldtests bestätigt werden können.

Die Struktur der Config- und Result-Attribute wurde so allgemein wie möglich gehalten, um die Visualisierung von Daten aus unterschiedlichen Datenquellen zu ermöglichen. Dies ist möglich durch die Verwendung von MongoDB⁷, einer NoSQL Datenbank ohne feste Strukturen. Damit die Visualisierung der Ergebnisse funktioniert, muss jedoch ein

⁷<https://www.mongodb.com>

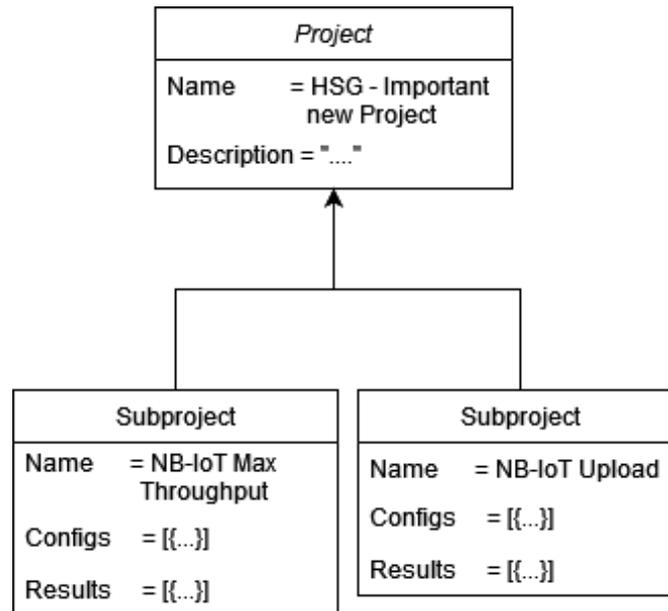


Abbildung 28: Vereinfachtes Beispiel wie Projekte und deren durchgeführte Experimente gesichert werden.

Mindestmaß an Struktur vorhanden sein. Die Herausforderung besteht darin ein Datenbankschema zu definieren, das flexibel genug ist, um Daten aus allen Quellen abzubilden und doch genügend strukturiert ist, um alle Daten gemeinsam zu visualisieren.

Die Datenbank ermöglicht es zwar nahezu beliebige Daten abzuspeichern, für die Visualisierung muss jedoch ein Datenformat zugrundegelegt werden. Das verwendete Datenformat ist in Abbildung 29 dargestellt. In der `Subproject`-Klasse sind nur die für die Visualisierung relevanten Attribute abgebildet. Zunächst muss festgelegt werden, welche Daten überhaupt für eine grafische Darstellung infrage kommen. Das sind alle Parameter, deren Wert zwischen den getesteten Konfigurationen in einem Unterprojekt variiert. Die Namen dieser Parameter werden als Array im Attribut `variables` definiert. Im Attribut `result` werden die Ergebnisse der Tests gespeichert.

Jedes Ergebnis hat drei Attribute: Die Datenquelle aus der das Ergebnis stammt (`source`), die verwendete Konfiguration (`config`) und die Ergebnisdaten (`data`). Die Datenquelle wird als Zeichenkette angegeben (z.B. `ns-3` oder `aptb`). Die Konfiguration besteht aus Schlüssel-Wert-Paaren, wobei der Schlüssel der Variablenname ist. Die Werte (in Abbildung 29 mit dem Typ `Parameter` bezeichnet) können Zeichenketten, Zahlen oder Booleans sein. Ähnlich sieht es bei den Ergebnissen aus. Die Schlüssel sind Namen von Messgrößen. Die Werte sind Objekte bestehend aus dem (in mehreren Messdurchläufen ermittelten) Mittelwert, einer Varianz und einem Konfidenzintervall (Konfidenzniveau 95%).

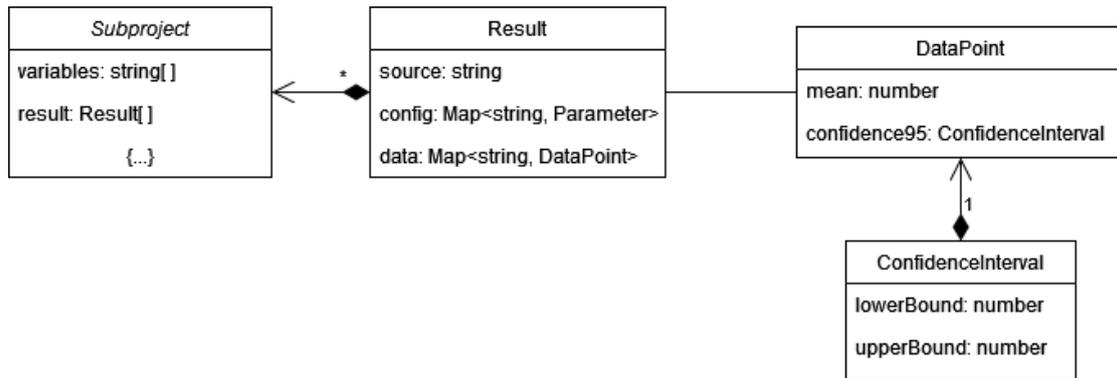


Abbildung 29: Datenformat der visualisierbaren Ergebnisse

8.2. Systemarchitektur

Abbildung 30 zeigt das System aus Abbildung 7 erweitert um die Visualisierungskomponenten. Es gibt drei verschiedene Benutzerschnittstellen. Über das Simulationsfrontend können Projekte und Unterprojekte angelegt und Simulationen definiert werden. Realisiert wird das Simulationsfrontend als Webanwendung. Das heißt, die Interaktion mit dem Frontend erfolgt über eine grafische Benutzeroberfläche in einem Webbrowser. Die eingegebenen Informationen zu Projekten und Simulationen werden an den Simulations- und Datenserver geschickt und von diesem in der Datenbank gespeichert. Der Simulations- und Datenserver nimmt außerdem Befehle zum Starten von Simulationen entgegen. Die Simulationen werden auf dem Simulations-Runner ausgeführt. Dieser ist über eine SSH-Schnittstelle mit dem Simulations- und Datenserver verbunden, über die auch die Simulationsergebnisse zurückgesendet werden. Der Simulations- und Datenserver speichert diese Ergebnisse ebenfalls in der Datenbank.

Der Simulations- und Datenserver stellt außerdem eine Schnittstelle für externe Daten und Projekte bereit. Dabei handelt es sich beispielsweise um Messungen vom APTB oder aus Feldtests.

Die Visualisierung wird ebenfalls browserbasiert als grafische Weboberfläche bereitgestellt. Die zu visualisierenden Daten holt sich die Webanwendung von einem Visualisierungsserver, der Zugriff auf die Datenbank hat. Die Bedienung der Visualisierungsanwendung wird im nächsten Abschnitt beschrieben.

8.3. Visualisierungsanwendung

Zum Betrachten der Ergebnisse im Visualisierungsfrontend muss zunächst das entsprechende Subproject ausgewählt werden. Die Abbildung 31 zeigt die einzelnen Schritte dafür. Zunächst muss das Projekt ausgewählt werden, in Abbildung 31a wird die Auswahl Ansicht dafür gezeigt. Es werden alle Projekte angezeigt, man kann die Projekte nach Namen filtern und neue Projekte anlegen. Nach der Auswahl des Projektes springt man in die Ansicht mit allen Unterprojekten des gewählten Projektes wie in Abbildung 31b zu

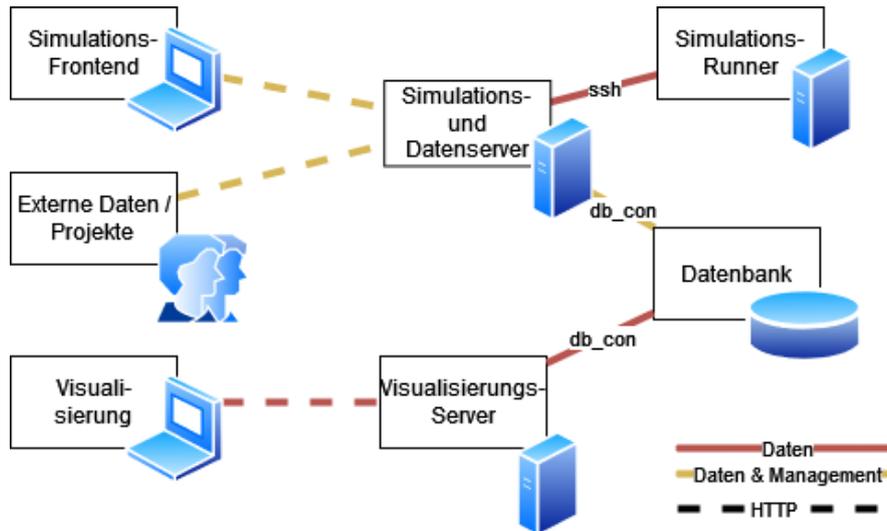


Abbildung 30: Systemansicht der Architektur für Projektmanagement und Visualisierung der Daten

sehen ist. Hier werden also alle ausgeführten Tests des Projektes aufgelistet.

Projects

Filter
Wifi simulations

Project	Description	Action
Wifi	Wifi simulations	

Items per page: 5 1 - 1 of 1 < >

[Create project](#)

(a) Auswahl des Projects

Wifi Subprojects

Filter

Subproject	Type	Action
wifi	NS3-Subproject	

Items per page: 5 1 - 1 of 1 < >

(b) Auswahl des Subprojects

Abbildung 31: Schritte zur Auswahl der einzelnen Tests

Nach der Auswahl des gewünschten Unterprojekts wird zunächst die Konfiguration des durchgeführten Experiments wie in Abbildung 32 angezeigt. Hier handelt es sich um die Konfiguration des ns-3 Projektes, welches in Listing 1 bereits vorgestellt wurde.

Das Visualisierungsfrontend bietet danach die Möglichkeit auszuwählen, welche Variable auf der X-Achse dargestellt wird. Hier wurde die *maxPacketSize_byte* ausgewählt.

Abbildung 33 zeigt die mit der Auswertesoftware abgespeicherten und verarbeiteten Ergebnisse der in Abbildung 32 dargestellten Simulation.

Zur Darstellung der Ergebnisse eines Unterprojekts fragt das Visualisierungsfrontend die Ergebnisse und Konfiguration des Unterprojekts beim Visualisierungsserver ab.

Das Visualisierungsfrontend zeichnet alle in der Datenbank gespeicherten Messun-

wifi
Wifi

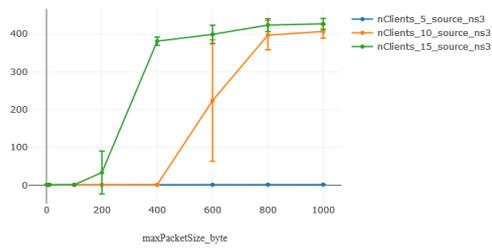
Name	Value
<i>repeatNumber</i>	5
<i>warmUpTime_s</i>	0
<i>path</i>	./result/
<i>saveRawDataLevel</i>	0
<i>simulationTime_s</i>	20
<i>nClients</i>	start: 5, end: 15, stepsize: 5
<i>maxPacketSize_byte</i>	1, 10, 100, 200, 400, 600, 800, 1000
<i>interPacketInterval_ms</i>	5

Abbildung 32: Kurzzusammenfassung der in Listing 1 beschriebenen Simulation.

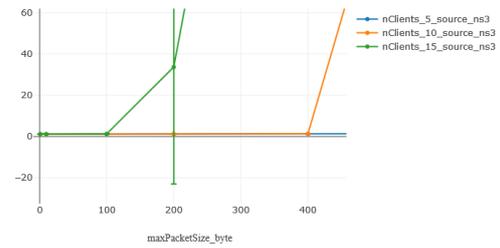
gen des Unterprojekts. Für jede Messgröße wird ein Diagramm erzeugt. Darin ist ein Graph für jede Konfiguration dargestellt, mit der eine Messung durchgeführt wurde. Zusätzlich wird nach der Quelle der Ergebnisse getrennt. Dadurch lassen sich beispielsweise Ergebnisse aus Simulationen mit Feldtests vergleichen, denen dieselben Konfigurationsparameter zugrunde lagen.

Im Beispiel sind lediglich drei der erzeugten Diagramme abgebildet. Die Abbildungen 33a und 33b zeigen die gleiche Messung. Jedoch ist 33b stark vergrößert – ein Feature des Visualisierungsfrontends. Diese Funktion ist wichtig für die Validierung von Ergebnissen. Ohne diese Funktion wäre bei der *nClients_5_source_ns3* Messung nicht erkennbar, dass der Messwert nicht 0 ist. Dabei ist hier die Latenz bei dieser Messung nur im normalen Bereich, während bei den anderen Experimenten vermutlich der Kanal überlastet ist (daher die deutlich höheren Latenzen). Die Ergebnisse aus 33c und 33d können diese These bestätigen. Da die Messungen zeigen, dass für *nClientes_5* die wenigstens Pakete verloren gehen und der Durchsatz am höchsten ist.

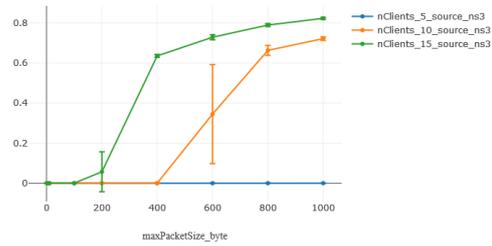
Mit diesem automatisierten Visualisierungswerkzeug können also Experimente effektiv und aussagekräftig ausgewertet werden. Sollten weitere Auswertungen beispielsweise für wissenschaftliche Veröffentlichungen oder zum Einbinden in andere Systeme notwendig sein, ist es möglich, alle Ergebnisse im maschinell lesbaren JSON-Format herunterzuladen.



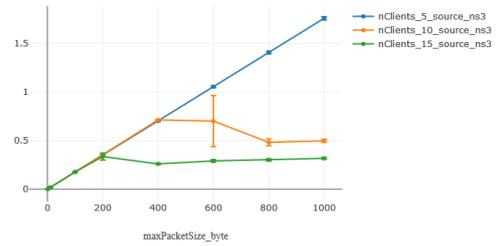
(a) One Way Latency in ms



(b) One Way Latency - Herangezoomt



(c) Paket Verlustrate in Prozent



(d) Durchsatz Kbit/s

Abbildung 33: Einige Ergebnisse der in Listing 1 beschriebenen Simulation

9. Aufbau und Beschreibung von Testfällen

Dieses Kapitel deckt die Arbeitsschritte von AP 4 und AP 6 ab und beschreibt die Testphasen, Testfälle, Testgeräte und den Testaufbau, sowie die Testausführung. Diese Arbeitspakete wurden maßgeblich von ivESK durchgeführt, während Hahn-Schickard abstimmend und unterstützend mitwirkte.

9.1. Testphasen

Die fokussierten Technologien wurden in mehreren Testkampagnen, unter Nutzung der entwickelten Testlösungen, getestet und evaluiert. Zur Untersuchung unterschiedlicher Aspekte der Technologien und Abstraktionsebenen wurden vier Testphasen definiert. In Tab. 16 sind die Testphasen innerhalb des Projektes dargestellt. Der Fokus der Tests liegt dabei auf der Uplink Kommunikation (Endgerät zur Basisstation), da dies die wichtigste Kommunikationsrichtung darstellt. Zu Beginn ist es besonders wichtig, sich mit den Möglichkeiten einer Technologie zunächst unter realen Bedingungen vertraut zu machen. Dieser Schritt dient als Referenz für die folgenden Schritte. Für die finalen Phasen wird in den abstrahierten Umgebungen gearbeitet, um mögliche Optimierungspotentiale aufzudecken.

Tabelle 16: Testphasen innerhalb des Projektes

Phase	Funkumgebung	Testzweck
Technologievergleich	Feld	Vergleich der Geräte unter realen Bedingungen – Was sind erwartbare Ergebnisse für die KPIs?
Abstraktionsebene	Feld, APTB, Simulation	Vergleich von KPIs für unterschiedliche Abstraktionsebenen – Wie und wodurch unterscheiden sich die Abstraktionsebenen?
Parametervariation	APTB, Simulation	Einfluss ausgewählter Parameter auf die KPIs
Optimierung	APTB, Simulation	Validierung von Parameterkombinationen für die Anwendungsfälle – Mit welchen Parametern lassen sich die KPIs optimieren?

9.2. Beschreibung von Testfällen

Das Ziel der Testfälle war es, die vorhandenen Technologien systematisch zu vergleichen sowie die Technologien auf unterschiedlichen Abstraktionsebenen zu vergleichen. Die so ermittelten Daten wurden als Grundlage benutzt, um die in Abschnitt 4 beschriebenen Anwendungsfälle zu validieren.

9.2.1. Testfälle

Ausgehend von den Anforderungen der Anwendungsfälle (Abschnitt 5.1 Tab. 11) konnten Testfälle abgeleitet werden, indem die relevante Kommunikationslogik, das relevante Verkehrsprofil, die Umgebungsbedingungen sowie die Netzwerkgröße ermittelt wurden. Je Kommunikationsmuster (Abschnitt 4.1 Tab. 4) wird ein neuer Testfall implementiert, der über die Konfigurationsparameter (siehe Abschnitt 7.1) parametrierbar ist. Der Zusammenhang zwischen Kommunikationsmuster und deren Relevanz für Anwendungsfälle ist in Tab. 17 dargestellt.

Tabelle 17: Kommunikationsmuster und deren Relevanz für Anwendungsfälle

Kommunikation	Beschreibung	Anwendungsfall
Uplink	ED sendet Daten an einen Server	FA1, PA, M2M
Downlink	Server sendet Daten an ein ED	PA, M2M
Uplink Burst	ED sendet große Menge an Daten an einen Server	PA2
Firmware-Update	Server sendet eine große Menge an Daten an ein ED	PA, M2M
Uplink triggert Downlink	ED sendet Daten an einen Server, der unverzüglich antwortet	FA
Downlink triggert Uplink	Server sendet Daten an ein ED, das unverzüglich antwortet	PA, M2M

Für die Kommunikationsmuster ergeben sich unterschiedliche Verkehrsparameter, die über die Datenstruktur in Listing 5 abgebildet werden kann.

Listing 5: Datenstruktur zur Beschreibung von Verkehrsparametern

```
type record PacketParameter {
    integer num_packets,
    integer packet_size,
    integer response_packet_size,
    float transmit_int,
    IpParameter ip_param optional
}
```

9.2.2. Parametrierung

Die Testfälle sind über zahlreiche Parameter konfigurierbar (vgl. Abschnitt 7.1.3). Erst über die konkrete Parametrierung ergibt sich ein konkreter Nutzen aus den Tests. Für die Testfallbeschreibung wird schließlich definiert, welche Werte konstant gehalten werden und welche variiert werden. Daher wird für die Beschreibung der Tests ein Testbeschreibungsschema wie in Tab. 18 benutzt um einen Test vollständig zu beschreiben.

Tabelle 18: Tabellarisches Testbeschreibungsschema

Parameter	Beschreibung
Testfall ID	Eindeutige Identifikation der Testausführung
Testzweck	Was sind die Ziele? Welche Fragestellung soll durch die Testausführung beantwortet werden?
Relevanz	Für welche Technologien und welche Anwendungsfälle ist der Testfall relevant?
Testfall	Referenz auf den Testfall, sowie die Parametrierung dieses
Variablen	Auflistung der variablen Parameter und deren Werte(bereiche)n
Relevante KPIs	Auflistung der relevanten KPIs
Erwartete Ergebnisse	Beschreibung, was erwartet wird und wie es verwendet werden soll, z.B. die Art von Diagramm

9.3. Testaufbau

Auf den unterschiedlichen Abstraktionsebenen gab es klare Unterschiede, die beachtet werden mussten. Während sich Verkehrsparameter und Netzwerkparameter aus den Anforderungen ableiten ließen, musste für die Funkumgebung auf allen Ebenen eine Abstraktion vorgenommen werden, wie in Abb. 34 dargestellt. Da die von uns beschriebenen Anwendungsfälle keiner konkreten Anwendung entsprachen, gab es auf der Feldebene kein Äquivalent der Funkumgebungen. Die Funkumgebung wurde nach Tab. 11 nach deren Größe und deren charakteristischen Eigenschaften eingeordnet. Somit ließen sich ähnliche Punkte in einer Ersatz-Feldtestumgebung finden, wie in diesem Projekt auf dem Campusgelände der Hochschule Offenburg.

In der Emulationsumgebung wurde mit Dämpfungsgliedern gearbeitet, um Bedingungen wie in der Feldtestumgebung zu erzeugen. In der Simulation wurde auf Kanalausbreitungsmodelle zurückgegriffen, welche i.d.R. das Einstellen von Distanz, Höhe und Modell erlauben, woraus ein Pfadverlust berechnet wird.

Für die untersuchten Anwendungsfälle sind meist nur wenige Endgeräte von Interesse oder so viele, dass dies nicht mit der vorhandenen Hardware getestet werden kann. Daher wurden für die Tests auf Feld- und Emulationsebene, Tests mit einem Endgerät durchgeführt. In der Simulationsumgebung wurde schließlich untersucht, wie viele Endgeräte insgesamt unterstützt werden können. Der generelle Testaufbau ist in Abb. 35 dargestellt. Die Endgeräte und die Basisstation werden vom Testsystem gesteuert, und entsprechend die erforderlichen Daten im Lauf des Tests gespeichert.

Bei der Durchführung der Parametervariationen wurde zusätzlich der Energiebedarf

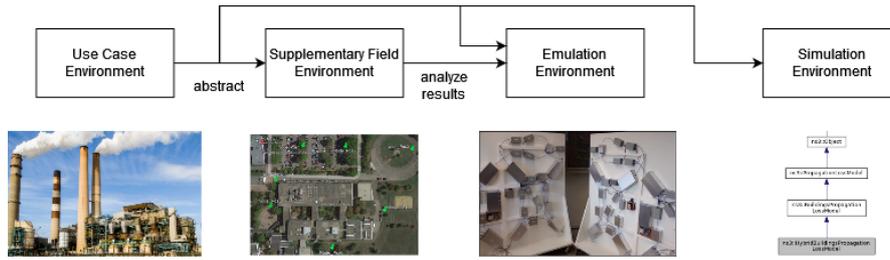


Abbildung 34: Konzept der Umgebungsabstraktion

erfasst. Hierfür wurde ein Power Analysator (Power Profiler Kit II [65]) verwendet, um die Endgeräte mit Energie zu versorgen und gleichzeitig deren Bedarf zu erfassen.

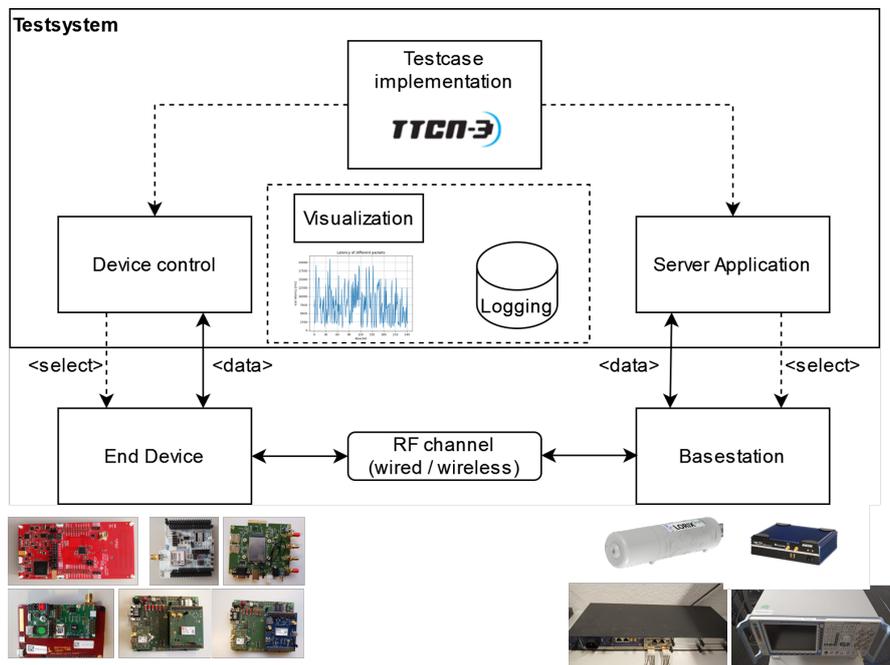


Abbildung 35: Testaufbau mit einem Endgerät

9.3.1. Eingesetzte Geräte

In Tab. 19 ist eine Übersicht zu der eingesetzten Hardware dargestellt.

Tabelle 19: Eingesetzte Hardware

Technologie	Basisstationen	Endgeräte
NB-IoT	Amarisoft basierte SDR-Lösung (NAT-AMC-ZYNQUP-SDR) CMW500	Commsolid Gr581 (Cert-Board) UBlox Sara R510S (Eval-Board) UBlox Sara N310 (Eval-Board)
LTE	Amarisoft basierte SDR-Lösung (NAT-AMC-ZYNQUP-SDR) CMW500	Quectel RM 500Q-GL (Eval-Board)
LoRa	Lorix One	STM32WL55JC1 + LoRa Eval Stack (Eval-Board)
MIOTY	MIOTY M500	Launchxl-CC1312R1 + MIOTY Eval Stack (Eval-Board)

Für die zellulären Lösungen wurde mit einer Amarisoft basierten SDR gearbeitet. Nur beim Test mit variierenden Parametern wurde der CMW500-Kommunikationstester verwendet.

9.3.2. Feldtestumgebung

Um möglichst realitätsnahe Umgebungen zu erzeugen, wurden mehrere Testpunkte auf dem Campusgelände der Hochschule Offenburg identifiziert, wie in Abb. 36 dargestellt. Neben Indoor-Bedingungen, wurde eine Funkstrecke vom Innern eines Gebäudes nach außen benutzt um Freifeld- und Abschattungseffekte mit einzubeziehen. Aufgrund des industriellen Fokus der Anwendungsfälle wurde zudem in den Industrielaboren der Hochschule Offenburg gemessen.

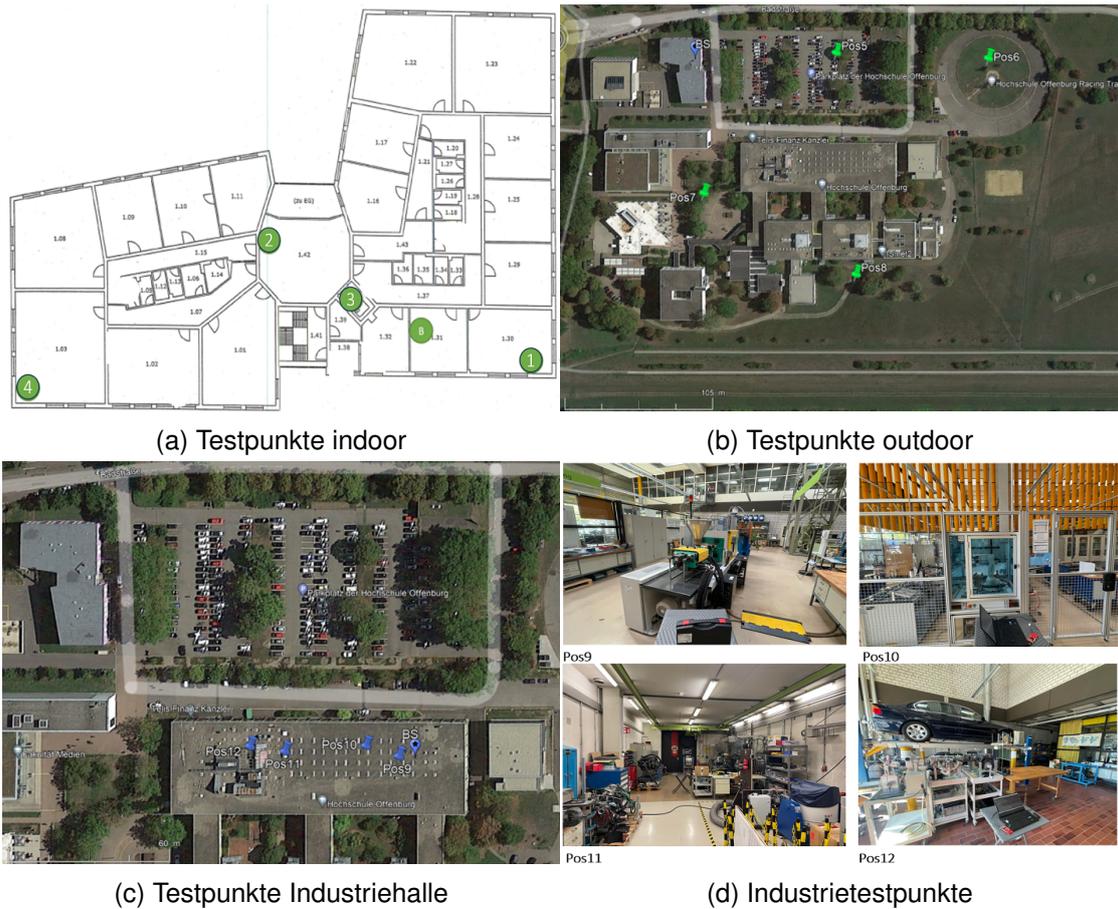


Abbildung 36: Feldtestpunkte

Ein Überblick über alle Testpunkte ist in Tab. 20 dargestellt.

Tabelle 20: Charakterisierung der Feldtestpunkte

Position	Distanz [m]	Charakterisierung
1	10	Gute Bedingungen, NLOS, Indoor, Bürowände zwischen Teilnehmern
2	15	Gute Bedingungen, NLOS, Indoor, Bürowände zwischen Teilnehmern, Kommunikation über Reflektionen möglich
3	10	Mittlere Bedingungsgüte, NLOS, Indoor, Aufzug in anderem Stockwerk, Schirmung durch Metall
4	35	Mittlere Bedingungsgüte, NLOS, Indoor, Mehrere Bürowände zwischen Teilnehmern
5	100	Gute Bedingungen, LOS, Outdoor, Parkplatz, Große Ausbreitungsfläche
6	200	Gute bis mittlere Bedingungen, NLOS, Outdoor, Leichte Abschattung durch Bäume, Große Ausbreitungsfläche
7	100	Schlechte Bedingungen, NLOS, Outdoor, Starke Abschattung durch Stahlbetonwände von Gebäuden
8	200	Sehr schlechte Bedingungen, NLOS, Outdoor, Viele Gebäude zwischen der Kommunikationsverbindung
9	10	Gute Bedingungen, LOS, Industrie, Direkte Sichtverbindung, leichte Schirmung und Reflektionen von nahestehenden Maschinen
10	20	Gute Bedingungen, NLOS, Industrie, ED in der Nähe von Fertigungsroboter, Abschirmung durch Trennwand und Stahlkäfig
11	55	Schlechte Bedingungen, NLOS, Industrie, Abschirmung durch mehrere Trennelemente und Motorenprüfstandkammer
12	65	Sehr schlechte Bedingungen, NLOS, Industrie, Abschirmung durch mehrere Stahlbetonwände, Trennelemente und Motorenprüfstand

9.3.3. Emulationsumgebung

Zum Vergleich der Technologien zwischen den Umgebungen wurde das APTB verwendet. Hierbei wurden ein Endgerät und eine Basisstation jeweils in einer geschirmten Box untergebracht und mit dem HF-Interface des APTB verbunden. Der Kanal wurden schließlich so angepasst, dass der im Feld gemessene RSRP erreicht war.

Aufgrund der abstrakten Anwendungsfälle und der Vielzahl von möglichen Umgebungsbedingungen wurden die Feldtestbedingungen in der Hochschulumgebung nicht modelliert, sondern stattdessen auf definierte Bedingungen zurückgegriffen. Das 3GPP hat Kanalbedingungen für unterschiedliche Szenarien definiert. Effekte wie Verzögerungen und Mehrwegeausbreitung spielen hierbei eine herausragende Rolle. Folgende Profile wurden definiert [66]:

- Extended Pedestrian A model (EPA)
- Extended Vehicular A (EVA)
- Extended Typical Urban (ETU)

Für unsere Anwendungsfälle wurde das ETU Profil ausgewählt, das über Verzögerungszeiten und relative Leistung definiert ist. Durch die Verwendung von Koaxialkabeln vom Typ LMR-195 und Dämpfungsgliedern konnte das Profil nachgestellt werden. Das Profil, sowie die Kabellängen und die verwendeten Dämpfungen ist in Tab. 21 dargestellt. Hierbei ist anzumerken, dass die letzten drei Pfade des ETU-Profiles aufgrund der zu hohen Kabeldämpfung nicht umgesetzt werden konnten. Die Kabel wurden separat in Bezug auf Verzögerung, Phasenverschiebung und Dämpfung für verschiedene Frequenzen vermessen, sodass neben der Einstellung des ETU Profils auch gezielt bestimmte Phasenverschiebungen oder Verzögerungen eingestellt werden können. Insbesondere ist es durch den Aufbau möglich, zwischen verschiedenen Pfaden umzuschalten.

Tabelle 21: ETU Profil

Verzögerung [ns]	Relative Leistung [dB]	Kabel-länge [m]	Zusätzliche Dämpfung [dB]
0	-1	0,5	36
50	-1	12	36
120	-1	30	30
200	0	50	26
230	0	55,6	20
500	0	120	0
1600	-3	-	-
2300	-5	-	-
5000	-7	-	-

9.3.4. Simulationsumgebung

Auf der Simulationsebene können Teilnehmer in einem einstellbaren Radius um eine Basisstation herum platziert werden. Hierbei ist die Höhe der Teilnehmer und der Basisstation einstellbar.

Um Umgebungsbedingungen zu modellieren wurden verschiedene Kanalausbreitungsmodelle und Fading-Modelle eingesetzt, um eine realitätsnahe Funkkommunikation zu modellieren:

- Friis: Für Szenarien mit LOS Eigenschaften
- COST231: Für Szenarien, die im urbanen Raum auftreten
- FixedRssLoss: Zur Einstellung eines fixen RSRPs

Das Friis Modell ist ein einfaches Modell für die Beschreibung der Funkausbreitung im freien Raum und wird über folgenden Zusammenhang beschrieben:

$$P_r = P_t \cdot G_t G_r \cdot \frac{\lambda^2}{(4\pi d)^2 L}$$

Es wird deutlich, dass die Empfangsleistung P_r proportional zur Sendeleistung P_t und den Antennengewinnen auf Sender- und Empfängerseite (G_t , G_r) ist, und reziprok proportional zur quadratischen Entfernung und Frequenz ist.

Das COST231-Modell ist eine Erweiterung des Hata-Modells [67] und basiert auf empirischen Studien im urbanen Bereich. Es gibt zahlreiche weitere empirische Modelle, die verschiedene Szenarien realitätsgetreu nachbilden zu versuchen [36]. In diesem Projekt wird hauptsächlich das COST231 benutzt.

Um wie auch in der Emulationsumgebung Mehrwegeeffekte zu berücksichtigen, wird ein Fading-Modell eingesetzt. Hier wird das ETU Profil von 3GPP benutzt. Dafür wurden in ns3 bereits Traces in Matlab vorberechnet und als Datei im Simulationsmodell hinterlegt. Dies sorgt dafür, dass die Signalamplitude nicht konstant ist, sondern über Zeit und Frequenz variiert. Das Spektrum dieses Profils ist in Abb. 37 dargestellt.

Die wichtigsten Rahmendaten für die Konfiguration der Simulation ist wie folgt an den Feldtestbedingungen orientiert:

- Sendeleistung eNB, UE: 23 dBm
- Ausbreitungsmodell *COST231*
- UE-Höhe: 2 m
- eNB-Höhe: 5 m
- ul earfcn: 5790
- dl earfcn: 23 790

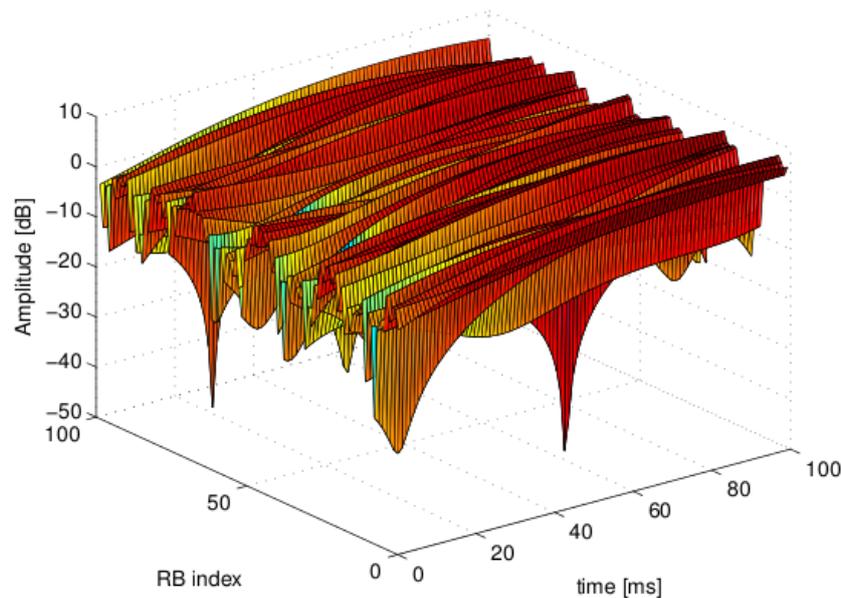


Abbildung 37: Signalleistung über Ressourcenblöcke und Zeit des simulierten ETU Profils [68]

Bei den Simulationen ergibt sich je nach eingestellter Distanz ein gewisser SNR, je nach eingestellter Sendeleistung, Ausbreitungsmodell, Modell usw. Für die oben erläuterte Konfiguration ergeben sich die folgenden Distanzen für NB-IoT, wobei zu beachten ist, dass die Distanz insbesondere stark von der Höhe der Antenne der Basisstation abhängt:

- Max. Reichweite
 - Uplink: 340 m (SNR: -17.4 dB)
 - Downlink: 160 m (SNR: -4.2 dB)
- Zellengrenze
 - Uplink: 250 m (SNR: -12 dB)
 - Downlink: 125 m (SNR: 0 dB)
- Normal
 - Uplink: 50 m (SNR: 16 dB)
 - Downlink: 30 m (SNR: 16 dB)

Für LTE hingegen ergeben sich folgende Zusammenhänge:

- Max. Reichweite: 165 m (SNR: -5 dB)

- Zellengrenze: 125 m (SNR: 0 dB)
- Normal: 50 m (SNR: 16 dB)

Im Vergleich zu LTE sind die Reichweiten in Uplink und Downlink symmetrisch.

9.4. Testsystematik

Nachfolgend werden die jeweiligen Testphasen sowie die Systematik zur Ausführung der Tests dargestellt.

9.4.1. Technologievergleich

Zu Beginn wurde ein Technologievergleich unter identischen Bedingungen durchgeführt. Diese Evaluierung konnte einen Eindruck zum Verhalten der Technologien unter realen Bedingungen geben. Gleichzeitig konnte so das unterschiedliche Grundverhalten der Technologien herausgearbeitet werden und deren Unterschiede verstanden werden. Schließlich konnte bereits in diesem Schritt ein erster Vergleich der technologischen Leistungsfähigkeiten erstellt werden. Da nicht für alle Technologien Simulationsmodelle vorhanden waren, wurde ein Technologievergleich auf Feld- und Emulationsebene durchgeführt. Die relevanten Testpunkte sind in Abschnitt 9.3.2 beschrieben.

Für diesen Test, wurden 100 Pakete mit 10 B und 50 B Payload Größe in einem Intervall von 10 s gemäß dem *Uplink*-Testfall (vgl. Abschnitt 9.2.1) gesendet. Die Parameter wurden so gewählt, um ein statistisch relevantes Ergebnis zu erhalten, typische Nutzdatengrößen abzudecken bei gleichzeitiger Unterstützung aller Protokolle ohne den Funkkanal auszulasten.

9.4.2. Abstraktionsebenen

Die Besonderheit des entwickelten Testsystems ist die einheitliche Testausführung auf mehreren Abstraktionsebenen.

Um die Leistung der vorgeschlagenen einheitlichen Methodik und der implementierten Testumgebung auf mehreren Abstraktionsebenen zu bewerten, wurde ein einheitlicher Testfall in allen Umgebungen ausgeführt.

Wie in Tab. 22 dargestellt, wurden Feldtests an verschiedenen Positionen durchgeführt und die entsprechenden Signalqualitätsparameter und HF-Effekte im APTB emuliert und mit den ns-3-Simulationsmodellen simuliert. Zur Reproduktion von Feldtestergebnissen im APTB wurde der RSRP so eingestellt wie der Median-Wert in den Feldtests.

Tabelle 22: Vergleich der Einstellung der Umgebungsbedingungen in unterschiedlichen Abstraktionsebenen

Position	Position 1	Position 2	Position 3	Position 4
Feld	100 m LOS	200 m LOS	100 m NLOS	200 m NLOS
Emulation	RSRP: -80 dBm	RSRP: -90 dBm	RSRP: -95 dBm	RSRP: -105 dBm
Simulation	100 m Friis	200 m Friis	100 m COST231	200 m COST231

Es wurde ein periodisches Uplinkverhalten als Test gewählt, um das typische funktionale Verhalten der periodischen Uplink-Übertragung durch die erfolgreiche Verbindung von einem ED und einer Basisstation (BS) zu verifizieren, gefolgt vom Senden von UDP-Paketen und der Messung der Leistung von KPIs auf drei verschiedenen Abstraktionsebenen (Simulation, Emulation und Feldtests) für die NB-IoT-Technologie.

Für diesen Test wurden 100 Pakete mit 10 B und 50 B Payload Größe in einem Intervall von 10 s gemäß dem *Uplink*-Testfall gesendet (vgl. Abschnitt 9.2.1).

9.4.3. Parametervariationen

Technologievergleiche und unterschiedliche Abstraktionsebenen helfen bei der Technologieauswahl. Zusätzlich ist es von hoher Relevanz, zu untersuchen wie sich die Parametrierung der Netze auf deren Leistungsfähigkeit auswirkt.

In dieser Testphase wurden die Netzwerkparameter einer Technologie systematisch variiert und deren Einfluss auf die Leistungsparameter gemessen. Dabei wurde neben der Emulationsumgebung auch, sofern möglich, die Simulationsumgebung verwendet. Neben einem tieferen Verständnis für die Protokolle und deren Konfigurationsmöglichkeiten diente dieser Schritt als Grundlage, um Aussagen über optimierte Parametersätze für bestimmte Anwendungen zu treffen.

Es wurden Testfällen für die periodische Uplink- und die periodische Downlink-Kommunikation betrachtet.

Die Umgebungsbedingungen spielen eine wichtige Rolle bei der drahtlosen Kommunikation. Für eine systematische Untersuchung, wurden drei Umgebungsbedingungen definiert und in den Tests benutzt. Das Sensitivitätslimit der Technologien wurde dabei in praktischen Untersuchungen mit den Geräten ermittelt.

- Max. Reichweite (Max Coverage): Sensitivitätslimit ist erreicht
- Zellengrenze (Cell Edge): 5 dB Abstand zur Sensitivitätsgrenze
- Normal:: Mittlere Signalstärke auf der Sensitivitätsskala

Da der Energiebedarf für viele Anwendungen relevant ist (vgl. Abschnitt 4), wurde in dieser Untersuchung auch der Energiebedarf für die EDs gemessen werden.

Für die Betrachtung des Energiebedarfs wurden die Ströme mit einem Leistungsanalysator aufgezeichnet sowie die Ströme, Zeiten und Energiebedarfe unterschiedlicher Zustände erfasst. Das 3GPP gibt eine Lebensdauer von 10 Jahren, bei einer Batteriekapazität von 5 Wh und einem Sendeintervall von 24 h und 200 B an [69]. Abschätzungen zur Lebensdauer beziehen sich daher auf die erfassten Ströme und unterschiedliche periodische Sendeszenarien sowie eine 5 Wh Batterie, die über einen Sicherheitsfaktor von $SF_{bat} = \frac{1}{3}$ abgeschätzt wurde, wie in [34] beschrieben.

Nachfolgend sind die zu variierenden Parameter je Technologie dargestellt. Für jede Technologie ergeben sich verschiedene Parameter, sodass die entsprechenden Parameter und deren Variation als Teil der Ergebnisbeschreibung beschrieben werden.

In Tab. 23 sind die variierten Parameter sowie die zu beantwortenden Fragen für NB-IoT dargestellt.

Tabelle 23: Parametervariation für NB-IoT

Parameter	Abstraktion	Zielfragestellung
Paketgröße	Simulation, Emulation	Ist es besser, mit größeren Paketen zu senden? Gibt es Einschränkungen bezüglich des Sendeintervalls?
HARQ	Emulation, Simulation	Kann die Reichweite erhöht werden?
Ressourcen	Emulation	Wie wirken sich die vorhandenen Netzwerkressourcen auf die Performanz aus?
Sendewiederholungen	Emulation	Wann lohnt es sich Sendewiederholungen einzusetzen?
Leistung	Emulation	Wann lohnt es sich die Leistung zu verringern?
RAI	Emulation	Wie weit kann der Energiebedarf durch das Feature verringert werden?
PSM	Emulation	Wie gering ist der Energiebedarf? Für welche Szenarien kann es eingesetzt werden?
Sendezustände	Emulation	Sind gewisse Protokollzustände beim Senden von Daten zu favorisieren?
Kommunikationsmuster	Simulation	Wie beeinflusst das Kommunikationsmuster die Performanz?
Teilnehmer	Simulation	Ab wann treten Paketverluste auf? Wie viele Teilnehmer können unterstützt werden?

In Tab. 24 sind die variierten Parameter sowie die zu beantwortenden Fragen für LTE dargestellt.

Tabelle 24: Parametervariation für LTE

Parameter	Abstraktion	Zielfragestellung
Paketgröße	Emulation, Simulation	Ist es besser, mit größeren Paketen zu senden? Gibt es Einschränkungen bezüglich des Sendeintervalls?
HARQ	Emulation, Simulation	Kann die Reichweite erhöht werden?
Ressourcen	Emulation	Wie wirken sich die vorhandenen Netzwerkressourcen auf den Durchsatz aus?
Bandbreite	Emulation, Simulation	Wie wirkt sich die Bandbreite auf den Durchsatz und die Reichweite aus?
Leistung	Emulation	Wann lohnt es sich die Leistung zu verringern?
Modulation	Emulation	Wie wirkt sich die Modulation auf den Durchsatz aus?

In Tab. 25 sind die variierten Parameter sowie die zu beantwortenden Fragen für LoRa dargestellt.

Tabelle 25: Parametervariation für LoRa

Parameter	Abstraktion	Zielfragestellung
Paketgröße	Emulation	Ist es besser, mit größeren Paketen zu senden? Gibt es Einschränkungen bezüglich des Sendeintervalls?
Leistung	Emulation	Wann lohnt es sich die Leistung zu verringern? Wie weit kann der SNR erhöht werden?
Spreizfaktor	Emulation	Wie weit kann die Latenz reduziert werden? Welchen Einfluss hat dies auf die Sendeintervalle?
Sende- wiederholungen	Emulation	Wann lohnt es sich Sendewiederholungen einzusetzen?
ACK	Emulation	Was ist der Einfluss auf den Energiebedarf und mögliche Szenarien?

In Tab. 26 sind die variierten Parameter sowie die zu beantwortenden Fragen für MIOTY dargestellt.

Tabelle 26: Parametervariation für MIOTY

Parameter	Abstraktion	Zielfragestellung
Paketgröße	Emulation	Ist es besser, mit größeren Paketen zu senden? Gibt es Einschränkungen bezüglich des Sendeintervalls?
Leistung	Emulation	Wann lohnt es sich die Leistung zu verringern?
Modi	Emulation	Wie wirken sich die vorhandenen Sendemodi auf die Performanz aus?
Empfangs- fenster	Emulation	Was ist der Einfluss auf den Energiebedarf und mögliche Szenarien?

9.4.4. Optimierung

In dieser Testphase wurden die Ergebnisse der Parametervariationen auf die ausgewählten Anwendungsfälle bezogen, um eine geeignete Konfiguration finden zu können. Es wurden schließlich die relevanten Testfälle mit den konkreten Parametern der Anwendungsfälle durchgeführt, um die Nutzbarkeit der Technologien für einen bestimmten Anwendungsfall zu evaluieren.

9.5. Testausführung

Für die Durchführung von Testfällen wurde das Testsystem verwendet. Bei unterschiedlichen Ausführungsebenen wurden dieselben Parameter verwendet. In Abb. 38 ist ein Aktivitätsdiagramm dargestellt, das den Gesamtablauf des Testprozesses darstellt. Für die Analyse der Daten wurde auf die erfassten Datensätze bzw. Visualisierungen zurückgegriffen. Je nach Testebene waren außerdem weitere Daten verfügbar, die zusätzliche mit analysiert werden können, wie Netzwerklogs, Endgeräte-Logs und die Logs des Testsystems.

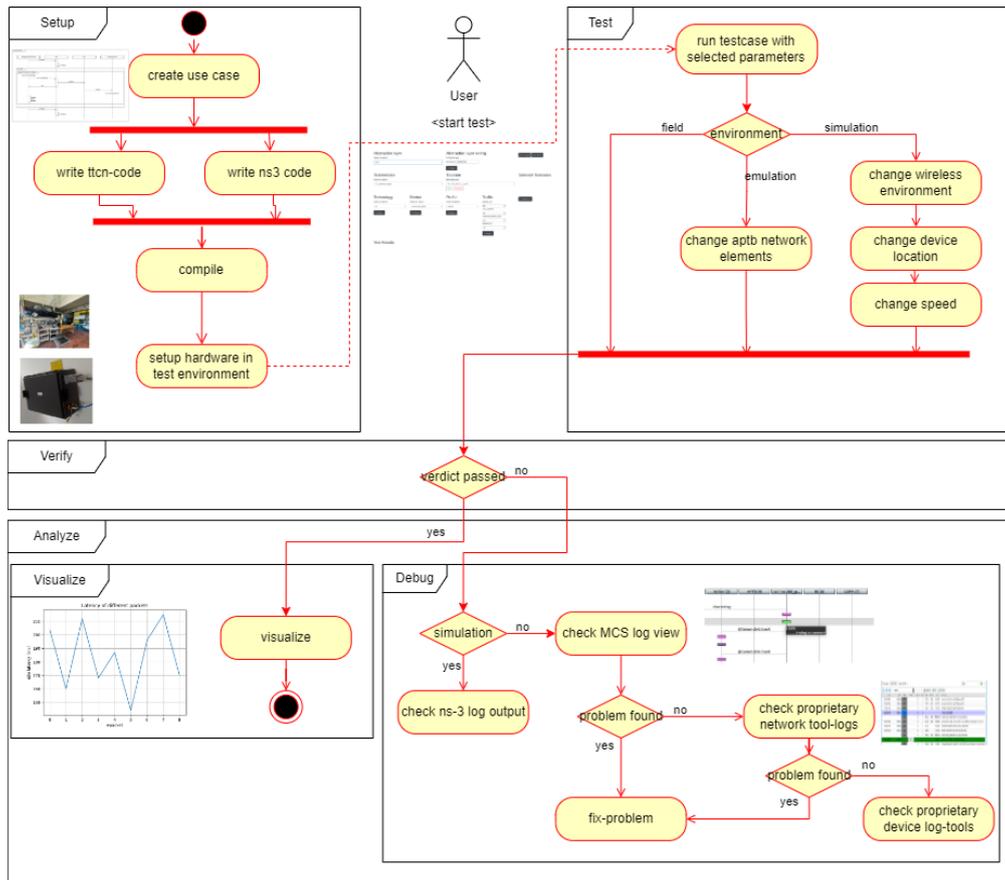


Abbildung 38: Aktivitätsdiagramm für den Testablauf auf unterschiedlichen Ebenen

10. Testergebnisse

Dieses Kapitel deckt die Arbeitsschritte von AP 7 und AP 8 ab, in denen zahlreiche Testkampagnen definiert und mit den entwickelten Komponenten durchgeführt wurden, um ausgewählte Technologien und Anwendungsfälle zu evaluieren. In diesem Abschnitt werden die wichtigsten Resultate der in Abschnitt 9 Tab. 16 beschriebenen Testkampagnen vorgestellt.

Diese Arbeitspakete wurden maßgeblich von ivESK durchgeführt. Hahn-Schickard unterstützte vor allem bei der Ausführung und Analyse von Simulationen.

10.1. Technologievergleich

In diesem Abschnitt werden die Ergebnisse des Technologievergleichs dargestellt. Der Testaufbau sowie die Feldtestpunkte können in Abschnitt 9.3.2 betrachtet werden.

Aufgrund der Vielzahl durchgeführter Tests und Visualisierungen ist hier nur ein Ausschnitt zu den wichtigsten Ergebnissen dargestellt. Die Leistungsparameter wurden vom Testsystem gemessen und visualisiert (vgl. Abschnitt 3.4.2).

Genauere Untersuchungen zu den Auswirkungen der unterschiedlichen Konfigurationen sind in Abschnitt 10.3 dargestellt.

Wie in Abschnitt 9.3.2 erläutert wurde dieser Testfall für die Erfassung des Grundverhaltens der Technologien in realen Umgebungen verwendet. Es wurde der *Uplink*-Testfall durchgeführt, um die Leistungsparameter an unterschiedlichen Feldtestpunkten zu ermitteln. Zur Charakterisierung der Kanalbedingungen wurden die Signalstärken an den Testpunkten gemessen (vgl. Abschnitt 3.4.2).

In Abb. 39 sind die Signalstärken an unterschiedlichen Indoor-Positionen für unterschiedliche Technologien dargestellt. Anhand der Grafik wird deutlich, dass für die Technologien an einer Position unterschiedliche Signalstärken gemessen werden, beispielsweise aufgrund anderer Bandbreite, Modulation oder Rauschzahl. Gleichzeitig wird deutlich, dass beim Wechsel zwischen Positionen nicht von einer Technologie auf eine andere geschlossen werden kann. Beispielsweise fällt beim Wechsel von Position 1 auf 2 auf, dass sich die Signalstärke für die zellulären Technologien erhöht, während sie bei den beiden LPWA Lösungen fällt. Grundsätzlich sind bei den LPWA Lösungen auch deutlichere Schwankungen der Signalstärke erkennbar (bei einem (relativ) statischen Kanal). Paketverluste traten meist bei Ausreißern des SNRs auf, wodurch das Signal nicht mehr richtig dekodiert werden konnte. Es konnten aber auch speziell bei LoRa Paketverluste unter vergleichsweise guten Bedingungen festgestellt werden. Eine Korrelation zwischen Latenz und dem gemessenen SNR bzw. den Umgebungsbedingungen konnte jedoch nicht hergeleitet werden.

Als praktischen Nutzen kann daraus gezogen werden, dass Tests zur Charakterisierung der Umgebungsbedingungen und der Geräteempfindlichkeiten immer mit der konkret einzusetzenden Hardware unter den konkreten Umgebungsbedingungen durchgeführt werden müssen.

Die Ergebnisse in Bezug auf Latenz und Paketverlust über alle 12 Feldtestpositionen sind in Tab. 27 zusammengefasst.

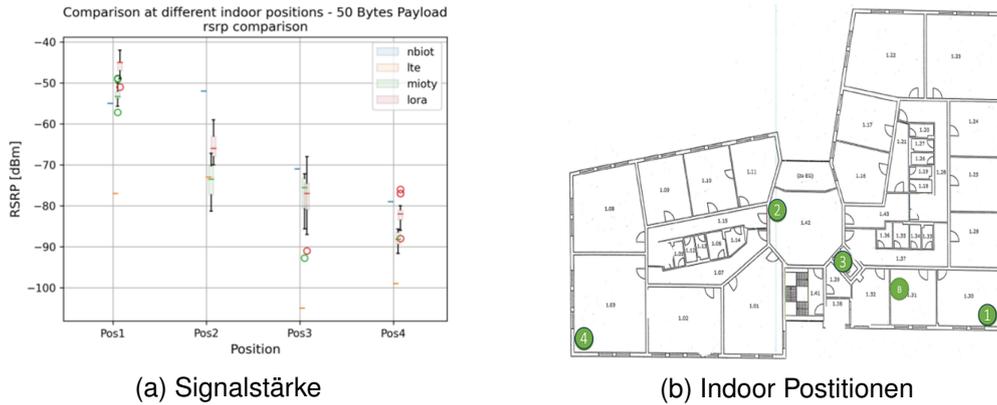


Abbildung 39: Vergleich der Signalstärke an unterschiedlichen Indoor Positionen

Tabelle 27: Zusammenfassung der Feldtestergebnisse für Latenz und Paketverlust

Technology	#Bytes	E2E-Latenz [ms]				Verlust
		mean	std	min	max	
NB-IoT	10	1320	450	113	2227	nein
	50	1344	454	256	3009	
LTE	10	285	205	13	1965	ja
	50	340	282	19	2419	
Mioty	10	3867	36	3781	3973	ja
	50	10544	107	10255	10811	
LoRa	10	444	60	405	1231	ja
	50	872	169	775	1791	

Es fällt insbesondere die hohe Standardabweichung bei der Latenz für die zellulären Technologien auf. Die Ausreißer bei den zellulären Technologien konnten im späteren Verlauf als ein Problem interner Empfangsfilter innerhalb der SDR identifiziert werden. Durch diesen Umstand kam es im Basisband zu Störungen, wodurch insbesondere die Random Access Requests fehlschlugen, und erst über ein entsprechendes Power-Ramping konnten Ressourcen von den eNB angefordert werden. Dies hat stellenweise zu sehr hohen Ausreißern geführt. Dies zeigt die Bedeutung von Hardwaretests und den Zustand aktueller Implementierungen von privaten zellulären Netzwerken.

10.2. Abstraktionsvergleich

In diesem Abschnitt werden ausgewählte Testergebnisse vorgestellt, die auf unterschiedlichen Ebenen, unter den in Abschnitt 9.4.2 erläuterten Konfigurationen, ermittelt worden sind.

Die Ergebnisse in Abb. 40 zeigen die Auswirkungen der Abstraktion von Funk- und Kanaleigenschaften in der Simulation. Es wurden große Unterschiede in den Simulationsergebnissen festgestellt, die hauptsächlich auf die Abstraktionen zurückzuführen sind. Der Jitter in der Simulation war stabil und im Vergleich zu den Feldtestergebnissen sehr gering. Die Emulationsergebnisse bezüglich der untersuchten KPIs liegen nahe an den realen Feldtestergebnissen. Dies zeigt den Nutzen einer emulierten Testumgebung für kontrollierbare Experimente.

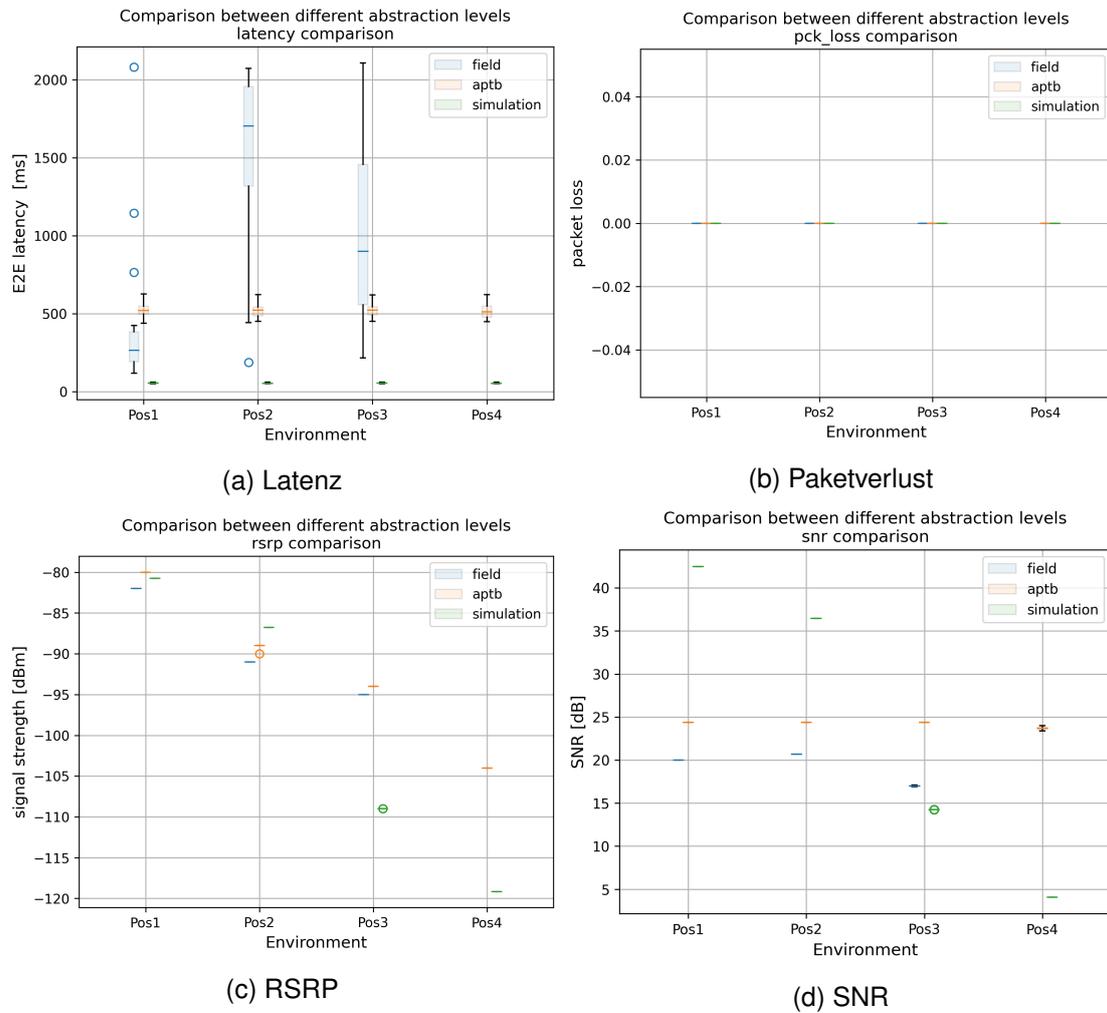


Abbildung 40: Vergleich von RSRP und SNR in Feld- und Emulationsumgebung

Zur Reproduktion von Feldtestergebnissen im APTB wurde der RSRP so eingestellt wie der Median-Wert in den Feldtests. In Abb. 41 ist sowohl SNR und RSRP in der Feldtest- und der Emulationsumgebung verglichen. Es wird deutlich, dass eine Einstellung des RSRP anhand der vorhandenen Dämpfungs-Komponenten gut funktioniert. Es wird allerdings auch deutlich, dass die Varianz praktisch komplett reduziert wird und der SNR

gegenüber den Feldtestbedingungen leicht angehoben ist. Dies lässt darauf schließen, dass die Bedingungen nicht vollständig identisch sind, beispielsweise durch fehlende Reflektionen, Mehrwegekomponenten oder Interferenzquellen. Dieses Verhalten kann für alle Technologien beobachtet werden. Nur bei LoRa bleibt eine ähnliche Varianz im SNR erhalten. Dies lässt auf Probleme innerhalb der Hardware-Komponenten (Endgerät oder Gateway) schließen.

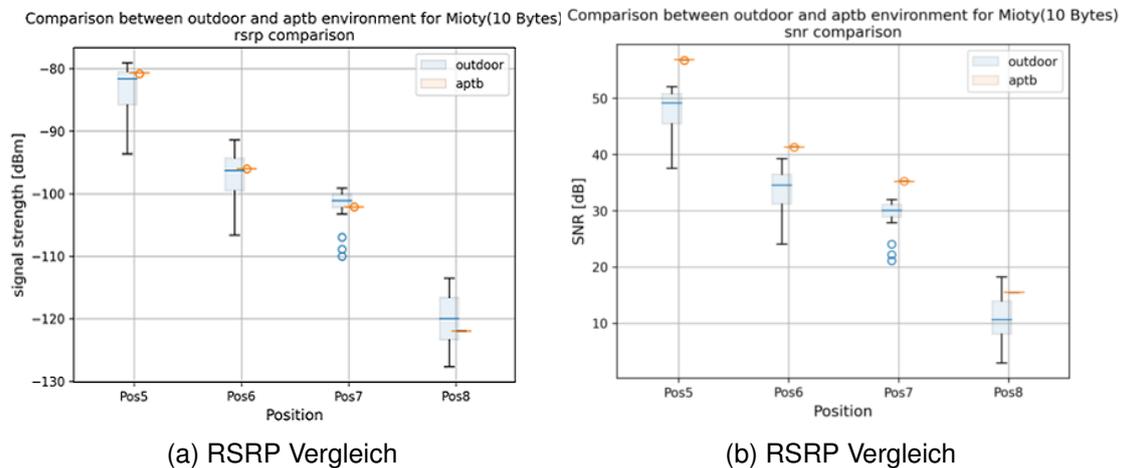


Abbildung 41: Vergleich von RSRP und SNR in Feld- und Emulationsumgebung

10.3. Parametervariationen

In diesem Kapitel werden die Ergebnisse der Parametervariationen, wie in Abschnitt 9.4.3 erläutert, vorgestellt.

Es werden hierbei für jede Technologie zunächst die variierten Parameter in den entsprechenden Unterpunkten dargestellt. Liegen Ergebnisse auf unterschiedlichen Abstraktionsebenen vor, so werden diese gemeinsam dargestellt. Danach folgt eine Betrachtung der Lebensdauer anhand der zuvor ermittelten Daten und schließlich ein Bezug zu den Anwendungsfällen.

10.3.1. NB-IoT

Für die Variation der Parameter bei NB-IoT wurde der CMW500 als Basisstation und das Commsolid GR581 als UE eingesetzt. Nachfolgend sind die Ergebnisse zu den zu variierenden Parametern aus Abschnitt 9.4.3 dargestellt.

Paketgröße Die Payloadgröße ist ein wichtiger Parameter für eine Anwendung. NB-IoT Modems unterstützen dabei 1024–1358 B für ein einzelnes UDP-Paket. Hier soll der Einfluss der Paketgröße auf die Latenz und den Energiebedarf untersucht werden und ermittelt werden, mit welchen Parametern möglichst effizient gesendet werden kann.

In Abb. 42 sind die Umgebungsbedingungen anhand des SNRs sowie die Latenz im Uplink in Abhängigkeit der Paketgröße und der Zelleneinstellung dargestellt. Es ist ein deutlicher Einfluss der Paketgröße auf die Latenz erkennbar. Die Latenz steigt aber unterproportional mit steigender Paketgröße, sodass es effizienter ist größere Pakete zu senden. Weiterhin häufen sich Ausreißer nahe der Zellengrenze, wodurch die Varianz der Latenz ansteigt. Die Minimallatenz ändert sich aber praktisch nicht, und die meisten Pakete kommen mit einer ähnlichen Latenz wie unter normalen Bedingungen an. Dennoch konnte auch in der Nähe der Zellengrenze kein Paketverlust festgestellt werden. Eine weitere Verschlechterung führte zu einem kompletten Verbindungsabbruch, der einen Test unmöglich machte.

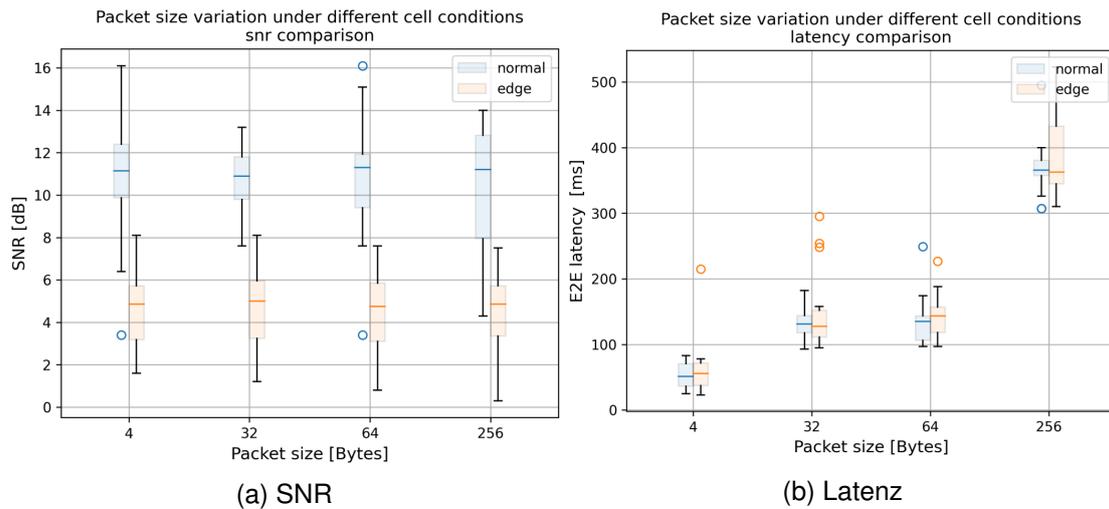


Abbildung 42: NB-IoT: Vergleich unterschiedlicher Paketgrößen

In Tab. 28 sind die Ergebnisse für die Emulationen tabellarisch für Latenz und Paketverlust dargestellt.

Tabelle 28: Zusammenfassung der Variation der Paketgröße für Latenz und Paketverlust

Bedingungen	#Bytes	E2E-Latenz [ms]				Verlust
		mean	std	min	max	
normal	4	53	18	25	83	nein
grenze	4	60	42	23	215	nein
normal	32	132	23	93	182	nein
grenze	32	149	56	95	295	nein
normal	64	133	35	97	249	nein
grenze	64	144	32	97	227	nein
normal	256	367	40	307	495	nein
grenze	256	659	1155	310	5415	nein

Zur Überprüfung des Verhaltens wurden ebenfalls Simulationen durchgeführt. In Abb. 43 ist die Latenz über die Paketgröße für den Uplink und den Downlink innerhalb der Simulation dargestellt. Wie auch in der Emulation erhöht sich die Latenz in Uplink und Downlink mit steigender Paketgröße. Die Latenz im Downlink ist dabei leicht angehoben gegenüber dem Uplink.

Es kann eine Häufung von Ausreißern nahe der Zellengrenze unabhängig von der Kommunikationsrichtung beobachtet werden. Es kommt zu Paketverlust im Uplink und im Downlink. Während im Uplink praktisch kein Paketverlust beobachtet werden kann, treten im Downlink vermehrt Paketverluste auf.

Die in der Simulation erzeugten Paketverluste treten vor allem aufgrund einer Exponentialverteilung beim Senden der Pakete auf. Werden zu kleine Intervalle gewählt oder ist die Paketgröße je Intervall zu groß, so ergeben sich Paketverluste. Es muss daher je Paketgröße mit einer gewissen Totzeit gerechnet werden.

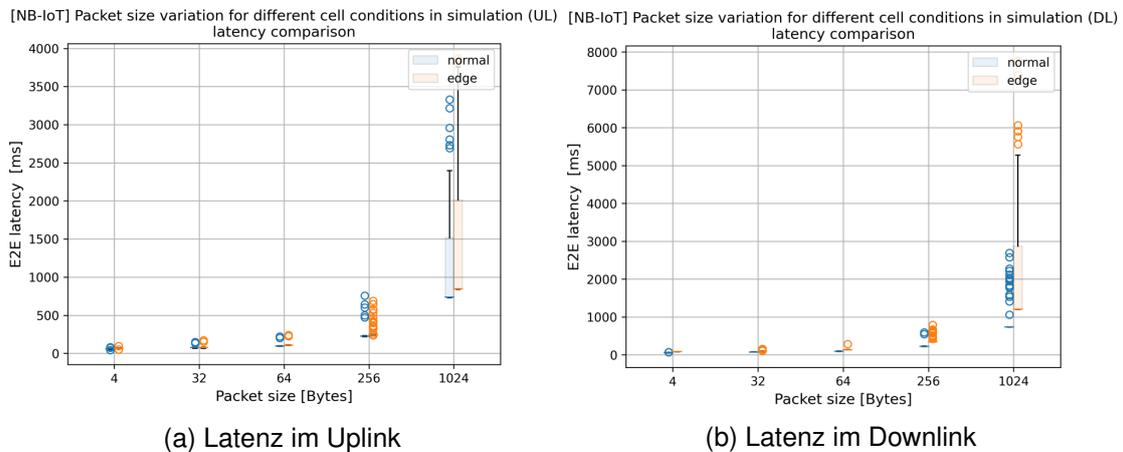


Abbildung 43: NB-IoT: Vergleich unterschiedlicher Paketgrößen in der Simulation

Es wird deutlich, dass der Zusammenhang zwischen minimalem Sendeintervall und Paketgröße ein entscheidender Faktor und zudem für viele Anwendungen von hoher Bedeutung ist. Innerhalb der Simulation konnte dieser Zusammenhang überprüft werden, indem ein fixes Sendeintervall eingestellt wird. Es wurde deutlich, dass das minimale Sendeintervall bei NB-IoT in etwa der Latenz entspricht, wobei im Downlink etwas höhere Intervalle notwendig waren. Wird dieses Intervall verkürzt, setzt Paketverlust ein. In Downlink-Richtung muss dabei die höhere Latenz beachtet werden.

Die Simulationsuntersuchungen zeigen, dass dieses Verhalten sehr sensitiv auf das Intervall reagiert, d.h. schon einige Millisekunden können zu Problemen führen. Weiterhin muss beachtet werden, dass nahe an der Zellengrenze höhere Latenzen auftreten und das Intervall entsprechend angepasst werden müsste. Bewegt sich eine Anwendung nahe dem minimalen Intervall, so müssen diese Faktoren mit einbezogen werden.

Es ist in jedem Fall ratsam einen Sicherheitsfaktor mit einzukalkulieren. Umgekehrt kann man argumentieren, dass die Reichweite auch vom Tupel Intervall-/Paketgröße abhängt, denn mit geringerem Solldurchsatz steigt der Paketverlust erst später an. Modelliert man

ein eventbasiertes Kommunikationsschema über eine Exponentialverteilung, so kommt es vermehrt zu Paketverlust aufgrund von teilweise sehr geringen Sendeintervallen, die jedoch über eine Totzeit eliminiert werden können. Senden Applikationen aperiodisch Daten und können die auslösenden Ereignisse in einem geringen Intervall auftreten, müssen Paketverluste erwartet werden. Es ist daher ratsam, abhängig vom gesendeten Payload eine definierte Zeit zu warten. Ist dies für eine Applikation kritisch, sollte ein anderes Übertragungsprotokoll in Betracht gezogen werden.

Zur Überprüfung des Energiebedarfs wurden die Strom-Zeit Verläufe für unterschiedliche Paketgrößen ermittelt. Während sich die Gesamtenergie bei größeren Paketen erhöht, ist die Energie je gesendetem Byte bei größeren Paketen geringer, wie in Abb. 44 dargestellt. Dieser Zusammenhang zeigt, dass es effizienter ist, größere Pakete zu senden anstatt mehrere kleinere Pakete.

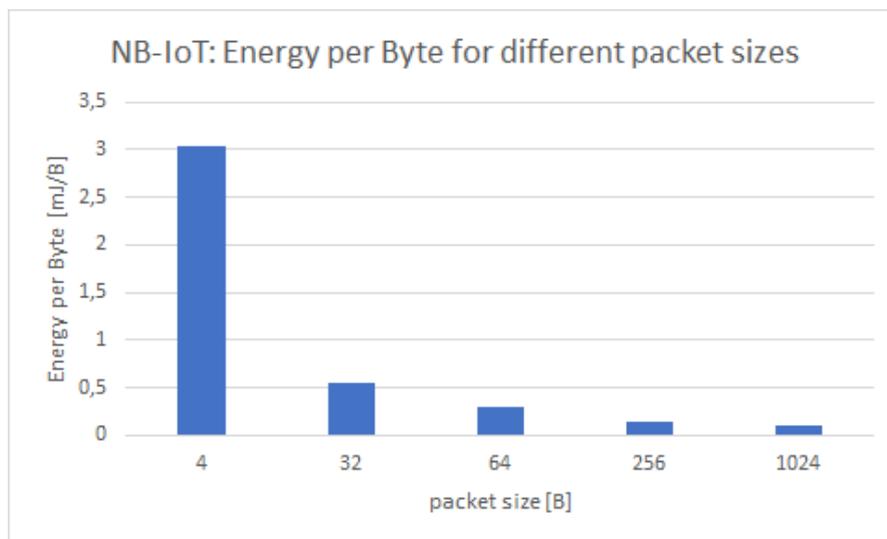


Abbildung 44: NB-IoT: Energiebedarf je Byte für unterschiedliche Paketgrößen

Leistung In NB-IoT ist der P_{max} Netzwerkparameter definiert. Er wird in der eNB gesetzt und limitiert die maximale Leistung eines UEs. Es gibt jedoch keine Möglichkeit die Sendeleistung statisch einzustellen, da dies nach wie vor dynamisch anhand der Kanalbedingungen und einem Schätzmodell für den Pfadverlust bestimmt wird.

Aus unseren Tests folgt, dass dieser Parameter nur mit Vorsicht eingesetzt werden sollte. Der Strombedarf kann dadurch nur unter bestimmten Umständen reduziert werden. Es besteht das Risiko, dass das UE bei schlechter werdenden Bedingungen mit zu wenig Leistung sendet und dadurch beispielsweise mehr Versuche benötigt, was wiederum die Energieaufnahme erhöht. Es wurde deutlich, dass gute initiale Leistungseinstellungen wichtiger sind. Hier sind die zwei Parameter `PO-Nominal-PUSCH` und `initialTargetReceivePower` zu erwähnen. Diese Parameter tragen maßgeblich zur Leistungsanpassung im Uplink und im PRACH Kanal bei. Die optimalen Parameter für

diese Werte können jedoch stark mit der verwendeten Hardware variieren und sollten entsprechend angepasst werden.

Unter guten Bedingungen konnte durch Reduzierung des P_{MAX} der Sendestrom von 124 mA auf 74 mA reduziert werden. Dies war allerdings nur möglich, da die erwartete eNB Leistung auf -80 dBm eingestellt war und entsprechend noch genug Rand für die Erkennung der Signale vorhanden war. Unter schlechteren Bedingungen kam es allerdings häufiger zu Verbindungsproblemen.

Abschließend kann gesagt werden, dass ein gutes initiales Power Setup sehr wichtig ist. Für Energieoptimierung sollte P_{MAX} nicht verwendet werden, da er nur einen Effekt hat, wenn das Modem mit mehr Energie senden möchte. Wenn ein Modem nicht die notwendige Leistung bereitstellen kann, kann es zu zusätzlichen Sendewiederholungen etc. kommen, die die Qualität der Anwendung und auch den Energiebedarf beeinflussen können. Stattdessen sollte die nominale Leistung sowie die initiale PRACH Leistung in der eNB entsprechend eingestellt werden.

Da die Sendeleistung direkt mit dem Pfadverlust und damit den Umgebungsbedingungen zusammenhängt, sollte die Sendeleistung außerdem aufgezeichnet werden, um mögliche optimale Sendezeitpunkte zu erfassen.

HARQ HARQ beschreibt einen Mechanismus, der die Zuverlässigkeit erhöhen soll, indem Paketbestätigungen genutzt werden, wenn ein Paket unverfälscht ankommt. Hierbei wird ein *Stop and Wait* Verfahren genutzt, bei dem ein Sender erst auf die Bestätigung warten muss. Seit 3GPP Release 14 werden bis zu zwei HARQ-Prozesse unterstützt. In diesen Tests wird der Einfluss dieses Features untersucht. Es wurde hierbei mit 32 B Daten in einem Intervall von 2 s gesendet.

In Abb. 45 ist der SNR und die Latenz für unterschiedliche HARQ Einstellungen dargestellt. Es werden positive Einflüsse sowohl auf den SNR als auch auf die Latenz sichtbar.

Die Nutzung von zwei HARQ Prozessen hat einen signifikanten Einfluss auf verschiedene KPIs wie Latenz, Durchsatz und Energiebedarf. Die Nutzung von zwei HARQ-Prozessen ermöglicht ein Parallelisieren des HARQ-Ablaufs. Dadurch kann Zeit bei der Datenübertragung gespart und damit Energie reduziert werden. Im Test mit 32 B und 2 s Sendeintervall im Uplink unter guten Bedingungen kann die mittlere Latenz um über 20% reduziert werden. Zudem sinkt die Varianz und der SNR ist deutlich stabiler. In unseren Tests mit 1024 Bytes und minimalem Sendeintervall konnte die Latenz um einen Faktor von 6 reduziert werden. Gleichzeitig kann damit auch die Varianz deutlich reduziert werden.

Die Verwendung von zwei HARQ Prozessen ist insbesondere für Anwendungen, die einen relativ hohen Durchsatz benötigen, interessant.

Eine Erhöhung der Distanz konnte trotz eines höheren gemessenen SNRs nicht bestätigt werden. Innerhalb der Simulation konnte für das HARQ Feature kein Einfluss auf die KPIs erkannt werden.

Betrachtet man den Energiebedarf, wird deutlich, dass eine geringere Sendedauer auch den Energiebedarf reduziert. Betrachtet man nur die Sendezeitpunkte für 32 B Daten, so

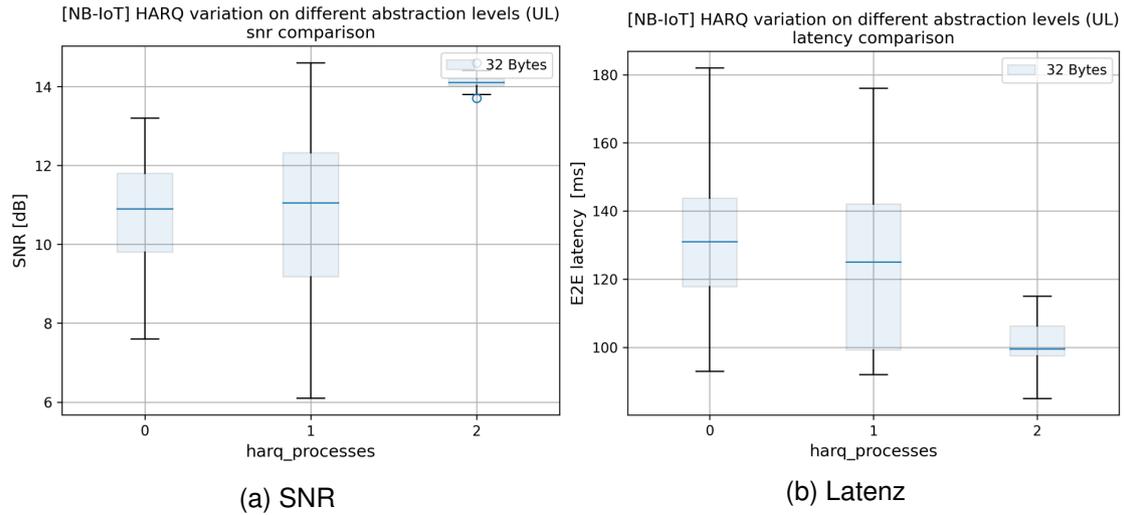


Abbildung 45: NB-IoT: Vergleich unterschiedlicher HARQ-Einstellungen

ergeben sich die Energiebedarfe wie in Tab. 29.

Tabelle 29: NB-IoT: Energie je Tx-Vorgang für unterschiedliche Anzahl von HARQ-Prozessen

HARQ-Prozesse	Ladung [mC]
0	14,2
1	8,0
2	8,0

Der Gesamtenergiebedarf ist jedoch relativ ähnlich aufgrund der hohen Ströme in RRC-Connected, sodass es aus Sicht von Anwendungen mit geringem Sendeintervall sinnvoll ist diese Zeit möglichst zu reduzieren. Weiterhin ist die Nutzung hoher Paketgrößen mit diesem Feature von besonderer Bedeutung. Somit ist das HARQ Feature auch für diese Anwendungen ein interessantes Feature.

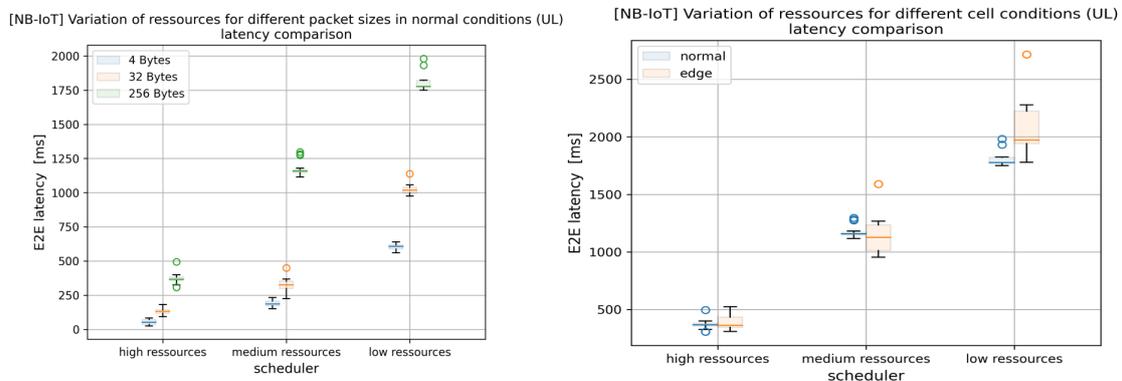
Scheduler Der Scheduler ist dafür zuständig, den Teilnehmern Ressourcen zuzuteilen. Dies soll eine optimierte Performanz für bestimmte Leistungsparameter erzeugen, z.B. effiziente Spektrumnutzung, hohe Netzauslastung usw. Die Algorithmen sind nicht spezifiziert, sodass hier herstellerspezifische Algorithmen zum Einsatz kommen. Für unsere Testgeräte kann ein statisches Scheduling benutzt werden, das es erlaubt, unabhängig von den Umgebungsbedingungen und der Anzahl der Teilnehmer eine Ressourcenanzahl zur Verfügung zu stellen. Es wurden entsprechend drei Szenarien mit einer unterschiedlichen Anzahl an verfügbarer Ressourcen definiert, wobei jeweils mit einem UE gearbeitet wurde. Dabei werden folgende Parameter variiert:

- Anzahl der Ressourcen-Einheiten
- Anzahl der Subframes
- TBS in UL und DL
- Anzahl der Hilfsträger

Die folgenden drei Szenarien wurden definiert:

- Hohe Netzwerklast: Nur wenige Ressourcen stehen für ein neues UE zur Verfügung
 - subcarrier: 1, RU: 1, SF: 1, UL_TBS: 1, DL_TBS: 1
- Mittlere Netzwerklast: Es sind einige Ressourcen verfügbar für ein neues UE
 - subcarrier: 6, RU: 3, SF: 3, UL_TBS: 3, DL_TBS: 3
- Geringe Netzwerklast: Die maximale Anzahl von Ressourcen stehen für ein neues UE zur Verfügung
 - subcarrier: 12, RU: 6, SF: 6, UL_TBS: 6, DL_TBS: 6

Der Einfluss von eingeschränkter Ressourcenverfügbarkeit auf die Leistungsfähigkeit ist in Abb. 46 dargestellt. Neben einem deutlichen Anstieg der Latenz bei unterschiedlichen Bedingungen wird deutlich, dass die Sensitivität bezüglich der Paketgröße bei weniger Ressourcen zunimmt. Sowohl die Sensitivität bezüglich der Paketgröße als auch der Sendeintervalle ist bei weniger Ressourcen deutlich größer. Der SNR und RSRP ändert sich nur geringfügig. Auch hier konnten keine Bedingungen eingestellt werden, die einzelne Paketverluste gezeigt haben.



(a) Latenz unterschiedlicher Paketgrößen

(b) Latenz für unterschiedliche Bedingungen

Abbildung 46: NB-IoT: Vergleich unterschiedlicher Scheduler Einstellungen

Für die Ermittlung des Energiebedarfs wurde die Anzahl der Hilfsträger zwischen 1 und 12 Hilfsträgern variiert und die TBS in den shared channels variiert.

In Tab. 30 sind die entsprechenden Energiebedarfe je Sendevorgang dargestellt.

Tabelle 30: NB-IoT: Energie je Tx-Vorgang für unterschiedliche Scheduler-Konfigurationen

Konfiguration	\bar{I} [mA]	I_{peak} [mA]	Zeit [ms]	Ladung [mC]
Standard	21,4	127,2	397	8,11
TBS tief	16,1	126,5	1725	27,7
TBS tief, SC tief	23	78	1928	44,5

Eine Verringerung der TBS und damit der Anzahl an Daten, die ein Teilnehmer auf der physischen Schicht je Scheduling-Einheit senden kann, führt auf der einen Seite zu einem geringeren mittleren Sendestrom, auf der anderen Seite wird allerdings die Sendedauer deutlich erhöht, wodurch der Energiebedarf insgesamt höher ist.

Eine Verringerung der Hilfsträger verringert zwar den Betrag einzelner Stromspitzen, deren Dauer nimmt allerdings zu, sodass eine Verringerung der Hilfsträger zu einer Erhöhung der notwendigen Energie führt.

Die Möglichkeit, mehr Teilnehmer zu unterstützen und hohe Reichweiten durch Ressourcenreduzierung zu erzielen, führt zu einer deutlich erhöhten Energieaufnahme, die die Lebensdauer deutlich beeinflussen kann.

Dieser Test zeigt nur exemplarisch die Abhängigkeit der Performanz von der Anzahl im Netz beteiligter Teilnehmer, deren Verkehrseigenschaften sowie des Scheduling-Algorithmus.

Sendewiederholung Sendewiederholungen sollen es ermöglichen, die Reichweite von NB-IoT zu erhöhen. Dabei werden jedes Mal alle eingestellten Pakete gesendet, unabhängig davon ob bereits das erste Paket erkannt wurde. Theoretisch kann man zeigen, dass sich der SNR dadurch verbessert [70]. In den Tests wurde mit einer, acht und 32 Sendewiederholungen für MSG3 und den PUSCH-Kanal gearbeitet. In unseren Tests konnten trotz hoher Sendewiederholungen keine höheren Distanzen bzw. Dämpfungsglieder überwunden werden. Der SNR auf Modemseite blieb ebenfalls praktisch unbeeinflusst von der Einstellung.

Die Latenz für 32 Byte Daten erhöht sich deutlich wie in Abb. 47 dargestellt, unabhängig von den eingestellten Bedingungen.

Der Energiebedarf, wie in Tab. 31 dargestellt, steigt deutlich an. Neben dem mittleren Strom steigt auch die Dauer deutlich an. Es konnte jedoch keine Anpassung zwischen acht und 32 Sendewiederholungen festgestellt werden. Ein steigender Energiebedarf wird bei weniger zur Verfügung stehenden Ressourcen zusätzlich zunehmen. Zwei zentrale Argumente für den Einsatz von NB-IoT, Reichweite durch Sendewiederholungen und hohe Skalierbarkeit (weniger Ressourcen je UE), werden damit deutlich vom Energiebedarf beschränkt.

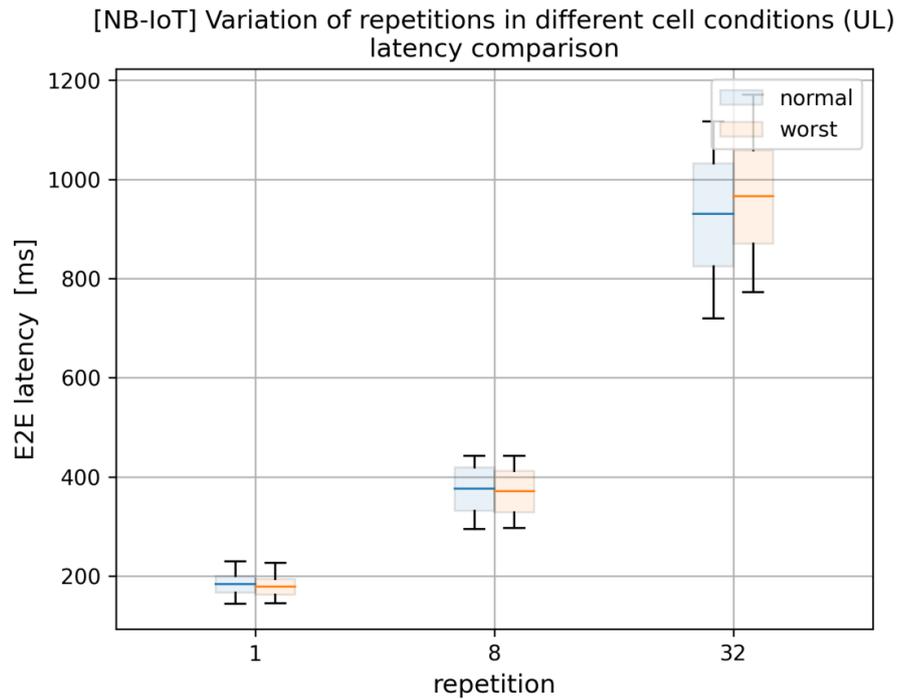


Abbildung 47: NB-IoT: Latenzerhöhung durch Sendewiederholungen

Tabelle 31: NB-IoT: Energie je Tx-Vorgang für unterschiedliche Anzahl an Sendewiederholungen

Wiederholungen	\bar{I} [mA]	I_{peak} [mA]	Zeit [ms]	Ladung [mC]
1	19,8	161	408	8,1
8	132	335	290	38,2
32	135	331	284	38,3

Zur Ermittlung der maximalen Reichweite wurde die Dämpfung im APTB erhöht, bis Paketverlust gemessen werden konnte. Die Anzahl der Sendungen wurde zwischen eins und 32 eingestellt. Für die unterschiedlichen Anzahlen an Sendewiederholungen ergeben sich die in Tab. 32 dargestellten Werte. In diesem Test konnten bei einer einzigen Sendung 3 dB weniger Dämpfung überwunden werden als in den anderen beiden Konfigurationen. Diese unterscheiden sich nicht in der Reichweite. Es muss beachtet werden, dass minimal nur 1 dB Dämpfung eingestellt werden kann und sich bei 32 Wiederholungen ein leicht höherer SNR ergibt, sodass eine leicht erhöhte Reichweite bei 32 Wiederholungen gegenüber acht Wiederholungen zu erwarten ist.

Tabelle 32: NB-IoT: Sensitivitätsgrenze in Abhängigkeit der Sendewiederholungen

Wiederholungen	Min. SNR [dB]	Dämpfung [dB]
1	-1,8	147,9
8	-4,6	150,9
32	-3,9	150,9

Im Bezug auf die Anwendungsfälle sind für PA1, MM1 und MM2 hohe Reichweiten gefordert. Diese Reichweiten sind mit den eingestellten Konfigurationen weder in Feldtests noch in der Simulation möglich. Es ist ratsam, die Hardware so auszulegen, dass die Distanzen auch ohne Sendewiederholungen erreicht werden können. Aufgrund kaum erzielter Vorteile in diesem Test, aber unter Berücksichtigung, dass Sendewiederholungen in anderen Tests einen Einfluss zeigen [70], sollten Sendewiederholungen nur mit Vorsicht und unter extremen Bedingungen eingesetzt werden. Für die im Projekt betrachteten Anwendungsfälle sollten sie nur für PA2 und MM2 in Betracht gezogen werden.

RAI Das release assistance indication Feature ermöglicht es den Energiebedarf zu reduzieren, indem die RRC-Verbindung zwischen UE und eNB frühzeitig gelöst wird. Dadurch wird der relativ hohe Energiebedarf in diesem Zustand vermieden. Zum Testen dieses Features wurden 32 B in einem Intervall von 20 s gesendet. Hintergrund dieses hohen Intervalls ist, dass ein Zustandswechsel in den `Disconnected` Modus erst 10 s nach der Registrierung möglich ist. Die Tests bestätigen eine Reduktion des Energiebedarfs durch die Vermeidung der Inaktiven Zeit im `RRC-CONNECTED` Zustand. Die zusätzlichen 10 s im `Connected` Zustand können praktisch komplett vermieden werden, wodurch sich etwa $12,8\text{mA} \cdot 10\text{s} \cdot 3,3\text{V} = 422\text{mJ}$ an Energie je Sendeprozess einsparen lassen.

Es zeigt sich zusätzlich, dass die Latenz in diesem Modus leicht erhöht ist, und speziell an der Zellengrenze signifikante Ausreißer auftreten können. Die Ausreißer, die in Abb. 48 zu 20 s und 40 s sichtbar werden, treten aufgrund von Puffereffekten auf. Die Pakete werden nicht direkt gesendet sondern erst aggregiert beim übernächsten Paket. Der Hintergrund dieses Verhaltens konnte nicht weiter ermittelt werden. Dies kann aber offensichtlich einen sehr starken Einfluss auf Anwendungen haben.

Das RAI-Paket benötigt jedoch etwas mehr Leistung. Dieser Umstand muss bei der Verwendung beachtet werden. Der entsprechende Energiebedarf für die reine TX-Phase ist in Tab. 33 dargestellt.

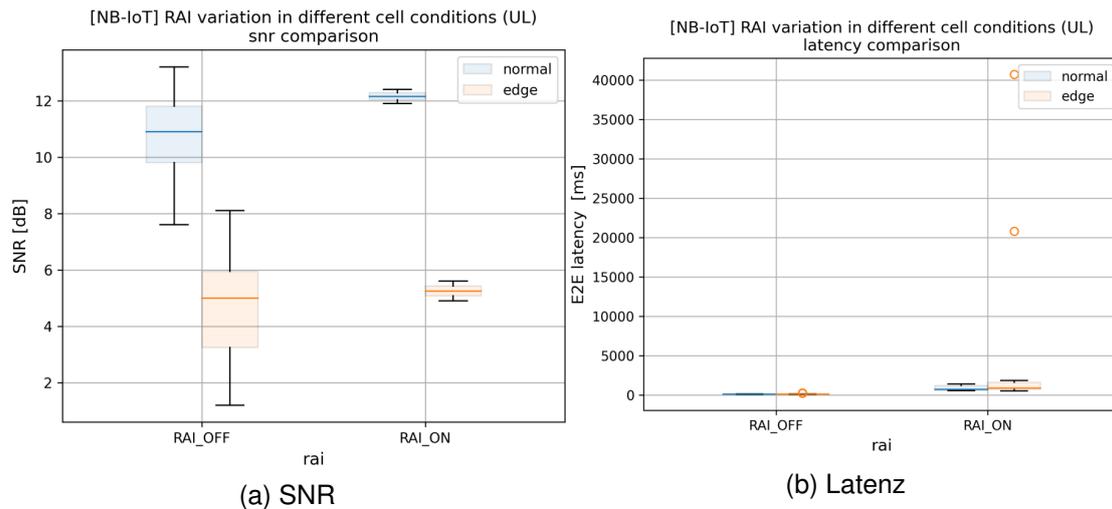


Abbildung 48: NB-IoT: Vergleich unterschiedlicher RAI Konfigurationen

Tabelle 33: NB-IoT: Energie je Tx-Vorgang für RAI Konfigurationen

Zustand	\bar{I} [mA]	I_{peak} [mA]	Zeit [ms]	Ladung [mC]
RAI aus	19,8	161	408	8,1
RAI an	24,1	134,7	645	15,5

Sendezustände Das UE kann verschiedene Zustände einnehmen. Je nachdem, in welchem Zustand sich das UE befindet, können Unterschiede beim Senden von Daten auftreten. In dieser Untersuchung wurden die folgenden drei wichtigen Zustände betrachtet.

- RRC Connected
- RRC Disconnected
- PSM

In Abb. 49 ist die Latenz dargestellt, die benötigt wird, um ein 32 Byte Paket aus dem entsprechenden Zustand vom UE an einen Server zu senden. In Tab. 34 sind die Energiebedarfe dargestellt. Es zeigen sich klare Unterschiede sowohl in Latenz als auch im Energiebedarf.

Interessanterweise wird im RRC_DISC-Zustand die meiste Energie sowie die höchste Latenz benötigt. Es ist anzumerken, dass dieser Zustand neben dem RRC_CONN-Zustand insbesondere für das Empfangen von Daten wichtig ist. Im RRC_DISC-Zustand selbst wird jedoch weniger Energie benötigt.

Für die Anwendungsfälle PA1 und MM1 muss aufgrund des Sendeintervalls immer im Connected Zustand gesendet werden, während in PA2 und MM2 der PSM Modus genutzt werden kann. Zum Empfangen von Daten sollte eine Anwendung, sofern möglich, die Zeit nach einem Uplink nutzen, sodass möglichst in den energiesparsamen PSM Modus gewechselt werden kann.

[NB-IoT] Sending from different protocol states under different conditions (UL)
latency comparison

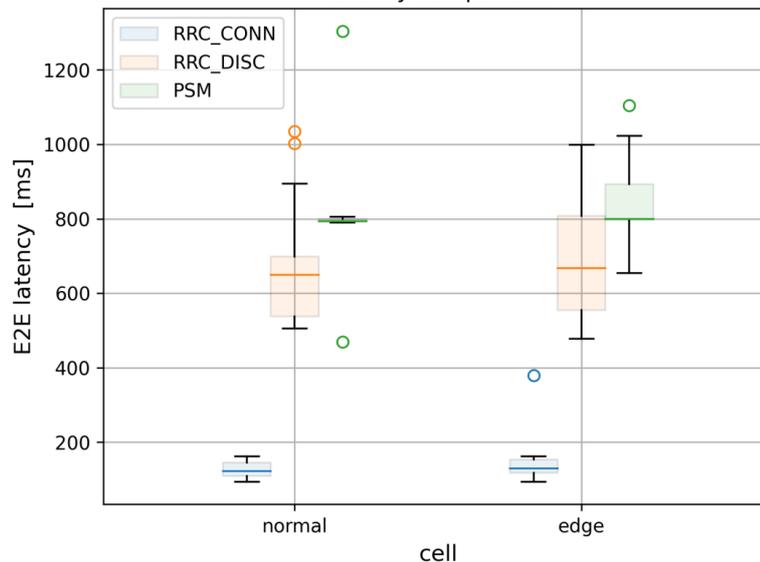


Abbildung 49: NB-IoT: Einfluss des Sendezustands auf die Latenz

Tabelle 34: NB-IoT: Energie je Tx-Vorgang aus unterschiedlichen Zuständen

Zustand	\bar{I} [mA]	I_{peak} [mA]	Zeit [ms]	Ladung [mC]
RRC-Connected	19,8	161	408	8,1
RRC-Disconnected	20,5	130	923	18,9
PSM	20,9	142	593	12,4

Teilnehmer Die Anzahl der Klienten im Netzwerk kann einen signifikanten Einfluss auf die Performanz haben. Die aktuelle Simulation ist auf eine maximale Anzahl von 300 Klienten begrenzt. Sind hier die Solldurchsätze hoch, kann es bereits bei 100 Klienten zu signifikantem Paketverlust und Erhöhung der Latenz kommen.

In Abb. 50 ist der Paketverlust und die Latenz für einen, zehn und 100 Klienten mit unterschiedlicher Paketgröße an der Zellengrenze dargestellt. Hierbei sendet jeder Klient in einem 2 s Intervall. Während sich bei einer Paketgröße von 32 Byte praktisch nichts an der Performanz ändert, sind klare Performanzeinbußen bei einer Paketgröße von 1024 Byte zu erkennen. Für die relevanten Anwendungsfälle sind Sendeintervalle von Stunden

bis Tagen relevant. Mit Intervallen im Minutenbereich konnte bei NB-IoT mit 100 Klienten kein Paketverlust erzeugt werden und keine Erhöhung der Latenz festgestellt werden. Die Anforderung, mehrere tausend Teilnehmer zu unterstützen, konnte aber weder in der Simulation noch in der Emulation getestet werden. Innerhalb der Emulation kann das Verwenden geringer Ressourcen, wie weiter oben dargestellt, nur einen Eindruck zu den zu erwarteten Verhältnissen geben. Grundsätzlich kann gesagt werden, dass Paketverlust einsetzt und sich die Latenz signifikant erhöht, sobald die Anzahl an Daten je Zeiteinheit in einen Bereich kommt, der im Bereich der maximalen TBS liegt.

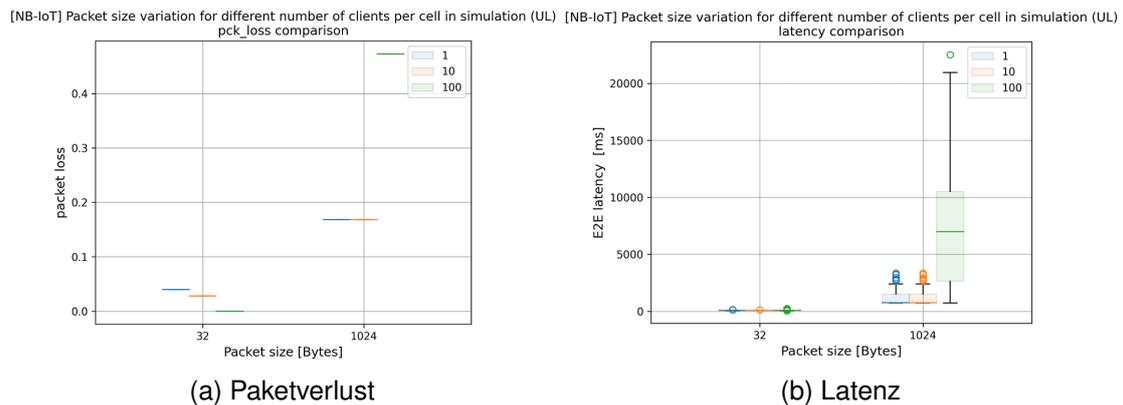


Abbildung 50: NB-IoT: Vergleich unterschiedlicher Teilnehmerzahlen

Weitere Kommunikationsmuster Werden kombinierte Kommunikationsmuster in der Simulation eingesetzt, wie z.B. „Uplink triggert Downlink“, so entsprechen die Ergebnisse denen der einzelnen Richtungen kombiniert. Die Latenz kann als Summe der Latenz in Uplink und Downlink ermittelt werden und der Paketverlust ergibt sich als das Maximum von Uplink und Downlink Richtung.

Für den Testfall eines Bursts können in der Simulation bis zu 10 000 B im Uplink und 5000 B im Downlink gesendet werden. Für das Sendeintervall muss allerdings die erhöhte Latenz von 7750 ms und 3433 ms berücksichtigt werden. Mit den Feldgeräten können Burst nur durch das Senden mehrerer Pakete maximaler Größe unterstützt werden, d.h. ab einer Größe über 1388 B muss ein neuer Sendeprozess getriggert werden.

Tests wurden entsprechend separat durchgeführt und wenn notwendig erst bei der Analyse zusammengeführt.

Für ein Uplink triggert Downlink bzw. Downlink triggert Uplink Szenario wie in den PA1 und MM1 Anwendungsfällen können 500 ms Intervalle noch mit bis zu 128 B in Uplink und Downlink unterstützt werden.

Energiebedarf und Lebensdauer Aus den vorherigen Tests wurden die Ströme und Zeiten unterschiedlicher Phasen aufgezeichnet. Es wurden mittlere Ströme anhand der

Daten von drei Paketen ermittelt, wobei die Strom-Zeitverläufe meist sehr ähnlich waren. Diese Daten werden benutzt, um näherungsweise eine Lebensdauer abzuschätzen. In Tab. 35 sind die Energiedaten für unterschiedliche Operationen dargestellt. Beim Senden mit Sendewiederholungen ist der Energiebedarf deutlich am höchsten, gefolgt vom Senden in RRC_DISC und mit RAI. Den wichtigsten Anteil stellt allerdings das Verbleiben im RRC_CONN Zustand dar, der aufgrund des Netzwerkparameters `inactivity_timer` sehr lange andauert. Ohne Energiesparmodus wird auch im RRC_DISC deutlich zu viel Energie benötigt.

Tabelle 35: NB-IoT: Ermittelte Energiedaten für unterschiedliche Operationen

Operation	\bar{I} [mA]	I_{peak} [mA]	Zeit [ms]	Ladung [mC]
Senden	19,8	161	408	8,1
Senden (HARQ 2)	18,5	154,9	433,9	8,04
Senden (RAI)	24,1	134,7	645	15,5
Senden (Rep. 8)	132	335	290	38,2
Senden von RRC_DISC	20,5	130	923	18,9
Senden von PSM	20,9	142	593	12,4
RRC_CONN	12,7	23,02	9950 (var)	126,38 (var)
RRC_DISC	6,9	7,9	1000 (var)	6,9 (var)
RRC_DISC + Sleep	10,1e-3	11,3e-3	500 (var)	5,0e-3 (var)
Idle-PSM	2,4e-3	2,4e-3	10000 (var)	24e-3 (var)
Idle-PTW (5.12 s)	0,595	49,4	5120 (var)	3,04 (var)
TAU-Update	17,04	191,86	903 (var)	15,39 (var)
Empfangen in RRC_CONN	0	0	0	0
Empfangen in EDRX	15,82	123,5	1169	18,51
Empfangen in TAU	16,5	127,2	1096,8	17,68

Anhand der Tabelle wird deutlich, dass im Idle Modus die meiste Energie benötigt wird. Daher ist die Verwendung des RAI-Features trotz etwas höheren Energiebedarfs beim Senden als beim Senden ohne RAI sinnvoll. Weiterhin kann festgestellt werden, dass Empfangen insgesamt mehr Energie benötigt als zu senden, sofern nicht mehrere Sendewiederholungen genutzt werden. Beim Empfangen sollte beachtet werden, dass ohne RAI im RRC_CONN-Zustand verblieben wird. Es ist daher wichtig, dass RAI für Uplink und Downlink eingestellt wird.

TAU Updates benötigen eine ähnliche Energie wie das reine Senden. Es ist daher effizienter zu senden statt auf das TAU-Update zu warten, um eine doppelte Energiebelastung zu vermeiden.

Benötigt eine Applikation Uplink- und Downlink Möglichkeiten, sollte im Downlink möglichst nach einer Uplink Nachricht gesendet werden. Es kann entsprechend die RAI Nachricht verwendet werden, die nach Empfang einer Downlink-Nachricht die Verbin-

dung freigibt.

Werden nur energiesparsame Modems in einem Netzwerk eingesetzt, ist es sinnvoll, in der eNB die `inactivity_timer` auf null zu setzen, sodass die Verbindung automatisch freigegeben wird.

Ist ein aperiodischer Empfang von Downlink-Daten notwendig, sollte das eDRX-Feature einem kurzen TAU-Intervall vorgezogen werden. Statt des TAU-Intervalls kann praktisch auch einfach ein Paket im Uplink gesendet werden, was diesen Timer entsprechend zurücksetzt.

In Abb. 51a ist die Lebensdauer für unterschiedliche UE-Konfigurationen, Sendeintervalle und Inaktivitätszeiten dargestellt. Dabei wurden die zuvor ermittelten Ströme verwendet, um die Lebensdauern für das Szenario des Sendens im Uplink zu berechnen. Hierbei wird die Energie folgendermaßen abgeschätzt:

$$\bar{E}_{cycle} = U_{bat} \cdot (\bar{I}_{tx} \cdot t_{tx} + \bar{I}_{conn} \cdot t_{conn} + \bar{I}_{psm} \cdot (t_{period} - t_{tx} - t_{conn}))$$

Schließlich wird die Lebensdauer abgeschätzt unter der Annahme einer 5 Wh Batterie und einem Sicherheitsfaktor von $\frac{1}{3}$. Diese Berechnung soll nur eine grobe Orientierung und Einschätzung bieten und stellt den Sachverhalt nicht voll umfänglich dar.

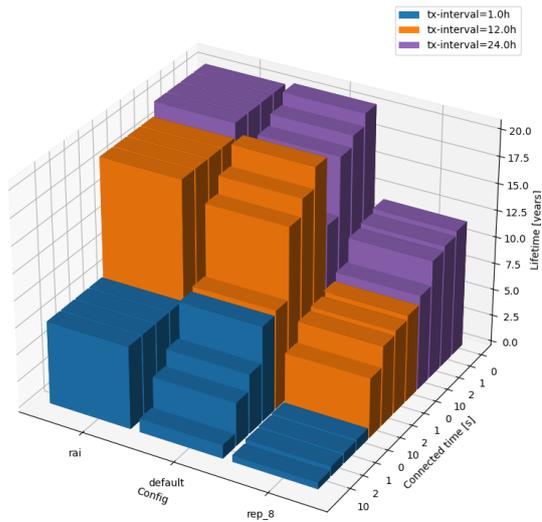
Die Abbildung macht den Einfluss des `inactivity_timer`s sehr klar deutlich. Im Falle der RAI Einstellung wird in jeder Konfiguration dieselbe Leistung benötigt, während sich in den Standardeinstellungen die Lebensdauern, je nach Einstellung, um einige Jahre reduzieren können. Im Falle von acht Sendewiederholungen wirkt sich dieser Effekt weniger aus, da die Energie im Vergleich zur Sendeenergie nicht zu sehr ins Gewicht fällt. Es wird allerdings sehr klar deutlich, wie deutlich der Energiebedarf von der Konfiguration abhängt – acht Wiederholungen sind praktisch nur bei Sendeintervallen von 12–24 h zu empfehlen.

Eine Unterstützung des Downlinks erfordert entweder das Verlassen des PSM oder die Nutzung des eDRX Features. Idealerweise wird direkt nach dem Senden empfangen. Es kann dann entsprechend direkt in den PSM gewechselt werden. In Abb. 51b ist dieser Umstand dargestellt. Während sich die Lebensdauer klar verkürzt, sind dennoch ausreichende Lebensdauern zu erwarten.

Bezug zu Anwendungsfällen Im Bezug auf die Anwendungsfälle können die Latenzanforderungen für FA2, PA1, PA2, MM1 und MM2 erreicht werden. Für FA2 ist jedoch das Sendeintervall zu gering – dies kann von der Anwendung nicht unterstützt werden. Da unter allen Umgebungsbedingungen keine Paketverluste auftraten, ist auch die Zuverlässigkeitsanforderung erfüllt. Eine Periodizität von 500 ms, wie in PA1 und MM1 gefordert, kann nur unter guten Bedingungen eingehalten werden – insbesondere für die maximalen Datengröße von 250 B für Anwendungsfall MM1. Hier sollte wenn möglich die Paketgröße auf 64 B reduziert werden oder die Periodizität auf >1 s eingestellt werden.

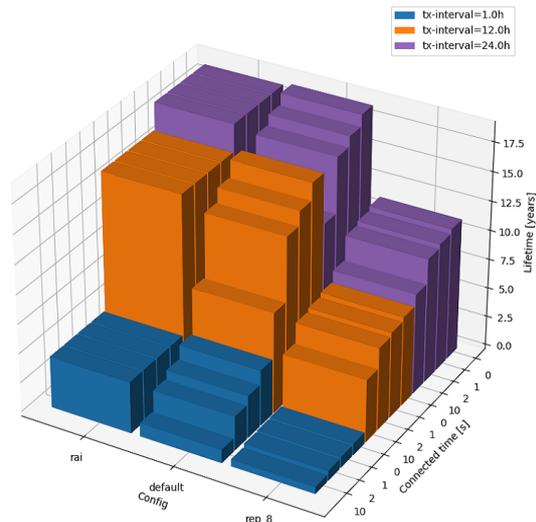
Bezüglich des Energiebedarfs sind solch kleine Sendeintervalle wie in PA1 und MM1 gefordert nur bei kleinen Batterielebensdauern im Bereich von Stunden bis Tagen möglich. Die Forderung von über einem Jahr Batterielaufzeit bei einem 500 ms Sendeintervall

[NB-IoT] Lifetime for periodic uplink transmission with different inactivity timer settings



(a) Uplink with sending mode variation

[NB-IoT] Lifetime for periodic UL-DL transmission with different inactivity timer settings



(b) Uplink + Downlink

Abbildung 51: NB-IoT: Einfluss des Sendeintervalls und des inactivity timers auf die Lebensdauer

sind nicht möglich, da das UE praktisch ständig aktiv mit Senden beschäftigt ist und von keinen Stromsparfunktionen Gebrauch machen kann, sodass der mittlere Strom so deutlich zu hoch ist.

Daran wird deutlich, dass Stromsparfunktionen eingesetzt werden müssen um solch lange Lebensdauern zu ermöglichen.

Die Skalierbarkeitsanforderungen konnten nur in der Simulation mit 100 Klienten überprüft werden. Bei Sendeintervallen im Bereich von Stunden traten hier keine Einbußen bei Latenz oder Paketverlust auf. Kleinere Sendeintervalle sollten allerdings vermieden werden.

Spielt der Energiebedarf eine Rolle, ergeben sich für NB-IoT damit Anwendungsgebiete wie auch für die LPWA-Lösungen. Die Vorteile geringer Latenzen, höherer Datenraten und des unbeschränkten Sendens können in diesem Fall nicht ausgespielt werden. Die Vorteile eines lizenzierten Spektrums konnten in dieser Kampagne nicht hinreichend untersucht werden, sodass dies als interessantes Argument für den Einsatz von NB-IoT verbleibt.

Als praktisches Fazit für mögliche Anwendungen kann mitgenommen werden, dass

- auch unter schlechten Bedingungen eine zuverlässige Verbindung aufrecht erhalten werden kann
- für Anwendungen mit hohem Sendeintervall mithilfe von RAI signifikant Energie eingespart werden kann
- PSM und RAI immer für stromsparende Anwendungen benutzt werden sollte

- immer zwei HARQ Prozesse verwendet werden sollten
- Anwendungen mit Sendeintervallen vom >1 h mit mehreren Jahren Batterielaufzeit unterstützt werden können
- Sendeintervalle unter 1 s nicht für batteriebetriebene Modems geeignet sind
- die Verwendung von Sendewiederholungen und eine hohe Anzahl an Teilnehmern nur mit hohen Sendeintervallen möglich sind
- ein gutes initiales auf die Hardware abgestimmtes Power Setup essentiell für eine gute Performanz ist
- es effizienter ist mit größeren Paketen zu senden
- in Feldtestaufbauten sichergestellt werden sollte, dass ein Rand von etwa 10 dB zum technologischen Limit vorhanden ist

10.3.2. LTE

LTE wird in diesem Projekt für die Anwendungsfälle betrachtet, die einen hohen Durchsatz und eine geringe Latenz benötigen (FA1 und FA2). Daher wird, wenn nicht anders vermerkt, folgende Standardkonfiguration verwendet:

- DL: RB100,64QAM,TBS15
- UL: RB100,16QAM,TBS10

Für die Evaluierung wird der Energiebedarf nicht betrachtet.

Paketgröße Um den Einfluss der Paketgröße auf die Latenz zu überprüfen, wurde im einem fixen Intervall von 1 s mit unterschiedlicher Paketgröße unter unterschiedlichen Bedingungen gesendet. Anhand Abb. 52 wird deutlich, dass die Latenz für die unterschiedlichen Bedingungen relativ stabil bleibt – mit einem Median von 10–11 ms. Paketverluste traten nicht auf.

Innerhalb der Simulation konnten ebenfalls keine Paketverluste aufgezeichnet werden. Es konnten jedoch vereinzelte Ausreißer in der Latenz an der Zellengrenze festgestellt werden. Die mittlere Latenz innerhalb der Simulation beträgt 22 ms im Uplink und 13 ms im Downlink unabhängig von der Paketgröße. Die TBS in LTE ist groß genug, um auch die maximale Paketgröße in einer Scheduling-Zeiteinheit zu senden.

Bei der Ermittlung der minimalen Sendeintervalle konnte auf Simulationsseite ein Minimalintervall von 1 ms festgestellt werden, während es mit den Feldgeräten auf 50 ms begrenzt war.

Die Paketgröße hat praktisch erst einen Einfluss, wenn sich das UE in der Nähe der Zellengrenze befindet oder die Kombination von Intervall und Paketgröße mehr Daten erzeugt, als von LTE gehandhabt werden kann. Dies geschieht praktisch aber erst für sehr viel Daten. In Abb. 53 ist der Zusammenhang zwischen der verfügbaren Paketgröße

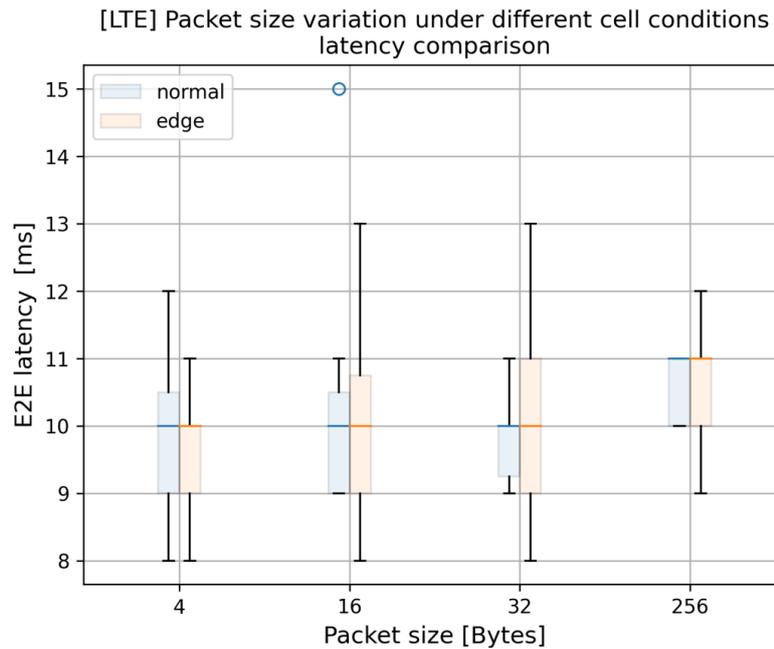


Abbildung 52: LTE: Latenz für unterschiedliche Paketgrößen

je Millisekunde, der eingestellten Bandbreite und dem MCS dargestellt. Als Referenz sind drei horizontale Linien mit unterschiedlichen Paketgrößen dargestellt. Selbst bei minimalen Einstellungen können so 16 bit ms^{-1} gesendet werden. Dies ist mehr als in unserem Test gesendet wird, sodass der Kanal nicht überlastet ist.

Bandbreite Die Bandbreite bestimmt, wie viele Netzwerkressourcen insgesamt verfügbar sind. Entsprechend treten Sättigungseffekte und Paketverluste bei einer geringeren Bandbreite schon früher ein.

Innerhalb der Simulation kann man zeigen, dass bei einem Sendeintervall von 1 ms und 1024 B Daten Paketverlust einsetzt, wenn die Bandbreite auf 1.4 MHz reduziert wird. Der Zusammenhang ist in Tab. 36 dargestellt.

Tabelle 36: LTE: Latenzanstieg und Paketverlust bei Reduktion der Bandbreite

Bandbreite [MHz]	Latenz [ms]	Paketverlust [%]
20	15,7	0
10	15,8	0
5	17,2	0
1,4	44	60

Zur Ermittlung des Durchsatzes in der Emulation wird die Anwendung `iperf3` [71] verwendet. Die Ergebnisse für die unterschiedlichen Bandbreiten sind in Tab. 37 dargestellt.

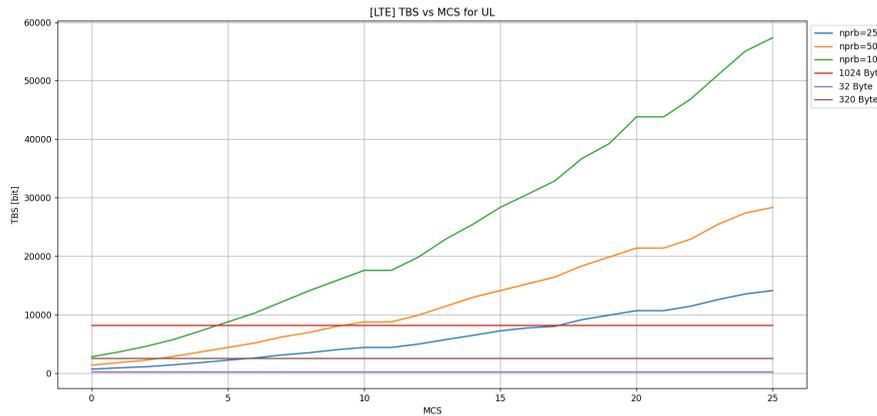


Abbildung 53: LTE: Zusammenhang zwischen TBS, MCS und Bandbreite

Tabelle 37: LTE: Durchsatz für unterschiedliche Bandbreiten im UL

Bandbreite [MHz]	Durchsatz [Mbit/s]
20	17,3
10	8,67
5	3,39
1,4	0,83

Modulation Die Modulation bestimmt die Symbolgröße und damit, wie viel Daten pro Zeiteinheit gesendet werden können. Die Modulation ist Teil der Scheduling Parameter und damit typischerweise nur dynamisch vom Scheduler einstellbar. Auf Simulationsebene kann die Modulation nicht eingestellt werden. Auf der Emulationsebene können jedoch statische Konfigurationen vorgenommen werden. Dabei müssen neben der Modulationsart auch die TBS angegeben werden. Innerhalb einer Modulationsart kann ein bestimmter Bereich von Paketgrößen abgedeckt werden. Die erreichten Durchsätze für die minimale und maximale TBS Konfiguration je verfügbaren Modulationsarten sind in Tab. 38 für den Uplink und in Tab. 39 für den Downlink dargestellt.

Tabelle 38: LTE: Durchsatz für unterschiedliche Modulationsarten im UL

Modulation	ITBS	Durchsatz [Mbit/s]
QPSK	0	2,83
	10	17,3
QAM-16	10	17,3
	22	54

Tabelle 39: LTE: Durchsatz für unterschiedliche Modulationsarten im DL

Modulation	ITBS	Durchsatz [Mbit/s]
QPSK	0	2,73
	9	15,5
QAM-16	15	29,9
	25	62,2

Auf Simulationsebene ist die Einstellung der Modulation nicht möglich. Es kann jedoch untersucht werden welche MCS bei welcher Distanz verwendet wird. Vom MCS lässt sich wiederum auf die Modulationsordnung schließen. Allerdings werden neben der Modulationsordnung weitere Parameter angepasst. In Abb. 54 ist der MCS und die Latenz über die Distanz dargestellt. Daraus wird zum einen ein nichtlineares Verhältnis zwischen Distanz und MCS sichtbar und weiterhin, dass die Latenz erst innerhalb der MCS=0 Einstellung deutlich ansteigt. Dies bedeutet, dass eine Verschlechterung der Latenz erst einsetzt, wenn das System mit seinen vielen dynamischen Parametern überfordert ist. Sobald der Spielraum der MCS-Einstellungen ausgeschöpft sind, kann die Latenz nicht mehr aufrechterhalten werden und es kommt zu erhöhten Latenzen. Innerhalb des Protokollstacks lässt sich ab diesem Punkt beobachten, dass einzelne Queues stetig gefüllt werden, und mit unterschiedlichen Verzögerungen von anderen Schichten ankommen.

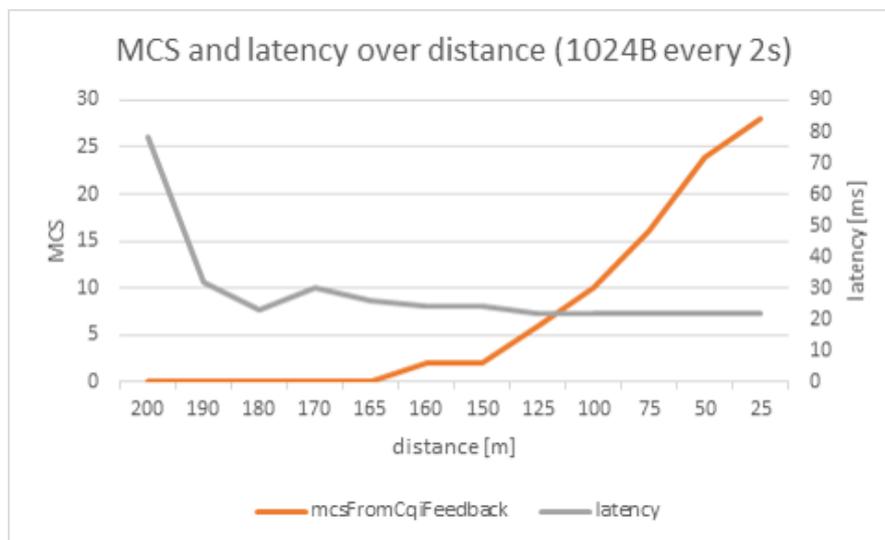


Abbildung 54: LTE: Zusammenhang zwischen MCS und Distanz

HARQ HARQ beschreibt einen Mechanismus, der die Zuverlässigkeit erhöhen soll, indem Paketbestätigungen genutzt werden, wenn ein Paket unverfälscht ankommt. Hierbei

wird ein *Stop and Wait* Verfahren genutzt, bei dem ein Sender erst auf die Bestätigung warten muss. Daher werden mehrere unabhängige HARQ-Prozesse innerhalb von LTE unterstützt [28].

Das HARQ Feature kann in der Simulation an- und ausgeschaltet werden. Im Test wurde mit unterschiedlichen Intervallen und Paketgrößen unter guten und schlechten Zellenbedingungen gesendet. Es konnte bei keiner Variation ein Einfluss auf die KPIs festgestellt werden. Auf der Emulationsebene wurde das HARQ Feature ausgeschaltet und mit vier HARQ-Prozessen verwendet. Es konnte weder unter guten noch unter schlechten Zellenbedingungen ein Einfluss auf die KPIs festgestellt werden. Aufgrund nicht festgestellter Einflüsse sind hier keine Ergebnisse dargestellt.

Teilnehmer Die Anzahl der Teilnehmer reduziert die Anzahl der Ressourcen, die für ein einzelnes UE zur Verfügung stehen. Da in LTE sehr viel Ressourcen vorhanden sind und die Simulation auf eine Anzahl von 300 Klienten beschränkt ist, wurden kleine Sendeintervalle und große Paketgrößen gewählt um diesem Umstand zu testen. Für die Datenmengen und Sendeintervalle der FA-Anwendungsfälle (2–32 B alle 10 ms) konnten keine nennenswerten Einschränkungen bei den bis zu zehn Klienten festgestellt werden. Wird die Paketgröße auf 1024 B erhöht, lässt sich Paketverlust ab einer Bandbreite von 10 MHz und zehn Klienten feststellen, wie in Tab. 40 dargestellt. Wird in dieser Einstellung die Anzahl der Teilnehmer auf 100 Teilnehmer erhöht, schlägt die Simulation jedoch fehl.

Bei Intervallen ab 500 ms können auch 100 Teilnehmer mit maximaler Paketgröße ohne messbare Performanzeinbußen senden.

Tabelle 40: LTE: Latenzanstieg und Paketverlust bei Reduktion der Bandbreite

Teilnehmer	Bandbreite [MHz]	Latenz [ms]	Paketverlust [%]
1	10	23,0	0
	5	23,0	0
10	10	24,4	8
	5	121,8	9

Das Tupel Paketgröße und Sendeintervall stellt den entscheidenden Faktor dar. Für eine Anwendung sollte dies individuell bestimmt werden. Die FA-Anwendungsfälle können im Bezug auf die Skalierbarkeitsanforderungen laut den Simulationsergebnissen unterstützt werden.

Bezug zu Anwendungsfällen Aufgrund der guten Umgebungsbedingungen der FA Anwendungsfälle sind keine Paketverluste zu erwarten. Die Latenzanforderungen von 10 ms in FA1 können nicht erfüllt werden. Die Latenzanforderungen von 100 ms in FA2 können zwar sowohl in Uplink als auch in Downlinkrichtung erfüllt werden, allerdings konnten mit den verwendeten Feldgeräten Probleme bei Sendeintervallen <50 ms festgestellt werden. Innerhalb der Simulation ist dies aber prinzipiell möglich. Prinzipiell

können auch deutlich mehr als 2 B Daten gesendet werden.

Die Unterstützung von bis zu zehn Geräten pro Zelle stellt laut den Simulationsergebnissen kein Problem dar. Die Anzahl der Ressourcen reicht dafür problemlos aus. Interferenzen zwischen den Geräten sind aufgrund der orthogonalen Hilfsträger sowie der Allokation von Nutzern zu unterschiedlichen Zeiten nicht zu erwarten. Die Allokation der Zeitslots für einzelne Teilnehmer und die sichere Einhaltung vorgegebener Echtzeitgrenzen konnten in diesem Projekt nicht untersucht werden. Es ist zentraler Bestandteil der Scheduler Implementierung und Teil von anderen Forschungsgebieten. Für strikte Echtzeitanforderungen ist LTE allerdings nicht spezifiziert worden. Der Nachfolgerstandard 5G bietet erweiterte QoS Mechanismen an, die bei der Umsetzung solcher industrieller Anwendungsgebiete helfen sollen. Aber auch hier sind noch Erweiterungen notwendig, wie beispielsweise die Integration von TSN.

10.3.3. LoRa

Für LoRa werden die Umgebungsbedingungen anhand des SNRs über das APTB eingestellt. Mit der vorhandenen Hardware kann nur im Uplink gesendet werden.

Paketgröße Für diesen Test wurde die Paketgröße zwischen 4–51 B für einen Spreizfaktor von 12 variiert. Die Pakete wurden in einem Intervall von 10 s gesendet. Die maximale Paketgröße für die Spreizfaktoren von 7 bis 9 konnten mit dem vorhandenen Evaluationsstack nicht getestet werden.

Die Latenz und der Paketverlust für die unterschiedlichen Paketgrößen sind in Abb. 55 dargestellt. Es tritt vereinzelt Paketverlust auf und es kommt zu vereinzelt Ausreißern in der Latenz, insbesondere unter schwierigen Umgebungsbedingungen. Die Latenz steigt deutlich mit zunehmender Paketgröße an.

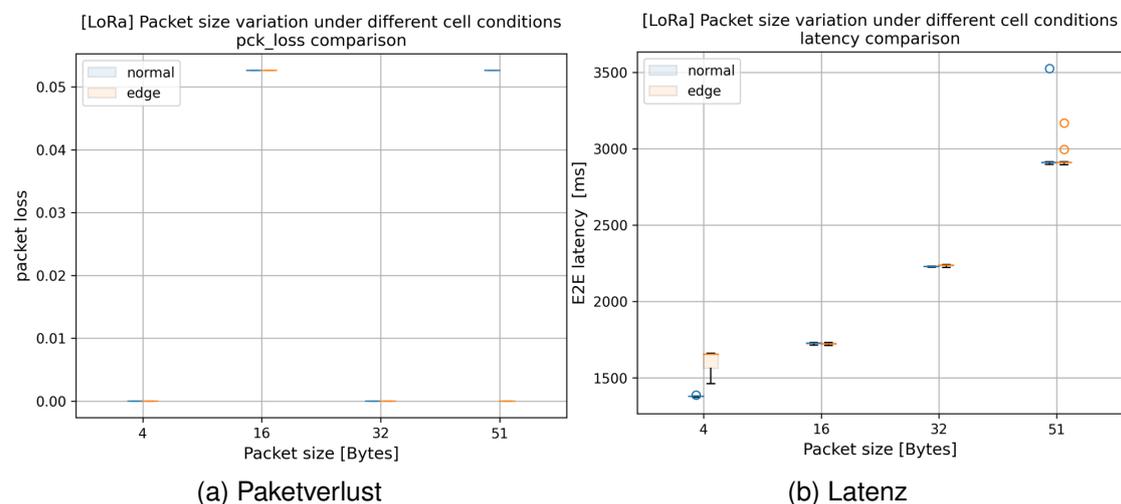


Abbildung 55: LoRa: Variation der Paketgröße

In Abb. 56 ist der Energiebedarf je Byte und Paketgröße dargestellt. Daraus ist ersichtlich, dass es effizienter ist mit größeren Paketen zu senden. Die gewonnene Effizienz nimmt aber mit zunehmender Paketgröße ab.

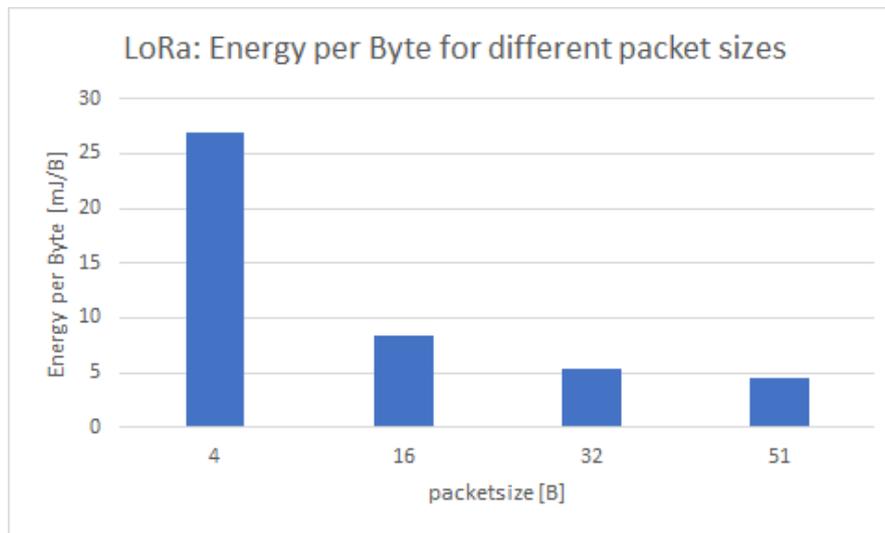


Abbildung 56: LoRa: Energiebedarf je Byte und Paketgröße

In Tab. 41 sind die ermittelten Ströme und Zeiten für das Senden mit unterschiedlichen Paketgrößen für einen SF von 12 dargestellt. Sie dienen als Grundlage für eine Abschätzung der Lebensdauer.

Tabelle 41: LoRa: Energiebedarf für den TX-Zustand unterschiedlicher Paketgrößen

Paketgröße [B]	\bar{I} [mA]	t_{tx} [ms]	Ladung [mC]
4	24,21	1344	32,55
16	23,87	1703	40,67
32	24,03	2198	52,84
51	24,48	2806	68,70

Leistung Im Vergleich zu NB-IoT und LTE lässt sich die Sendeleistung in LoRa fix einstellen. Dies kann bei klarer Kenntnis des Kanals von Vorteil sein. Der Test wurde unter normalen Bedingungen für einen SF von 12 für 32 B Daten durchgeführt. Ein Problem ist, dass eine Reduktion der Sendeleistung an der Zellengrenze zu Verbindungsverlust führt bzw. dass keine Pakete mehr gesendet werden können. Eine Reduktion des RSRPs und des SNRs ist selbstverständlich. Diese fällt allerdings kleiner aus als die Reduktion der Leistung, wie in Abb. 57a deutlich wird. Neben dieser Reduktion, kann zusätzlich eine Erhöhung der Latenz festgestellt werden, wie in Abb. 57b dargestellt.

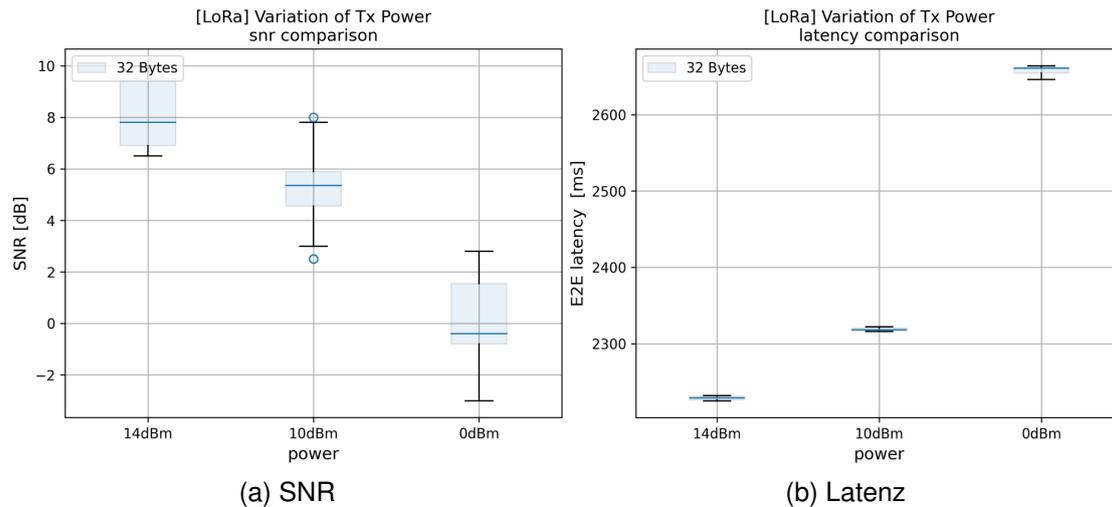


Abbildung 57: LoRa: Variation der Sendeleistung

Der Energiebedarf wird in dieser Einstellung deutlich reduziert. Bei einer Reduktion der Sendeleistung von 14 dBm auf 0 dBm reduziert sich der Energiebedarf je TX auf 46% und bei einer Reduktion auf 10 dBm auf 74%. Aufgrund konstanter Empfangs- und Ruhestrome wurde hier nur die TX-Phase betrachtet. Betrachtet man die etwas geringeren Einbrüche des SNRs, kann es sich lohnen die Leistung zu reduzieren. Dies ist jedoch auch geräteabhängig und kann nicht pauschalisiert werden.

Spreizfaktor Der Spreizfaktor stellt den wichtigsten Einstellparameter von LoRa dar. Für die unterschiedlichen Spreizfaktoren sind unterschiedliche Sensitivitäten möglich. Um die minimale Sensitivität zu bestimmen wurden Daten im Uplink gesendet und mittels des APTBs die Dämpfung erhöht, bis Paketverlust auftrat. Der SNR für die Bedingungen, in denen gerade noch kein Paketverlust auftrat, sind in Tab. 42 dargestellt. In diesem Test kann mit einem $SF = 12$, 12 dB mehr Dämpfung überwunden werden als mit einem $SF = 7$. In diesem Test wurden keine Interferenzen oder Mehrwegeausbreitungen betrachtet, die zu erheblichen Schwankungen des SNRs führen können.

Tabelle 42: LoRa: Sensitivitätslimit in Abhängigkeit des Spreizfaktors

SF	Min. SNR [dB]	Dämpfung [dB]
12	-17,7	146,9
11	-15,5	143,9
10	-15,2	141,9
9	-12,5	140,9
8	-7,9	135,9
7	-6,8	134,9

Zur Ermittlung der weiteren KPIs wurden in einem weiteren Test alle 5 s 32 B Daten bei variierten Spreizfaktoren unter normalen Umgebungsbedingungen im Uplink gesendet. In Abb. 58 sind SNR und Latenz dargestellt. Mit sinkendem SF kann ein leichter Abwärtstrend des SNRs beobachtet werden. Für SF7 ist das technologische Limit bereits erreicht und weniger als der eingestellte SNR kann nicht erreicht werden, wie am einsetzenden Paketverlust festgestellt werden kann. Die Latenz ist bei einen höheren SF größer – zwischen zwei benachbarten Spreizfaktoren unterscheidet sie sich im Mittel um einen Faktor von 1,7.

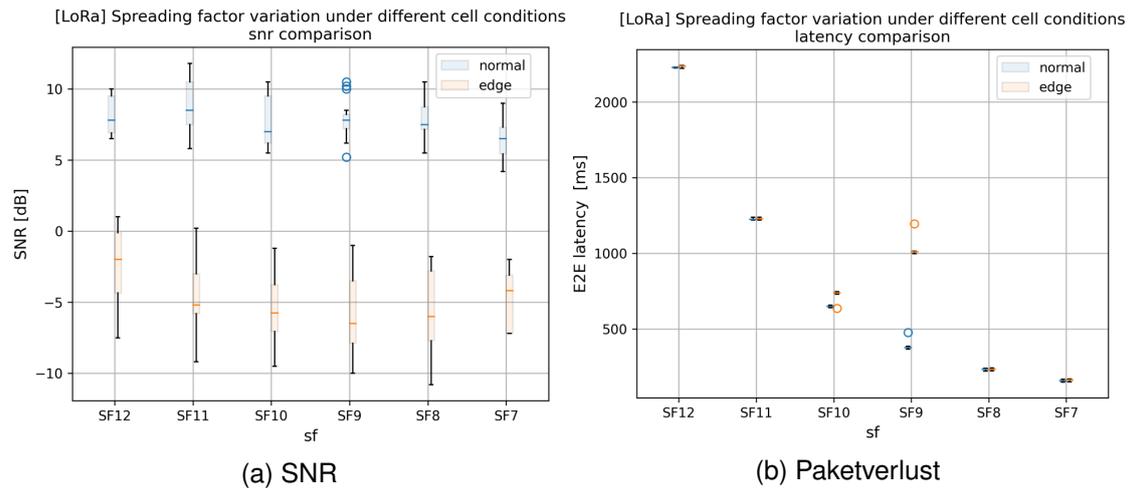


Abbildung 58: LoRa: Variation des Spreizfaktors

Die unterschiedlichen Spreizfaktoren wirken sich sowohl auf die Sendephase als auch auf die Empfangsphase (erstes RX-Fenster) aus – sowohl in der Zeitdauer als auch in der benötigten Energie. Die ermittelten Ströme und Zeiten sind dabei unabhängig von den Umgebungsbedingungen. Der gemessene Ruhestrom von $158 \mu\text{A}$ ist aufgrund des verwendeten Mikrocontrollers größer, als es von einer produktiven Anwendung erwartet werden kann. Dieser Strom ist so hoch, dass selbst ein tägliches Senden nicht ausreicht, um mehr als vier Monate mit einer 5Wh Batterie (Sicherheitsfaktor $\frac{1}{3}$) zu senden. Bei Annahme eines Ruhestroms von $3 \mu\text{A}$ (vgl. NB-IoT) ergeben sich Batterielebensdauern im Bereich von mehreren Jahren. Dies basiert auf den ermittelten Strömen und Zeiten in Tab. 43 für die unterschiedlichen Spreizfaktoren.

Tabelle 43: LoRa: Ströme und aktive Zeiten für unterschiedliche Spreizfaktoren

SF	I_{tx} [mA]	t_{tx} [ms]	I_{rx1} [mA]	t_{rx1} [ms]
12	23,8	2213	6	214
11	22,0	1210	6	145
10	20,2	634	5,9	95
9	19,1	366	5,9	68
8	17,4	222	5,7	61
7	15,4	149	5,6	58

Die entsprechende Energieeinsparung bei der Verwendung von geringeren Spreizfaktoren gegenüber einem SF12 sind in Abb. 59 dargestellt. Die Energie wird dabei zwischen benachbarten Spreizfaktoren um einen Faktor von etwa zwei reduziert.

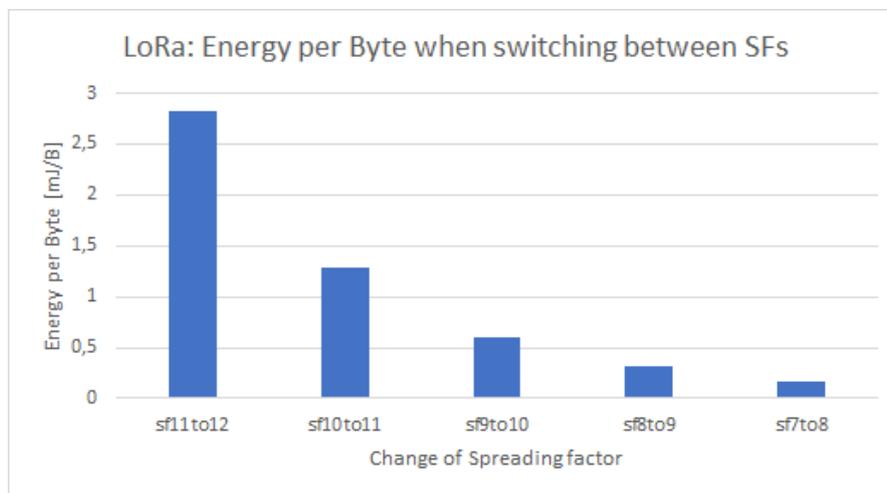


Abbildung 59: LoRa: Energieeinsparung nach Spreizfaktor

Sendewiederholung Sendewiederholungen können als Mittel zur Erhöhung der Reichweite und der Zuverlässigkeit eingesetzt werden. Die Anzahl der Sendewiederholungen wird durch den Parameter `NbTrans` mit einem Wertebereich von 1–9 eingestellt.

In unserem Test wurden 32 B Daten in einem Intervall von 10 s mit unterschiedlichem SF unter schlechten Bedingungen mit einer und drei Sendewiederholungen unbestätigt gesendet.

Bei der Betrachtung der Latenz und des Paketverlustes werden keine Unterschiede deutlich. Die Pakete kommen praktisch immer schon mit dem ersten Paket erfolgreich an, da Paketverlust nicht zuverlässig erzeugt werden konnte.

Der Energiebedarf steigt um den Faktor der Anzahl der Sendewiederholungen an, da dieselbe Prozedur exakt wiederholt wird, d.h. es wird bei jeder Sendewiederholung

gesendet, auf RX1 und auf RX2 gewartet. Mit den uns vorliegenden Daten können Sendewiederholungen daher nicht empfohlen werden. In einer Umgebung mit vielen Teilnehmern, die mit demselben SF senden, können Sendewiederholungen aber ein wichtiges Mittel darstellen um Anforderungen an die Zuverlässigkeit gerecht zu werden. Bei der Verwendung von Sendewiederholungen sollten Sendebestätigungen verwendet werden, da dadurch nicht alle Pakete gesendet werden müssen, sofern die Daten bereits erfolgreich angekommen sind. Dies erhöht aufgrund des geringeren Nachrichtenaustauschs vor allem die Netzwerkskalierbarkeit.

Sendebestätigung Für die Verwendung von Sendebestätigungen konnte kein signifikanter Einfluss festgestellt werden. Weder die Reichweite noch der Energiebedarf zeigten relevante Einflüsse. Sendebestätigungen sind insbesondere bei der Verwendung von Sendewiederholungen wichtig, um unnötige Sendewiederholungen zu vermeiden. Hier muss allerdings ein Konflikt zwischen Skalierbarkeit und Energiebelastung gewählt werden, da die zusätzlichen Downlink-Nachrichten die Kapazität in LoRaWAN-Netzwerken sehr stark beeinflussen kann [18].

Energiebedarf und Lebensdauer Die ermittelten Ströme in den einzelnen Zuständen mit unterschiedlicher Konfiguration dienen als Grundlage für eine Lebensdauerabschätzung. Die benötigte Energie wird folgendermaßen abgeschätzt:

$$\bar{E}_{cycle} = U_{bat} \cdot (\bar{I}_{tx} \cdot t_{tx} + \bar{I}_{wait} \cdot t_{wait} + \bar{I}_{rx1} \cdot t_{rx1} + \bar{I}_{rx2} \cdot t_{rx2} + \bar{I}_{idle} \cdot (t_{period} - t_{tx} - t_{rxwait} - t_{rx1} - t_{rx2}))$$

In Abb. 60 ist die Lebensdauer für das Senden im Uplink dargestellt unter der Annahme eines Ruhestroms von 3 μ A. Sendet eine Anwendung täglich, fällt der unterschiedliche Strombedarf beim Senden mit unterschiedlichen SFs kaum ins Gewicht. Bei stündlichen Sendeintervallen ergeben sich hingegen deutliche Unterschiede in der Lebensdauer, sodass ein Senden mit kurzen Sendeintervallen aus energetischer Sicht nur mit einem $SF < 10$ sinnvoll ist. Aufgrund der geringeren Sendedauer für geringere Spreizfaktoren erhöhen diese deutlich die Lebensdauer.

Die Sendeleistung spielt hingegen besonders bei hohen SFs eine Rolle, da hier nur der Strom, nicht aber die Sendedauer angepasst wird, die bei hohen SFs größer ist.

Die Verwendung eines SF7 reduziert die Sensitivität um etwa 12 dB gegenüber der SF12-Einstellung und hat damit einen ähnlichen Effekt auf die Reichweite wie die Reduzierung der Leistung von 14 dB auf 0 dB. Aus energetischer Sicht reduziert die SF7-Einstellung die Leistung allerdings stärker und mit dem weiteren Vorteil einer geringeren Latenz und damit auch deutlich geringeren möglichen Sendeintervallen. Insgesamt ergibt sich über die Leistungs-Spreizfaktor Variation ein Spielraum von 24 dB bezüglich der Sensitivität und mehrere Jahre Batterielebensdauer.

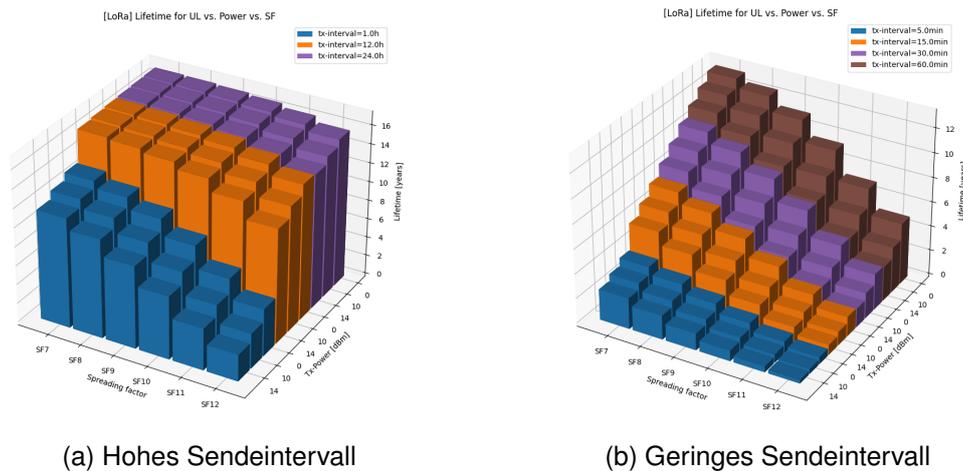


Abbildung 60: LoRa: Lebensdauer vs. Leistung vs. SF

Bezug zu Anwendungsfällen Für die Anwendungsfälle von PA2 und MM2 werden die Latenzanforderungen für alle Spreizfaktoren erreicht, bei PA1 mit SF7 bis SF11 und bei MM1 können sie nur unter Reduktion der Datenmenge vom 250 B auf 150 B erreicht werden. Problematisch sind hier allerdings die regulatorischen Einschränkungen, die ein solch geringes Sendeeintervall (500 ms) nicht erlauben. Selbst bei 4 B Payload und SF7 ergibt sich ein minimales Sendeeintervall von 5 s. Zur Berechnung der möglichen Sendeeintervalle muss ein Duty-Cycle von 1% beachtet werden. Das Sendeeintervall ergibt sich entsprechend als $T_{int} = t_{air} \cdot 100 \approx t_{lat} \cdot 100$. Aufgrund der vergleichsweise einfachen Zusammenhänge sind verschiedene Rechner zur Berechnung der Sendeeintervalle für die einzelnen Spreizfaktoren bei unterschiedlichen Paketgrößen vorhanden [72] [73]. Die hohen Anforderungen an die Batterielebensdauern erfordern insbesondere einen kleinen Ruhestrom. Geringere Spreizfaktoren zeigen Vorteile im Bezug auf Latenz und Energiebedarf, während die Reduzierung der Sendeleistung nur den Energiebedarf in einem kleineren Maße reduziert. Es wird daher empfohlen erst den SF und dann die Sendeleistung anzupassen. In jedem Fall müssen die entsprechenden Sensitivitätsgrenzen anwendungsspezifisch betrachtet werden. In unseren Tests konnten für LoRa insgesamt die höchsten Varianzen innerhalb des SNRs beobachtet werden, daher sollte eine großzügige Abschätzung mit 10–20 dB zum Sensitivitätslimit erfolgen. Entsprechend kann das Tupel (Spreizfaktor, Sendeleistung) gewählt werden. Die Nutzung von Sendewiederholungen muss individuell auf eine Anwendung abgestimmt werden. Insbesondere bei vielen Geräten mit denselben Spreizfaktoren sind Probleme zu erwarten, die mit Sendewiederholungen gelöst werden können. Aus energetischer Sicht ist es beispielsweise immer noch besser, drei Sendewiederholungen in SF7 als eine Wiederholung in SF12 zu senden. Aufgrund fehlender Simulationsmodelle konnten keine Skalierbarkeitstests durchgeführt werden. Andere Untersuchungen zeigen, dass das mögliche Sendeeintervall in LoRa sehr stark von der Anzahl der Teilnehmer und den Duty-Cycle Beschränkungen abhängt, was

für viele Anwendungen zu einschränkend sein kann. Zur Erhöhung der Skalierbarkeit sollten mehrere Gateways eingesetzt werden und die Downlink-ACKs reduziert werden, da dadurch das Gateway praktisch blockiert wird und nicht auf andere Uplink Nachrichten reagiert werden kann.

10.3.4. MIOTY

Für MIOTY werden die Umgebungsbedingungen anhand des SNRs über das APTB eingestellt. Mit der vorhandenen Hardware kann nur im Uplink gesendet werden.

Paketgröße Für diesen Test wird die Paketgröße zwischen 4–51 B im Standard Modus variiert und im Intervall von 10 Sekunden gesendet. In Abb. 61 ist der SNR sowie die Latenz dargestellt. Beide Werte sind unter allen Bedingungen sehr stabil mit nur einer geringen Varianz.

Die Latenz bei Mioty ist deutlich erhöht gegenüber den anderen Technologien. Durch das Telegram Splitting kann das Sendeintervall dennoch relativ gering gewählt werden. Im Vergleich zu LoRa unterscheidet sich die Latenz deutlich von der Zeit, die das Paket in der Luft ist. Betrachtet man den Latenzanstieg je Byte zwischen den ermittelten Paketgrößen, fällt auf, dass der Anstieg zwischen 32 und 51 Byte am geringsten ist je Byte.

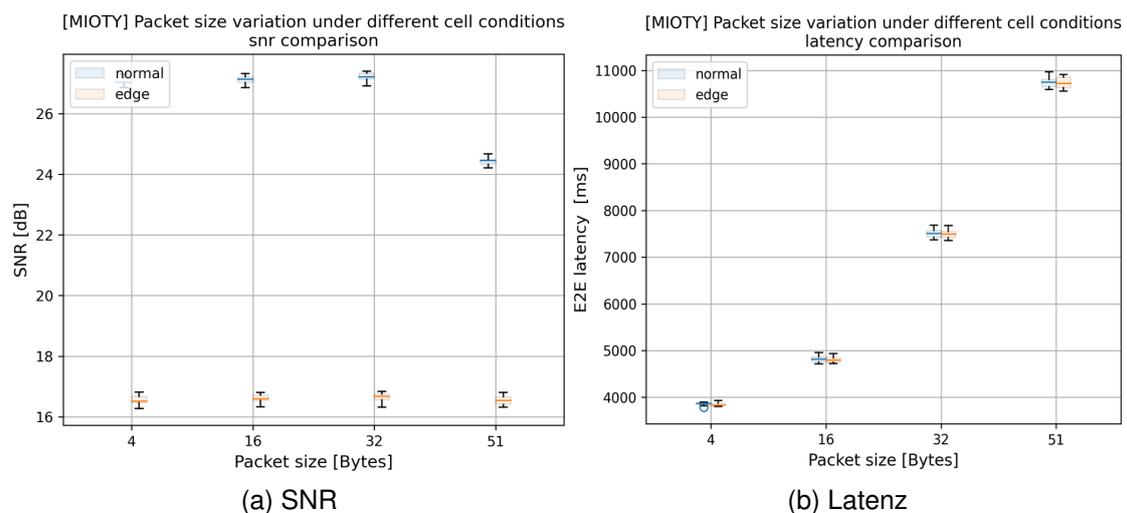


Abbildung 61: MIOTY: Variation der Paketgröße

Betrachtet man den Energiebedarf, so lässt sich ermitteln durch welchen Faktor die Energie beim Senden mit kleineren Paketen reduziert wird. Aus Abb. 62 wird deutlich, dass größere Pakete relativ gesehen weniger Energie je Byte benötigen.

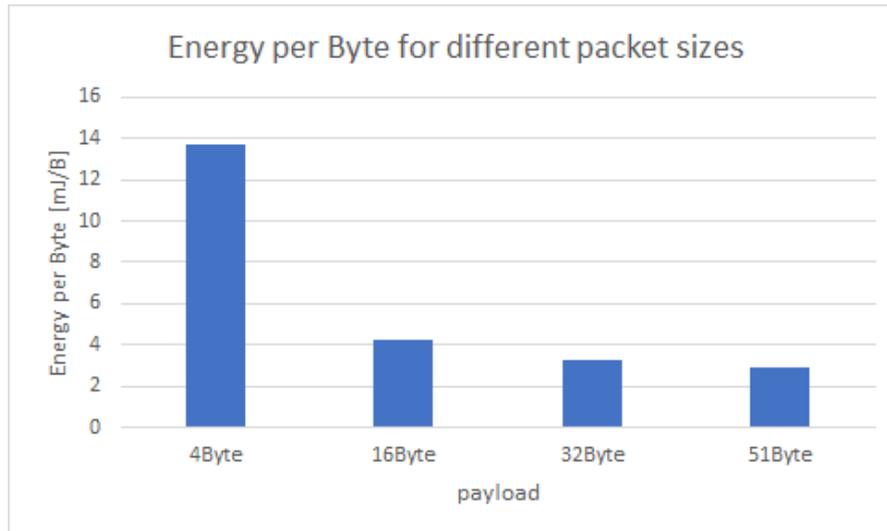


Abbildung 62: MIOTY: Energiebedarf je Byte für unterschiedliche Paketgrößen

Leistung Im Vergleich zu NB-IoT und LTE lässt sich die Sendeleistung fix einstellen. Dies kann bei klarer Kenntnis des Kanals von Vorteil sein. In diesem Fall wird der Test nur unter normalen Bedingungen durchgeführt. Ein Problem ist, dass eine Reduktion der Sendeleistung an der Zellengrenze zu Verbindungsverlust führt, bzw. dass keine Pakete mehr gesendet werden können. Eine Reduktion des RSRPs und des SNRs ist selbstverständlich. Die Latenz bleibt jedoch von dieser Einstellung praktisch unbeeinflusst, sofern sich der SNR noch nicht in der Nähe der Zellengrenze befindet.

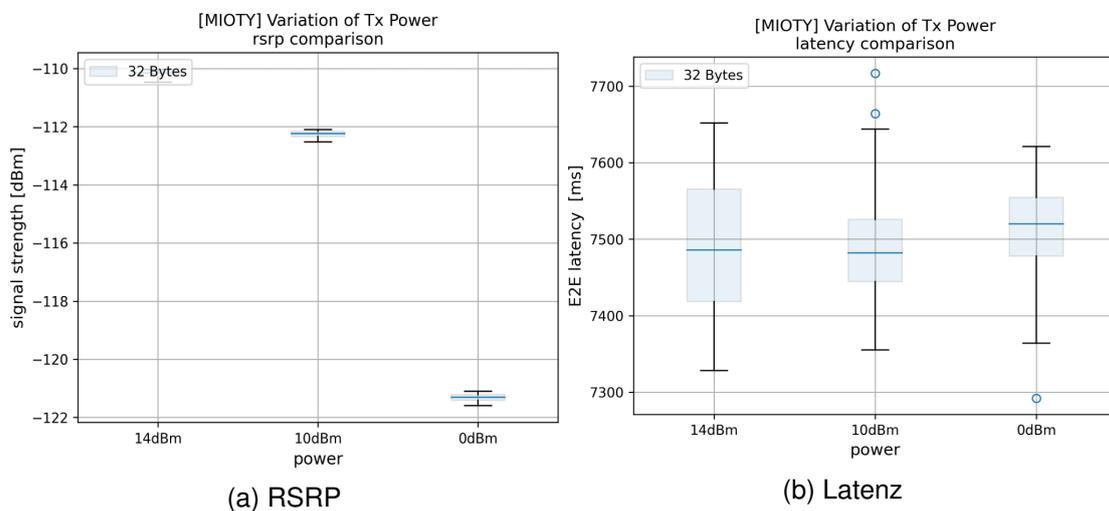


Abbildung 63: MIOTY: Variation der Sendeleistung

Eine Betrachtung der Stromverläufe macht deutlich, dass die Energie deutlich reduziert

werden kann – so weit, dass die Stromspitzen beim Empfangen über die beim Senden herausgehen. Die RX-Ströme sowie die Zeitfenster bleiben unverändert. Es muss beachtet werden, dass ein Mikrocontroller mit einer relativ hohen Stromaufnahme von 1.1 mA im Idle-Zustand verwendet wird.

Da die Zeitfenster und Ströme im Empfangsfenster unabhängig von der Sendeleistung sind, wird zum Vergleich nur der TX-Zustand betrachtet. Somit lässt sich der Energiebedarf bei der Verwendung von 0 dBm auf 60% und bei der Verwendung von 10 dBm auf 90% reduzieren. Aufgrund unveränderter Verhältnisse bezüglich der Empfangsfenster und einem hohen Ruhestrom ist es besser, Pakete zu aggregieren und entsprechend größere Pakete zu senden. Dadurch wird der Empfangsprozess nicht getriggert und der Controller kann beispielsweise komplett abgeschaltet werden. Bei pre-attachten Endpunkten macht außerdem eine Abschaltung zwischen Zeitpunkten Sinn, sofern hohe Ruhestrome vorliegen.

Empfangsfenster In MIOTY ist das Senden von Downlink Nachrichten nur nach einer vorausgehenden Uplink Nachricht möglich. Hierzu wird nach einer Sendepause für eine gewisse Zeit ein Empfangsfenster geöffnet. Aufgrund der beschränkten Hardwaremöglichkeiten war es nicht möglich Nachrichten im Downlink zu senden, sodass in diesem Test nur der Umstand untersucht wird, dass das Fenster geöffnet wird, nicht aber das Empfangen von Daten.

Es konnte kein Einfluss auf die KPIs abseits des Energiebedarfs und des minimalen Sendeintervalls festgestellt werden. Das Sendeintervall verkürzt sich deutlich. Sofern entsprechende Energiesparmodi der Chip-Implementierung unterstützt werden, könnte dadurch bereits früher in einen energiesparenden Modus gewechselt werden, da es weder notwendig ist die Empfangspause von 6.8 s noch das Empfangsfenster abzuwarten. Entsprechend könnte hier der Energiebedarf für diesen Prozess um bis zu 26% reduziert werden. Wird der Strom zwischen der Sendung und dem Öffnen des Empfangsfensters als vernachlässigbar angenommen, wird der Energiebedarf noch um 10% reduziert. Für eine energiesparsame Lösung ist es daher ratsam, das Empfangsfenster nur wenn notwendig explizit zu öffnen.

Modi MIOTY bietet drei Sendemodi an, über die eines der folgenden Profile eingestellt werden kann:

- Latenzoptimiert
- Reichweitenoptimiert
- Kompromiss aus Latenz und Reichweitenoptimierung (Standard)

Zum Test der Modi werden 32 B Daten im Uplink unter normalen Bedingungen gesendet. Hintergrund ist ein deutlich reduzierter SNR im latenzoptimierten Modus, der einen Test an der Zellengrenze nicht ermöglichte. Die verwendete Basisstation war nicht in der Lage die Pakete zu dekodieren, sodass die Latenz und SNR Messungen mit einer

anderen Basisstation (MIOTY M200), die erst kurz vor Projektabschluss zur Verfügung stand, durchgeführt wurde.

In Abb. 64 sind Latenz und SNR für die unterschiedlichen Modi vergleichend dargestellt. Es sind sehr deutliche Unterschiede insbesondere beim latenzoptimierten Modus zu erkennen. Im reichweitenoptimierten Modus sind praktisch keine Unterschiede in Latenz und SNR zu erkennen. Dies kann dadurch erklärt werden, dass MIOTY in diesem Modus die Zeiten zwischen einzelnen Packet-Bursts nahezu halbiert, wodurch die doppelte Anzahl an Paketen in derselben Zeit gesendet werden kann. Der SNR wird nicht beeinflusst, aber unter schlechten Bedingungen ist zu erwarten, dass dadurch die Reichweite erhöht werden kann. Die Latenz im latenzoptimierten Modus wird in diesem Fall um eine Größenordnung von 3,5 reduziert, bei einer Reduktion des SNRs um etwa 2 dB.

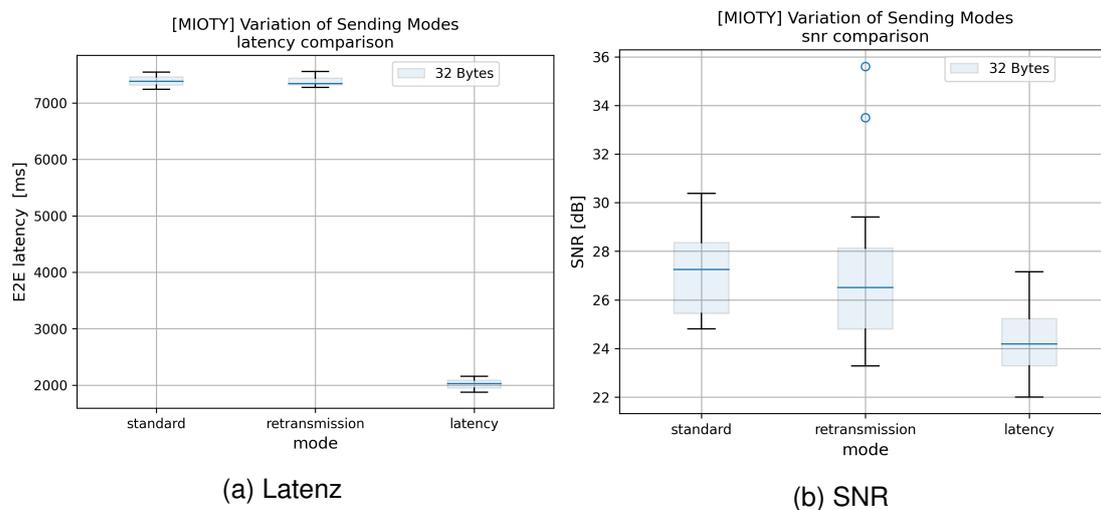


Abbildung 64: MIOTY: Vergleich der Sendemodi bei 32 B Uplink Daten

Zur Ermittlung der maximalen Reichweite wurde die Dämpfung im APTB erhöht, bis Paketverlust gemessen werden konnte. Für die unterschiedlichen Modi ergeben sich Werte wie in Tab. 44 dargestellt. In diesem Test konnten im Latenzmodus 2 dB weniger Dämpfung überwunden werden als in den anderen beiden Modi. Diese unterscheiden sich nicht in der Reichweite.

Tabelle 44: MIOTY: Sensitivitätslimit in Abhängigkeit der Sendemodi

Modus	Min. SNR [dB]	Dämpfung [dB]
Standard	-1,7	144,9
Sendewiederholung	-0,8	144,9
Latenz	-1,6	142,9

In Abb. 65 sind die Modi mit dem Standard-Modus anhand des Energiebedarfs und

der Latenz vergleichend dargestellt. Hierbei wird sich nur auf das TX-Interval bezogen, insbesondere aufgrund des relativ hohen Ruhestroms und unveränderter Ströme und Zeiten für die Empfangsphase.

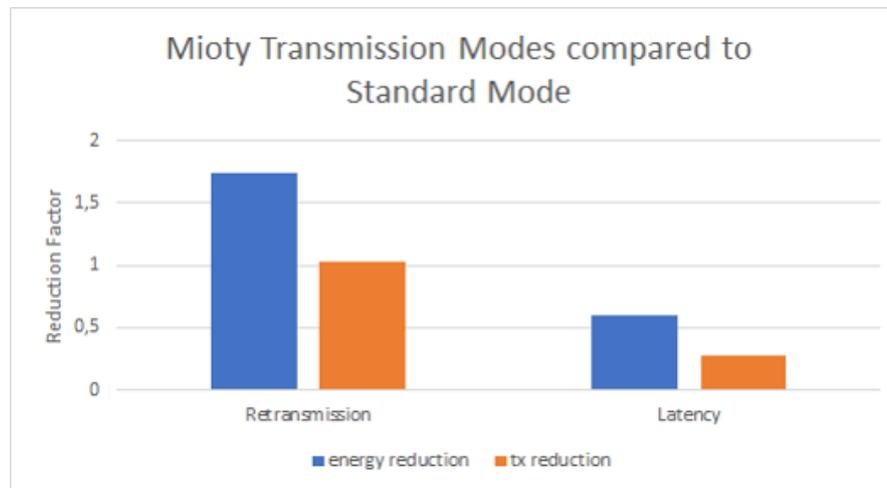


Abbildung 65: MIOTY: Vergleich der Sendemodi

Das Senden im Latenzmodus reduziert die Latenz um 70% und den Energiebedarf um 40%. Der Latenzmodus ist in Kombination mit geschossenem Empfangsfenster sinnvoll um maximal Energie zu sparen. Im Retransmission-Modus verändert sich die Sendezeit nicht. Der Energiebedarf steigt allerdings um 70% an. Bezüglich der Reichweite konnten keine nennenswerten Vorteile gemessen werden. Der SNR blieb praktisch unbeeinflusst und eine Verschlechterung der Kanalbedingungen konnte ebenfalls nicht erreicht werden.

Energiebedarf und Lebensdauer Für die Abschätzungen von Lebensdauern wurden die Ströme, Zeitintervalle und Energiebedarfe für die unterschiedlichen Zustände TX, RX_WAIT, RX, IDLE erfasst. Für die TX-Phase wurden 32 B Daten gesendet. Die Berechnung nimmt die erfassten Ströme als Grundlage und passt die Idle Phase an die Sendeperiode an. MIOTY belegt trotz hoher Latenzen den Kanal nur geringfügig, sodass aus regulatorischer Sicht je nach Paketgröße auch Intervalle im Minutenbetrieb möglich sind.

Beim gemessenen Ruhestrom ergeben sich für alle Szenarien Lebensdauern im Bereich von unter 20 Tagen. Wird ein mittlerer Strom von 3 μA angenommen, wie es beispielsweise von NB-IoT umgesetzt werden kann, erreicht man Lebensdauern von bis zu 16 Jahren bei einem Sendeintervall von 24 h. Ein Abschalten der Geräte bei diesen Sendeintervallen ist ebenfalls möglich – wurde in dieser Kampagne aber nicht weiter betrachtet. Die benötigte Energie wird folgendermaßen abgeschätzt.

$$\bar{E}_{\text{cycle}} = U_{\text{bat}} \cdot (\bar{I}_{\text{tx}} \cdot t_{\text{tx}} + \bar{I}_{\text{rxwait}} \cdot t_{\text{rxwait}} + \bar{I}_{\text{rx}} \cdot t_{\text{rx}} + \bar{I}_{\text{idle}} \cdot (t_{\text{period}} - t_{\text{tx}} - t_{\text{rxwait}} - t_{\text{rx}}))$$

Die Ergebnisse für Senden mit anschließender Öffnung des Empfangsfensters sind in

Abb. 66 dargestellt. Die Verwendung unterschiedlicher Modi wirkt sich vor allem auf kürzere Sendeintervalle aus. Im Latenzmodus können so 0.5–2 Jahre zusätzliche Lebensdauer gewonnen werden, während sich die Lebensdauer bei der Verwendung der Sendewiederholungen um etwa 1–1.5 Jahre verkürzt. Die Vermeidung des Empfangsfensters kann in dieser Berechnung die Lebensdauer nur minimal um wenige Monate erhöhen.

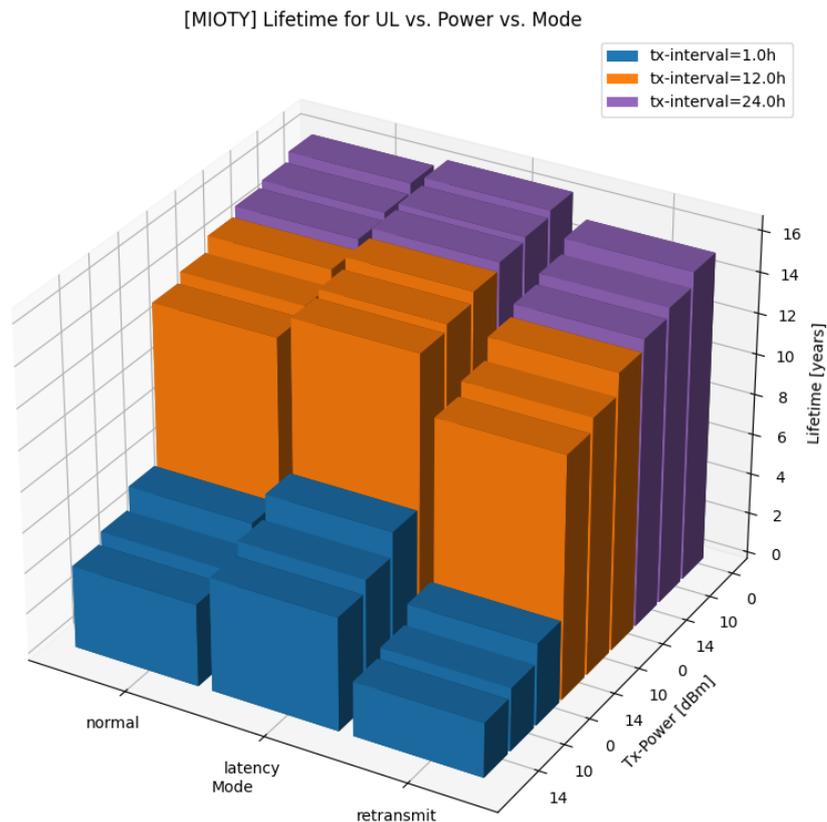


Abbildung 66: MIOTY: Lebensdauer vs. Sendeleistung vs. Sendemodus

Bezug zu Anwendungsfällen Aufgrund der hohen Latenzen spielen für MIOTY praktisch nur die Anwendungsfälle PA2 und MM2 eine Rolle. In den Tests konnte MIOTY eine gute Performanz und ein stabiles Verhalten unter verschiedenen Bedingungen zeigen. Die unterschiedlichen Modi haben im Bezug auf die Anwendungsfälle keine relevante Latenzreduktion oder Reichweitenerhöhung bezwecken können. Im Bezug auf den Energiebedarf erweist sich der Latenzmodus als sparsamste Variante, wobei es diesen Vorteil nur bei vergleichsweise kleinen Sendeintervallen ausspielen kann, ähnlich wie eine Vermeidung des Empfangsfensters. Für die Anwendungsfälle PA2 und MM2

können die Lebensdaueranforderungen erfüllt werden, sofern die Paketgröße auf die maximale Paketgröße von MIOTY begrenzt wird.

Da MIOTY eine relativ neue Technologie ist, gibt es praktisch keine verfügbaren wissenschaftlichen Bewertungen oder gar Simulationsmodelle. Daher konnten die Skalierbarkeitsanforderungen nicht bestätigt werden. Aufgrund des Telegram Splitting Verfahrens, welches einen Verlust von bis zu 50% der Packetbursts zulässt, und nur minimalem Verkehr im Downlink ist die Skalierbarkeit aber als besser gegenüber LoRaWAN einzustufen.

Aufgrund des einfachen Protokolls ist es vergleichsweise gut möglich, eine Vorhersage zur Lebensdauer zu treffen und entsprechende Maßnahmen frühzeitig einzuleiten. Es sollten jedoch klare Anforderungen und Verkehrsdefinitionen dargestellt werden, um eine idealisierte Konfiguration vornehmen zu können.

Bezüglich der Sendeleistung sollten die schlechtesten Verbindungseigenschaften (SNR, RSRP) bestimmt werden in denen der Sensor noch senden soll. Ist je nach Anwendung mehr als 10–20 dB Rand zum technologischen Limit vorhanden, kann die Sendeleistung entsprechend reduziert werden. In den Tests konnte erst bei SNR Werten sehr nahe der Zellengrenze ein Einfluss und Paketverlust festgestellt werden. In realen Umgebungen sind je nach Szenario starke Schwankungen zu erwarten, die in einer Bestandsaufnahme erfasst werden müssen. Aufgrund der statischen Einstellung im UE muss entweder immer mit dem Worst-Case gerechnet werden, oder aber die Anwendung definiert eigenständige Nachrichtenformate, die der UE-Anwendung eine Anpassung der Signalstärke erlauben. Wir halten es weiterhin für sinnvoll, dass eine UE-Applikation auch unabhängig vom Sendeprozess Daten empfangen kann. Synchronisierungsmaßnahmen gestalten sich aber aufgrund von Regulierungen schwierig.

10.4. Optimierungspotentiale

Als Abschluss der durchgeführten Tests können einige Parameter identifiziert werden, die einen größeren Einfluss auf bestimmte Leistungsparameter haben. Zur Ableitung der wichtigsten Parameter für ein Optimierungskriterium wird für jeden Parameter vermerkt, welchen Einfluss er auf die KPIs hat.

In Abb. 67 ist der Einfluss der jeweiligen Technologieparameter auf die untersuchten KPIs dargestellt. Der Einfluss ist in drei Stufen charakterisiert. Wird beim Setzen oder beim Erhöhen eines Parameters auch die KPI erhöht, ist dies durch das Zeichen \uparrow markiert. Verringert sich die KPI, ist es durch das Zeichen \downarrow markiert. Durch diese Beschreibung kann schnell erkannt werden, wie stark und in welche Richtung ein Parameter die KPIs beeinflusst. Die Tests und Resultate zu den jeweiligen Parametern können aus den vorigen Unterkapiteln bezogen werden.

Technology	Parameter	KPI		
		Latency	Range	Energy
NB-IoT	P_MAX	↑	↑↑	↑
	Packetsize	↑↑	-	↑↑
	RAI	↑	-	↓↓↓
	HARQ	↓↓↓	↑	↓↓
	PSM	↑	-	↓↓↓
	Ressources	↓↓↓	↓↓	↓↓
	Repetitions	↑↑↑	↑	↑↑
LTE	TX-Power	-	↑↑	#
	Packetsize	-	-	#
	Bandwidth	↓	↓	#
	Harq	-	-	#
	Modulation	↓↓	↓	#
Lora	TX-Power	-	↑↑↑	↑↑
	Packetsize	↑↑	-	↑↑
	SF	↑↑	↑↑	↑↑↑
	Repetitions	↑	↑	↑↑↑
	ACK	-	-	-
MIOTY	TX-Power	-	↑↑	↑↑↑
	Packetsize	↑↑	-	↑↑
	Mode	↕↕↕	↑	↑↑
	RX-Window	-	-	↑

Abbildung 67: Übersicht zum Einfluss der variierten Parameter auf die KPIs

- ↑: Erhöhung des Parameters erhöht die KPI
 ↓: Erhöhung des Parameters verringert die KPI
 -: kein Einfluss; #: nicht gemessen

Nachfolgend sollen Fragen zur Optimierung und verbleibende Projektfragestellungen beantwortet werden:

Mit welchen Anpassungen könnten die verfügbaren NB-WWAN-Technologien so erweitert und optimiert werden, dass sie für die in der industriellen Automatisierungstechnik relevanten Anwendungen möglichst effizient eingesetzt werden können? Anhand der Parametervariationen und der Übersicht in Abb. 67 konnten Profile spezifisch für die Anwendungsfälle erstellt werden, wie in Abb. 68 mit den entsprechenden Parametern dargestellt. Hierbei muss beachtet werden, dass die FA-Anwendungsfälle mit den NB-WWAN Technologien nicht umgesetzt werden können.

Die so gefundenen Profile werden mit den entsprechenden Anwendungsfallparametern getestet und schließlich im Bezug auf die Anwendungsfälle verifiziert. Aufgrund zahlreicher bereits durchgeführter Parametervariationen unter unterschiedlichen Bedingungen konnten die Ergebnisse zu einem Großteil auf die Anwendungsfälle bezogen werden. Für NB-IoT wurden zusätzliche Tests mit den dargestellten Parameterkombinationen durchgeführt, die noch nicht vorher durchgeführt worden sind.

Technology	PA1	PA2	MM1	MM2
NB-IoT	HARQ: 2 RAI: OFF PSM: OFF Subcarrier: 12 Scheduler: dynamic Repetitions: 1 PMAx: 23	HARQ: 2 RAI: ON PSM: ON Subcarrier: 6 Scheduler: dynamic Repetitions: 8 PMAx: 23	HARQ: 2 RAI: OFF PSM: OFF Subcarrier: 12 Scheduler: dynamic Repetitions: 2 PMAx: 23	HARQ: 2 RAI: ON PSM: ON Subcarrier: 1 Scheduler: dynamic Repetitions: 32 PMAx: 23
LoRa	SF: SF7-8 TX-Power: 14dBm Repetitions: 1-2 by priority ACK: ON	SF: SF9-12 oder ADR TX-Power: 10dBm Repetitions: 1-3 by priority ACK: OFF or only by priority	SF: SF7-9 TX-Power: 14dBm Repetitions: 1-2 by priority ACK: OFF or only by priority	SF: SF9-12 oder ADR TX-Power: 14dBm Repetitions: 1-4 by priority ACK: OFF or only by priority
MIOTY	Mode: Latenz TX-Power: 10dBm RX-Window: ON	Mode: Latenz TX-Power: 14dBm RX-Window: ON	Mode: Latenz TX-Power: 14dBm RX-Window: ON	Mode: Normal TX-Power: 14dBm RX-Window: ON

Abbildung 68: Anwendungsfallsspezifische Profile

Kann mit diesen Ansätzen eine zuverlässige und hart echtzeitfähige Funkkommunikation im Industriebereich unterstützt werden? Eine zuverlässige und hart echtzeitfähige Funkkommunikation im Industriebereich kann mit den untersuchten Technologien nicht unterstützt werden. Abseits von LTE sind die notwendigen Sendeintervalle und Latenzen nicht mit den vorhandenen Technologien umsetzbar. Trotz hoher Datenraten und geringer Latenzen mit LTE konnten in den Tests zahlreiche Ausreißer erzeugt werden. Eine harte Echtzeitfähigkeit ist trotz eines privaten Campusnetzwerkes mit einer Vielzahl möglicher Einstellparameter nicht oder nur eingeschränkt möglich. Die QoS Mechanismen in LTE sind nicht auf solche Anforderungen ausgelegt. Der neue 5G Standard ist teilweise an industriellen Anforderungen orientiert, und es sind erweiterte QoS Mechanismen definiert. Auch hier sind jedoch Erweiterungen für eine hart echtzeitfähige Funkkommunikation notwendig und geplant, die zum aktuellen Zeitpunkt aber nicht verfügbar sind.

Welche Anwendungen können mit diesen Leistungsdaten sinnvoll unterstützt werden? In Abb. 69 sind die Einschränkungen zur Erfüllbarkeit der Anwendungsfälle dargestellt. Grün bedeutet, dass die Technologie die Anwendung theoretisch unterstützen kann, gelb bedeutet, dass nicht alle Anforderungsparameter verifiziert werden konnten und rot bedeutet, dass es mindestens eine Einschränkung gibt, die den Einsatz

verhindert. Es muss beachtet werden, dass insbesondere die Skalierbarkeitsanforderungen nur innerhalb von LTE und NB-IoT näherungsweise betrachtet werden konnte, während diese Betrachtung bei LoRa und MIOTY nicht möglich war.

Technologie	FA1	FA2	PA1	PA2	MM1	MM2
LTE	Latenz: zu hoch Intervall: zu klein	In Simulation: Ja In Emulation: Intervall zu klein	Nicht untersucht, da Energiebedarf zu hoch	Nicht untersucht, da Energiebedarf zu hoch	Nicht untersucht	Nicht untersucht, da Energiebedarf zu hoch
NB-IoT	Latenz: zu hoch Intervall: zu klein	Latenz: zu hoch Intervall: zu klein	Intervall: Energiebedarf zu hoch	Skalierbarkeit: in Simulation mit 100 Klienten verifiziert	Oberes Limit der Latenz	Skalierbarkeit: in Simulation mit 100 Klienten verifiziert
LoRa	Latenz: zu hoch Intervall: zu klein	Latenz: zu hoch Intervall: zu klein	Intervall: Energiebedarf zu hoch	Skalierbarkeit: nicht verifiziert	Paketgröße: max. 150B bei ben. Latenz Intervall: Duty-Cycle	Skalierbarkeit: nicht verifiziert
MIOTY	Latenz: zu hoch Intervall: zu klein	Latenz: zu hoch Intervall: zu klein	Latenz: zu hoch Intervall: Duty Cycle	Skalierbarkeit: nicht verifiziert	Latenz: zu hoch Intervall: Duty Cycle	Skalierbarkeit: nicht verifiziert

Abbildung 69: Erfüllbarkeit der Anwendungsfälle

grün: erfüllbar

gelb: erfüllbar aber nicht komplett getestet

rot: nicht erfüllbar (mindestens ein Parameter außerhalb der Kriterien)

Die gezeigten Parametrierungen ermöglichen, dass einige der industriellen Szenarien theoretisch unterstützt werden können. Insgesamt sind die Parametrierungsmöglichkeiten aber durch die technischen Grundfähigkeiten der Protokolle beschränkt, sodass die untersuchten Technologien für die meisten Anwendungen der industriellen Automatisierungstechnik ungeeignet sind. Abseits von LTE können die Technologien nicht in den Bereich der Fabrikautomation vordringen. So ergeben sich hauptsächlich Anwendungen im Bereich der Prozessautomatisierung wie der Zustandserfassung, der Predictive Maintenance und der Warenverfolgung. In diesen Anwendungen kann von den hohen Reichweiten und geringen Energieaufnahmen profitiert werden. Die unterstützten Anwendungen beziehen auf einfache Verkehrsmuster mit relativ großen Sendeintervallen im Bereich von >1 h.

Obwohl NB-IoT keinen regulatorischen Beschränkungen unterliegt und theoretisch auch mit Sendeintervallen im Bereich von 300–500 ms Bereich gesendet werden kann, ist diese Möglichkeit durch Anforderungen an die Batterielebensdauer stark beschränkt. Praktisch ist ein jahrelanger Batteriebetrieb nur unter Nutzung des PSM und RAI möglich. Hierbei muss das RAI Feature erst innerhalb der Basisstation eingestellt werden. In den Untersuchungen wurde ein deutlicher Komplexitätsunterschied zwischen den zellulären und den anderen Technologien deutlich, der beim Betrieb privater Netzwerke beachtet werden muss. Der Zusammenhang und die Auswirkungen einzelner Netzwerkparameter auf KPIs ist nur unter viel Aufwand zu ermitteln.

Hohe Datenraten und ständige Verfügbarkeit für Downlink Nachrichten kann NB-IoT als Konkurrenz zu anderen energiesparenden Lösungen nicht ausspielen. Im Vergleich zu den anderen LPWA Technologien zeichnet sich NB-IoT durch eine sehr gute Zuverlässigkeit, die größere Payload- und Intervallflexibilität und ein lizenziertes Spektrum aus, welches sich insbesondere positiv auf die Skalierbarkeit auswirken kann. NB-IoT kann die anderen LPWA dort ergänzen wo Flexibilität bezüglich der Sendeintervalle und der

Payloadgrößen notwendig ist und ein lizenziertes Spektrum Vorteile verschaffen kann. Aufgrund der höheren Kosten und einer erhöhten Komplexität sind LoRa und MIOTY als gute Alternative anzusehen, sofern klare Intervalle in einer Größenordnung von >1 h und kleinere Paketgrößen von einer Anwendung umsetzbar sind und gleichzeitig höhere Lebensdauern von Bedeutung sind.

Für LoRa können mittels unterschiedlicher Spreizfaktoren und Sendeleistungseinstellungen große Bereiche von unterschiedlichen Datenraten und Reichweiten abgedeckt werden, wodurch sich eine Vielzahl an möglichen Anwendungen ergeben. Im Vergleich zu NB-IoT ist es stromsparender, sodass auch kürzere periodische Sendeintervalle bei einem geringen Spreizfaktor möglich sind, um eine mehrjährige Lebensdauer zu erreichen. Die regulatorischen Beschränkungen im öffentlichen Spektrum limitieren die Flexibilität bezüglich des Sendezeitpunktes, sodass ereignisorientierte Anwendungen mit deterministischen Übertragungszeiten nur schlecht unterstützt werden können. Weiterhin können theoretische Datenraten durch diese Regulierungen nicht erreicht werden. Es eignet sich für Anwendungen mit mittlerer Kapazität, Sendeintervalle >30 min bei mehrjähriger Batteriedauer und geringem Kosteneinsatz.

MIOTY kann die Duty Cycle Beschränkungen durch das Telegram-Splitting gut nutzen, sodass trotz einer vergleichsweise sehr hohen Latenz ein relativ kurzes Sendeintervall verwendet werden kann. MIOTY eignet sich dennoch nur für Anwendungen mit großen Sendeintervallen und geringen Anforderungen an die Latenz. In diesen Intervallen zeigen sich ähnliche Lebensdauern wie für LoRa. In den Tests zeigte sich MIOTY sehr zuverlässig und stabil unter verschiedenen Bedingungen. Es ist besonders für einfache Abläufe geeignet, die hauptsächlich Uplink-Fähigkeiten erfordern. Die fehlende Möglichkeit, Downlinknachrichten unabhängig von Uplinknachrichten zu empfangen, macht diese Technologie im Vergleich zu den Technologien relativ unflexibel gegenüber möglichen Anwendungen.

In welcher Weise können LPWAN-Technologien mit zellularen Mobilkommunikationstechnologien konkurrieren, bzw. diese sinnvoll ergänzen? Die zellulären Technologien zeichnen sich vor allem durch höhere Datenraten, keine regulatorischen Einschränkungen und ständige Downlinkverfügbarkeit aus. Dies kann für einen langjährigen Batteriebetrieb allerdings nicht ausgespielt werden. NB-IoT kann die anderen LPWA dort ergänzen, wo Flexibilität bezüglich der Sendeintervalle und der Payloadgrößen notwendig ist und ein lizenziertes Spektrum Vorteile verschaffen kann. Bei Anwendungen, die flexible Sendeszenarien oder geringe Sendeintervalle benötigen, aber die Möglichkeit besitzen in regulären Intervallen aufgeladen zu werden, bietet NB-IoT eine sehr attraktive Lösung, die jedoch klaren Beschränkungen bezüglich der untersuchten KPIs unterliegt. Sofern keine komplexen oder flexiblen Szenarien notwendig sind und klare Sendeintervalle in einer Größenordnung von >1 h und kleinere Paketgrößen im Bereich von 50 B von einer Anwendung umsetzbar sind und gleichzeitig höhere Lebensdauern von Bedeutung sind, sind LoRa und MIOTY als gute Alternative anzusehen. Hierbei bietet LoRa eine etwas höhere Flexibilität bezüglich der Parametrisierung und ist zudem schon länger ausgereift, wodurch die Hardware- und Softwareverfügbarkeit deutlich hö-

her ist. Somit können sich die zellulären und LPWAN-Technologien durchaus ergänzen und die Wahl hängt sehr stark von der entsprechenden Anwendung ab.

Wie kann die Leistungsfähigkeit einer Technologie, eines Produkts oder einer Installation durch Simulationsmodelle effizient vorhergesagt werden? In diesem Projekt zeigten sich die Simulationsmodelle vor allem hilfreich zur Abschätzung der Grundfähigkeiten der Protokolle und als Referenz für die Verhaltensweise eines Protokolls. Durch Simulationen auf Systemebene können somit die wichtigsten und grundlegenden Zusammenhänge erfasst werden. Sie stellen eine klare Bereicherung für den Testprozess dar, da einige Zusammenhänge in Tests mit realen Geräten so nicht erfasst werden können, neue noch nicht verfügbare Features getestet werden können und gleichzeitig relativ schnell umfangreiche Studien durchgeführt werden können, sofern eine geeignete Simulation vorhanden ist. Es wurde aber deutlich, dass sich der Detailgrad verfügbarer Simulationen deutlich unterscheidet und bei der Betrachtung der Ergebnisse genau berücksichtigt werden muss. Eine 1:1 Beziehung zu Feld- oder Emulationstests ist nicht möglich. Die Simulation kann nur ein Teil eines größeren Gesamttestprozesses sein. Der vorgestellte dreistufige Prozess erwies sich als sehr hilfreich bei der Evaluierung der Technologien.

Welchen Einfluss könnten und sollten diese Erfahrungen auf die zukünftige Standardisierung sowohl des 3GPP als auch der LPWAN-Technologien haben? In diesem Projekt wurden Netzwerkparameter untersucht und auf bestimmte KPIs hin optimiert. Die ermittelten Ergebnisse und Technologieprofile (vgl. Abb. 68) sind insbesondere für Anwender interessant, aber auch für Standardisierungsinstitutionen. Ein Problem insbesondere bei der Parametrierung privater Campus-Netzwerke ist die fehlende Transparenz bezüglich der Möglichkeiten einzelner Parameterkombinationen. Die in diesem Projekt ermittelten Parametersätze ermöglichen eine Empfehlung vorgefertigter Konfigurationen und Möglichkeiten dieser Einstellungen.

11. Zusammenfassung

Im Projekt TestKomm 4.0 wurde eine systematische Testumgebung aufgebaut, die es erlaubt, automatisierte Tests auf unterschiedlichen Abstraktionsebenen für unterschiedliche Technologien durchzuführen. Neben zellularen Funkkommunikationslösungen wie LTE und NB-IoT wurden auch LPWA-Technologien wie MIOTY und LoRa untersucht. Zur Umsetzung des Testsystems wurden verschiedene Komponenten entwickelt, darunter ein Simulator, ein Emulator, eine zentrale Teststeuerung und ein Visualisierungswerkzeug. Die so entstandene einheitliche Testumgebung wurde im Laufe des Projektes in zahlreichen Testkampagnen mit unterschiedlichen Technologien getestet und ihre Funktionalität validiert.

Im Rahmen des Projektes wurden Tests auf Feld-, Emulations- und Simulationsebene für verschiedene Technologien durchgeführt, um sowohl die Technologien als auch die Abstraktionsebenen zu vergleichen. Parameterstudien wurden an den verschiedenen Technologien durchgeführt, um Optimierungspotentiale im Bezug auf ausgewählte Indikatoren zu identifizieren.

Die Ergebnisse wurden auf ausgewählte industrielle Anwendungsfälle bezogen und im Hinblick auf deren Anforderungen bewertet und validiert.

11.1. Fazit

Das Ziel des Forschungsvorhabens wurde erreicht.

Die entwickelten Testumgebungen erlauben ein systematisches und automatisiertes Testen verschiedener Funktechnologien auf unterschiedlichen Abstraktionsebenen. Die im Projekt erstellte Testumgebung ermöglicht damit das Testen und Verifizieren unterschiedlichster Anwendungsfälle, von der Fabrikautomation bis zur M2M-Kommunikation. Mittels der entwickelten Testkomponenten konnten die drahtlosen Kommunikationslösungen bezüglich den im Projekt betrachteten Anwendungsfällen systematisch untersucht und verifiziert werden. Die im Projekt untersuchten Technologien können den industriellen Anforderungen nur teilweise gerecht werden. Die Technologien scheitern vor allem an den hohen Anforderungen an die Latenz innerhalb der Fabrikautomation. Die Technologien schaffen es nicht, die Brücke über die unterschiedlichen Anwendungsfälle zu schlagen. Vielmehr eignen sich die Lösungen für ein bestimmtes Teilanwendungsgebiet. Diese Teilgebiete und eine mögliche Ergänzung zwischen LPWAN und cloT Lösungen konnten im Projekt herausgearbeitet werden. Über die herausgearbeiteten technologischen Unterschiede können generelle Aussagen zur Nutzbarkeit einer Technologie für einen Anwendungsfall getroffen werden. Weiterhin können spezifische Anforderungen über das entwickelte Testsystem systematisch überprüft werden.

Die unterschiedlichen verfügbaren Testebenen verbessern den Gesamttestprozess deutlich.

Auf der Simulationsebene können detaillierte Analysen der Protokolle durchgeführt werden und es können wichtige Zusammenhänge für den Entwurf weiterer Testfälle

entworfen werden. Weiterhin erlauben die Simulationsmodelle die Durchführung weiterführender Tests und Szenarien, die mit realen Hardwarekomponenten kaum ausführbar sind. Je nach Modellgenauigkeit sind die Ergebnisse auf Simulationsebene nur bedingt auf reale Feldgeräte übertragbar. Somit können Simulationsmodelle insbesondere zu Beginn einer Evaluation Zeit sparen, die dann sinnvoll für Tests auf anderen Ebenen eingesetzt werden kann.

Das Emulations-Testbed ermöglicht die systematische Variation passiver Kanaleigenschaften, wodurch unterschiedlichste Szenarien modelliert werden können. Durch die erhöhte Flexibilität des neu entwickelten Aufbaus können Emulationsumgebungen einfach aufgebaut und erweitert werden. Durch die im Projekt entwickelten Mehrwege-Komponenten lassen sich zudem komplexere Szenarien zum Modellieren der Umgebung einstellen. Aufgrund vergleichbarer Ergebnisse innerhalb der Emulationsumgebung und der Feldtestumgebung bietet diese Testebene einen guten Abstraktionsgrad. Aufgrund des hohen Automatisierungsgrades und des einfachen Testaufbaus können auf dieser Ebene sehr gut Parametervariationen und Evaluationen der Technologien unter unterschiedlichen Bedingungen durchgeführt werden. Diese Testebene eignet sich für die Ausführung der meisten Testfälle.

Die durchgeführten Tests auf der Feldebene zeigen, dass sich die Technologien grundsätzlich unterscheiden und deutlich andere physische Parameter an den Testpositionen gemessen werden. Aufgrund der vielen technologischen Unterschiede, muss daher jede Technologie selbst im Feld gemessen werden, um das reale Verhalten in der konkreten Umgebung verstehen zu können. Diese Testumgebung ist wichtig für die Erfassung der physischen Parameter und der Grenzbedingungen für einen konkreten Anwendungsfall und ist daher insbesondere am Ende einer Untersuchung, zur Verifizierung eines Anwendungsfalles, notwendig.

11.2. Nutzen für KMU, innovativer Beitrag und industrielle Anwendungsmöglichkeiten

Die Ergebnisse des Projektes zeigen deutlich, dass die Entwickelte Testumgebung, sowie die damit im Projekt beispielhaft erzeugten Analysen einen großen Nutzen für die Industrie und insbesondere für KMU liefern. Die Ergebnisse zeigen deutlich, dass Tests mit realen Geräten für detaillierte Analysen unabdingbar sind, da z.B. einige Ergebnisse nicht immer den Erwartungen entsprechen. Zudem wurde im Projektverlauf deutlich, dass Emulationen eine wichtige und sinnvolle Abstraktion für Tests bieten, da sie insbesondere reproduzierbare, systematische und schnelle Evaluationen ermöglicht und im Gegenzug Simulationen vor allem für das Technologieverständnis und für Skalierbarkeitstests wichtig sind.

Somit wird die Sinnhaftigkeit der hier entwickelten, dreischichtigen Testausführung bestehend aus Simulation, Emulation und Feldtests weiter gestärkt und zeigt die Relevanz der entwickelten automatisierten Testumgebung über die Abstraktionsebenen hinweg. Dies macht solch ein Testsystem gerade auch für KMU interessant. Neben dem direkten Vorteilen des entwickelten Systems, wie das automatisierte Testen, ergeben sich auch weitere Nutzen durch das Testsystem. Dieses liefert ein Framework zur Beschreibung

von Testfällen und Komponenten. Es liefert eine klare Struktur bei der Entwicklung von Testfällen und integriert alle notwendigen Werkzeuge um Funktechnologien aus Anwenderperspektive zu evaluieren. Damit ist das Testsystem vor allem in der Praxis für reale Anwender interessant, die neue Anwendungen oder neue Funktechnologien testen möchten.

Die entwickelten Lösungen erlauben es KMU

- unternehmensspezifische Anwendungsfälle zu beschreiben, zu testen und zu verifizieren,
- kostengünstig Tests unter realen Umgebungen durchzuführen,
- Funknetzwerke mit realen Kanaleigenschaften über eine intuitive Benutzeroberfläche zu konfigurieren,
- Tests automatisiert und reproduzierbar im Labor durchzuführen,
- Parameterstudien durchzuführen, um optimale Parametersätze für einen Anwendungsfall zu finden,
- kurze Feedbackschleifen beim Testprozess zu erzeugen,
- über den Entwicklungsprozess hinweg Tests durch Nutzung unterschiedlicher Abstraktionsebenen durchzuführen,
- schnell neue Technologien zu integrieren und zu testen.

Das Testsystem reduziert insbesondere die Testzeit sowie die Entwicklungszeit für neue Tests. Es ermöglicht KMU auch Technologien zu evaluieren, die unter Umständen aus Zeitgründen nicht hätten betrachtet werden können.

Der innovative Beitrag ist auf mehreren Ebenen sichtbar. Zum Einen wurden drei unterschiedliche Domänen (Simulation, Emulation, Feld) betrachtet, analysiert und in ein Gesamtsystem integriert, um schließlich einen Mehrwert zu generieren. Die dadurch möglich werdenden Betrachtungsmöglichkeiten gehen über die Möglichkeiten einzelner Teillösungen heraus. Zudem konnte durch den weiterentwickelten Emulator dessen Mehrwert, aufgrund der flexiblen Konfigurationsmöglichkeiten sowie der dynamischen Ansteuerung der HF-Elemente, deutlich gesteigert werden. Dies ermöglicht neben statischen Tests nun auch Tests unter sich ändernden Umgebungsbedingungen. Über die erweiterte Mehrwegeumgebung können zudem gezielt Extremfälle erzwungen werden und ermöglichen es somit Worst-Case Betrachtungen durchzuführen.

Weiterhin wurden unterschiedliche LPWAN und IoT Lösungen sowie typische Anwendungsfälle untersucht, um repräsentative Anwendungsabläufe und Testfälle zu beschreiben. Die so entstandene Struktur zur Beschreibung von Tests innerhalb dieser Domänen unterstützt einen Entwickler bei der Erstellung neuer Testbeschreibungen oder bei der Integration neuer Technologien.

Schließlich konnten durch die ausgeführten Testkampagnen Unterschiede und Möglichkeiten der Funktechnologien aufgezeigt werden und durch die generische Testbeschreibung systematisch miteinander verglichen werden.

Industrielle Anwendungsmöglichkeiten ergeben sich entsprechend in den folgenden Bereichen:

- Verifizierung von Anwendungsfällen.
 - Schnelle Abschätzung einer Technologie für einen Anwendungsfall.
 - Parameterstudien zur systematischen Evaluierung einer Technologie
 - Evaluation von statischen und mobilen Anwendungen
- Systematische Untersuchung von Umgebungsbedingungen auf die Performanz einer Anwendung.
- Nutzung als Referenzplattform, z.B. zur Evaluation von eigenen Anwendungen, Simulationsmodellen etc.

11.3. Mögliche zukünftige Arbeiten

Mithilfe der in diesem Projekt entwickelten Testkomponenten können drahtlose Kommunikationstechnologien systematisch mit vergleichsweise kleinem Zeitaufwand getestet werden. Dies wurde durch mehrere Testkampagnen validiert. Dadurch ist es möglich, in Zukunft konkrete Anwendungsfälle aus der Industrie mit vergleichsweise geringem Aufwand zu untersuchen und zu verifizieren.

Die Testumgebung soll daher weiterhin für die Evaluierung von Funklösungen verwendet werden. Weiterhin soll die Testumgebung für weitere Technologien und KPIs erweitert werden. Insbesondere sollen automatisierte Optimierungsmöglichkeiten umgesetzt werden, um umfangreiche Parameterstudien hoch automatisiert ausführen und analysieren zu können.

Durch eine Erweiterung des Automatisierungsgrades der Testausführung könnten Parameterstudien vollautomatisch durchgeführt und analysiert werden sowie anhand der ermittelten Daten automatisch eine optimale Parameterkombination bezüglich eines oder mehrerer Leistungsparameter gefunden werden. Dies ist insbesondere bei den immer komplexer werdenden Protokollen wichtig, die möglichst flexibel an gewisse Situationen anpassbar sein sollen.

Dynamische Kanalanpassungen sind mit dem aktuellen Aufbau des Emulationstestbeds möglich. Die Modellierung dynamischer Umgebung im APTB sollte systematisch untersucht und entsprechend implementiert und getestet werden, um auch solche mobilen Anwendungsfälle modellieren zu können.

A. Verwendung der Zuwendung

FE	Wissenschaftlich-technisches Personal (Einzelansatz A.1 des Finanzierungsplans)	Geräte (Einzelansatz B des Finanzierungsplans)	Leistungen Dritter (Einzelansatz C des Finanzierungsplans)
FE1	35,47 PM	Nein	Nein
FE2	34,10 PM	Nein	Nein

B. Notwendigkeit und Angemessenheit der geleisteten Arbeiten

Die geleistete Arbeit entsprach in vollem Umfang dem begutachteten und bewilligten Antrag und war damit notwendig und angemessen.

C. Plan zum Ergebnistransfer in die Wirtschaft

Im Folgenden werden die einzelnen durchgeführten und noch geplanten Maßnahmen zum Ergebnistransfer tabellarisch aufgelistet. In Tab. 45 sind die durchgeführten Transfermaßnahmen und in Tab. 46 sind die geplanten Transfermaßnahmen nach Projektende dargestellt.

Übersicht auf die Transfermaßnahmen nach Projektende

Der Transfer der Testumgebung in die Industrie wurde aufgrund dessen Komplexität nicht innerhalb der Projektlaufzeit durchgeführt. Es wird nach Bedarf bzw. abhängig von der konkreten Nachfrage für die Firmen des PA und sonstige Interessierte angeboten. Der Bedarf bzw. die Ausgestaltung möglicher Schulungen und Informationsveranstaltungen wird nach Projektende insbesondere auch abhängig vom Feedback der beteiligten Unternehmen bzw. der Unternehmen aus den Netzwerken der FEs und der Forschungsvereinigung auf die erzielten Projektergebnisse konkret ermittelt. Aufgeführte sonstige Maßnahmen (z.B. Workshops) rund um den allgemeinen Wissenstransfer in die Industrie befinden sich seit Projektende in Vorbereitung und werden noch weiter konkretisiert.

Tabelle 45: Durchgeführte Transfermaßnahmen

Maßnahme	FE	Ziel	Ort/Rahmen	Datum/Zeitraum
Wissenstransfer innerhalb des PA	Hahn-Schickard, ivESK	Ergebnisberichte, Workshops, bilaterale Beratungsgespräche	Kickoff Web-Meeting	28. April 2021
			Sitzung des PA als Web-Meeting	25. März 2022
			Sitzung des PA als Web-Meeting	13. Oktober 2022
			Abschließende Sitzung des PA als Web-Meeting	27. März 2023
Feldtests	ivESK	Referenztests mit unterschiedlichen drahtlosen Technologien ohne automatisiertes Testsystem	Bei einer interessierten Firma (nicht Mitglied des PA)	September 2021
Wissenschaftliche Veröffentlichungen	ivESK	Fachlicher Austausch, Erkenntnisgewinn	Veröffentlicht in „Technologien für die intelligente Automation“ [74]	August 2022
			Vorge stellt auf der „INDIN 2023“ Konferenz und noch nicht veröffentlicht [75]	17-20 Juli 2023
Nichtwissenschaftliche Veröffentlichungen	Hahn-Schickard, ivESK	Laufende Berichte über den Fortschritt der Projektarbeiten	Websites von Hahn-Schickard und ivESK	laufende Updates
			Jahresbericht Hahn-Schickard	jährlich/1. Quartal
			Jahresbericht HS-Offenburg (fif)	jährlich/1. Quartal
Wissenstransfer auf Messen und Events	ivESK	Präsentation von Projekthaltungen für ein Publikum aus akademischem und industriellem Umfeld	Embedded World Conference	21. Juni 2022
			Digitalisierungstour BW2021	4. Oktober 2021
Wissenstransfer in der Lehre	ivESK	Wissensvermittlung und Erkenntnisgewinn	Master-Thesis	2020
			Praktikum	März - August 2022
			Projektarbeit	Oktober 2022 - Februar 2023

Tabelle 46: Geplante Transfermaßnahmen (nach Projektende)

Maßnahme	FE	Ziel	Ort/Rahmen	Datum/Zeitraum
Wissenstransfer in die Industrie	Hahn-Schickard und ivESK	Ergebnisberichte, Workshops, bilaterale Beratungsgespräche	Workshops	Nach Projektabschluss
Transfer der Testumgebung	Transfer der Testumgebung in die Industrie	Hahn-Schickard und ivESK	Mitgliedsfirmen des PA und alle interessierten Unternehmen	Nach Bedarf
Schulungen und Informationsveranstaltungen	Hahn-Schickard und ivESK	Transfer des Knowhows zu industriellen Partnern, v.a. zu PA-Mitgliedsunternehmen, aber auch zu weiteren Unternehmen.	eintägige Schulungen, Bereitstellung von Online-Materialien	nach Bedarf

D. Abkürzungsverzeichnis

3GPP 3rd Generation Partnership Project

ADR adaptive Datenrate

APTB Automated Physiscal Testbed

ATS Automatisierte Transportsysteme

AMC Advanced Mezzanine Card

BS Basisstation

cloT cellular IoT

CPRI common public radio interface

COTS commercial off-the-shelf

CSS Chirp-Spread-Spektrum

DL Downlink

DSP Digital Signal Processing

DUT Device Under Test

eDRX extended Discontinuous Reception

ETSI European Telecommunications Standard Institute

FTS Fahrerlose Transportsysteme

EC Element Controller

ED End Device

eNB evolved Node B

ER Extended Range

ERP Effektive Strahlungsleistung

embb eMBBenhanced mobile Broadband

EPA Extended Pedestrian A model

ETU Extended Typical Urban

EVA Extended Verhicular A

gNB 5g base station

EPC	evolved packet core
FPGA	field-programmable gate array
GbE	gigabit ethernet
GPIO	General Purpose Input Output
HARQ	automatic repeat request
HMI	Human Machine Interface
HTTPS	Hypertext Transfer Protocol
ivESK	Institut für eingebettete Systeme und Kommunikationstechnik
IIoT	Industrial Internet of Things
IoT	internet-of-things
KPI	key performance indicator
LPWAN	low power wide area network
LoRa	LoRa
LOS	Line of Sight
LTE	Long Term Evolution
LwM2M	Lightweight M2M
MC	Main Controller
MCL	Maximum Coupling Loss
M2M	Maschine-zu-Maschine
MIOTY	MIOTY
mMTC	massive machine type communication
MQTT	Message Queuing Telemetry Transport
NB-IoT	Narrowband IoT
NB-WWAN	Narrow Band Wireless Area Networks
NLOS	Non Line of Sight
NoSQL	Not Only SQL
NR	New Radio

OFDMA Orthogonal Frequency Division Multiple Access

PA Projekt begleitenden Ausschuss

PoE Power-over-Ethernet

PSM Power Safe Mode

PTC Parallel Test Component

QoS Quality of Service

RACH Random Access Channel

RAI release assistance indication

RAN Radio Access Network

RRC radio resource control

RF radio-frequency

RSRP Reference Signal Received Power

RSSI Received Signal Strength Indication

SA standalone

SDR software-defined radio

SoC system-on-chip

SCPI Standard Commands for Programmable Instruments

SF Spreizfaktor

SDL Specification and Description Language

SDR Software Defined Radio

SDWN Spatially Distributed Wireless Networks

SNR Signal to Noise Ratio

SPS semi persistent scheduling

SRWN Short-Range Wireless Networks

STTI short transmission time interval

TBS Transport Block Size

TCP Transmission Control Protocol

TestKomm 4.0 Integriertes Testbed für die ubiquitäre, durchgängige, echtzeitfähige und zuverlässige Kommunikation für Industrie 40

TDL Test Description Language

TI Texas Instruments

TPL Test Purpose Language

TSDB Time Series Data Base

TSMA Telegram Splitting Multiple Access

TTCN3 Testing and Test Control Notation Version 3

TMC TTCN3 Main Controller

UE User Equipment

UDP User Datagram Protocol

UNB Ultra Narrow Band

UL Uplink

ULP Ultra Low Power

URLLC Ultra-Reliable and Low-latency Communications

PU Periodischer Uplink

AU Aperiodischer Uplink

PD Periodischer Downlink

AD Aperiodischer Downlink

FU Firmware Update

BU Burst Uplink

APU Aperiodischer Uplink

APD Aperiodischer Downlink

UTD Uplink triggert Downlink

DTU Downlink triggert Uplink

E. Abbildungsverzeichnis

1.	Einfluss des Duty-Cycle auf Anwendungen in unlicenzierten Bändern . . .	6
2.	LoRaWAN Netzwerkarchitektur [20]	7
3.	Mioty Netzwerkarchitektur [25]	9
4.	Vereineinfachte 4G Netzwerkarchitektur	12
5.	Netzwerkaufbau der simulierten allgemeinen Star Topology.	39
6.	Klassendiagramm der Beschreibungssprache	40
7.	Systemansicht der Simulationsumgebung, einschließlich externer TTCN3-Schnittstelle	42
8.	Beschreibung einer Simulation	43
9.	Gesamtarchitektur des Testsystems	46
10.	Sequenzdiagramm des Uplink Testfalles	48
11.	Typischer Ablauf eines Testfalles innerhalb des Testsystems	49
12.	Konfigurationsebenen	49
13.	Interfaces und Klassen zur Implementierung der DUTs	51
14.	Interaktion mit dem Simulationsbackend über die REST-API	52
15.	Visualisierung der E2E-Latenz eines Testfalle	55
16.	Frontend zur Konfiguration und Ausführung des Testsystems	56
17.	Hardwarearchitektur des APTBs	57
18.	Aktueller APTB Aufbau in ivESK	59
19.	Softwarearchitektur des APTBs	59
20.	Softwarearchitektur des Main Controllers innerhalb des APTBs	60
21.	Klassendiagramm der Element Controller	61
22.	Beispiel für ein MQTT-Topic, das zur Aktualisierung des Zustands eines HF-Schalters verwendet wird	62
23.	Beispiel für die Verwendung von MQTT-Wildcards zum gleichzeitigen Abonnieren mehrerer Themen	62
24.	Aufbau der Mehrwegeelemente	64
25.	Vergleich des SNRs für unterschiedliche Phasenverschiebungen	65
26.	Vergleich des RSRPs für unterschiedliche Pfadkombinationen	66
27.	Vergleich der ETU Profile	66
28.	Vereinfachtes Beispiel wie Projekte und deren durchgeführte Experimente gesichert werden.	68
29.	Datenformat der visualisierbaren Ergebnisse	69
30.	Systemansicht der Architektur für Projektmanagement und Visualisierung der Daten	70
31.	Schritte zur Auswahl der einzelnen Tests	70
32.	Kurzzusammenfassung der in Listing 1 beschriebenen Simulation.	71
33.	Einige Ergebnisse der in Listing 1 beschriebenen Simulation	72
34.	Konzept der Umgebungsabstraktion	76
35.	Testaufbau mit einem Endgerät	76
36.	Feldtestpunkte	78

37. Signalleistung über Ressourcenblöcke und Zeit des simulierten ETU Profils [68]	82
38. Aktivitätsdiagramm für den Testablauf auf unterschiedlichen Ebenen	88
39. Vergleich der Signalstärke an unterschiedlichen Indoor Positionen	90
40. Vergleich von RSRP und SNR in Feld- und Emulationsumgebung	91
41. Vergleich von RSRP und SNR in Feld- und Emulationsumgebung	92
42. NB-IoT: Vergleich unterschiedlicher Paketgrößen	93
43. NB-IoT: Vergleich unterschiedlicher Paketgrößen in der Simulation	94
44. NB-IoT: Energiebedarf je Byte für unterschiedliche Paketgrößen	95
45. NB-IoT: Vergleich unterschiedlicher HARQ-Einstellungen	97
46. NB-IoT: Vergleich unterschiedlicher Scheduler Einstellungen	98
47. NB-IoT: Latenzerhöhung durch Sendewiederholungen	100
48. NB-IoT: Vergleich unterschiedlicher RAI Konfigurationen	102
49. NB-IoT: Einfluss des Sendezustands auf die Latenz	103
50. NB-IoT: Vergleich unterschiedlicher Teilnehmerzahlen	104
51. NB-IoT: Einfluss des Sendeintervalls und des <code>inactivity timers</code> auf die Lebensdauer	107
52. LTE: Latenz für unterschiedliche Paketgrößen	109
53. LTE: Zusammenhang zwischen TBS, MCS und Bandbreite	110
54. LTE: Zusammenhang zwischen MCS und Distanz	111
55. LoRa: Variation der Paketgröße	113
56. LoRa: Energiebedarf je Byte und Paketgröße	114
57. LoRa: Variation der Sendeleistung	115
58. LoRa: Variation des Spreizfaktors	116
59. LoRa: Energieeinsparung nach Spreizfaktor	117
60. LoRa: Lebensdauer vs. Leistung vs. SF	119
61. MIOTY: Variation der Paketgröße	120
62. MIOTY: Energiebedarf je Byte für unterschiedliche Paketgrößen	121
63. MIOTY: Variation der Sendeleistung	121
64. MIOTY: Vergleich der Sendemodi bei 32 B Uplink Daten	123
65. MIOTY: Vergleich der Sendemodi	124
66. MIOTY: Lebensdauer vs. Sendeleistung vs. Sendemodus	125
67. Übersicht zum Einfluss der variierten Parameter auf die KPIs	127
68. Anwendungsfallspezifische Profile	128
69. Erfüllbarkeit der Anwendungsfälle	129

F. Tabellenverzeichnis

1.	Zusammenhang zwischen Spreizfaktor, Datenrate, Übertragungsdauer, Duty-Cycle [22, 22]	8
2.	Übersicht wichtiger Mioty-Parameter [23]	10
3.	Theoretischer Vergleich von Leistungsparameter für unterschiedliche drahtlose Technologien	15
4.	Übersicht zu relevanten Kommunikationsmustern	23
5.	Charakterisierung von FA1	25
6.	Charakterisierung von FA2	26
7.	Charakterisierung von PA1	27
8.	Charakterisierung von PA2	28
9.	Charakterisierung von MM1	29
10.	Charakterisierung von MM2	30
11.	Anforderungen der Anwendungsfälle and die Kommunikationslösungen .	32
12.	Anforderungen an das zentrale Testsystem	34
13.	Anforderungen an Feldtests	35
14.	Anforderungen an den Netzwerkemulator	36
15.	Anforderungen an den Simulator	36
16.	Testphasen innerhalb des Projektes	73
17.	Kommunikationsmuster und deren Relevanz für Anwendungsfälle	74
18.	Tabellarisches Testbeschreibungsschema	75
19.	Eingesetzte Hardware	77
20.	Charakterisierung der Feldtestpunkte	79
21.	ETU Profil	80
22.	Vergleich der Einstellung der Umgebungsbedingungen in unterschiedlichen Abstraktionsebenen	84
23.	Parametervariation für NB-IoT	85
24.	Parametervariation für LTE	86
25.	Parametervariation für LoRa	86
26.	Parametervariation für MIOTY	87
27.	Zusammenfassung der Feldtestergebnisse für Latenz und Paketverlust .	90
28.	Zusammenfassung der Variation der Paketgröße für Latenz und Paketverlust	93
29.	NB-IoT: Energie je Tx-Vorgang für unterschiedliche Anzahl von HARQ-Prozessen	97
30.	NB-IoT: Energie je Tx-Vorgang für unterschiedliche Scheduler-Konfigurationen	99
31.	NB-IoT: Energie je Tx-Vorgang für unterschiedliche Anzahl an Sendewiederholungen	100
32.	NB-IoT: Sensitivitätsgrenze in Abhängigkeit der Sendewiederholungen . .	101
33.	NB-IoT: Energie je Tx-Vorgang für RAI Konfigurationen	102
34.	NB-IoT: Energie je Tx-Vorgang aus unterschiedlichen Zuständen	103
35.	NB-IoT: Ermittelte Energiedaten für unterschiedliche Operationen	105
36.	LTE: Latenzanstieg und Paketverlust bei Reduktion der Bandbreite	109
37.	LTE: Durchsatz für unterschiedliche Bandbreiten im UL	110

38. LTE: Durchsatz für unterschiedliche Modulationsarten im UL	110
39. LTE: Durchsatz für unterschiedliche Modulationsarten im DL	111
40. LTE: Latenzanstieg und Paketverlust bei Reduktion der Bandbreite	112
41. LoRa: Energiebedarf für den TX-Zustand unterschiedlicher Paketgrößen	114
42. LoRa: Sensitivitätslimit in Abhängigkeit des Spreizfaktors	115
43. LoRa: Ströme und aktive Zeiten für unterschiedliche Spreizfaktoren	117
44. MIOTY: Sensitivitätslimit in Abhängigkeit der Sendemodi	123
45. Durchgeführte Transfermaßnahmen	139
46. Geplante Transfermaßnahmen (nach Projektende)	140

G. Literatur

- [1] "Beecham research." [Online]. Available: <http://www.beechamresearch.com/downloadFile.aspx?name=Beecham-Sector-Map.ppt>
- [2] "Lora alliance®." [Online]. Available: <https://www.lora-alliance.org/>
- [3] "Sigfox 0g technology." [Online]. Available: <http://www.sigfox.com/en/>
- [4] "mioty® – drahtlose lpwan technologie." [Online]. Available: <https://www.iis.fraunhofer.de/de/ff/lv/net/telemetrie.html>
- [5] "mioty alliance - mioty alliance." [Online]. Available: <https://mioty-alliance.com/>
- [6] "3gpp – the mobile broadband standard." [Online]. Available: <http://www.3gpp.org/>
- [7] "LTE-M – optimizing LTE for the internet of things," 2015. [Online]. Available: http://networks.nokia.com/sites/default/files/document/nokia_lte-m_-_optimizing_lte_for_the_internet_of_things_white_paper.pdf
- [8] M. Bremmer, "Telekom bringt erste narrowband-iot-servicepakete auf dem markt," Jun 2017. [Online]. Available: <https://www.computerwoche.de/a/telekom-bringt-erste-narrowband-iot-servicepakete-auf-dem-markt,3331042>
- [9] commsolid - connecting.amazing.things. [Online]. Available: <https://commsolid.com/>
- [10] sentec elektronik. [Online]. Available: <https://www.sentec-elektronik.de/>
- [11] ublox. [Online]. Available: <https://www.u-blox.com/en/>
- [12] Stackforce - embedded.connectivity.solutions | iot-konnektivitätslösungen. [Online]. Available: <https://stackforce.com/de>
- [13] M. Gidlund, T. Lennvall, and J. Åkerberg, "Will 5G become yet another wireless technology for industrial automation?" in *2017 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1319–1324.

- [14] J. Cheng, W. Chen, F. Tao, and C.-L. Lin, "Industrial IoT in 5G environment towards smart manufacturing," vol. 10, pp. 10–19. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2452414X18300049>
- [15] P. Varga, J. Peto, A. Franko, D. Balla, D. Haja, F. Janky, G. Soos, D. Ficzere, M. Maliosz, and L. Toka, "5G support for Industrial IoT Applications— Challenges, Solutions, and Research gaps," vol. 20, no. 3, p. 828. [Online]. Available: <https://www.mdpi.com/1424-8220/20/3/828>
- [16] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, 2017. [Online]. Available: <https://doi.org/10.1109/comst.2017.2652320>
- [17] E. Commission. Commission Implementing Decision (EU) 2017/1483 of 8 August 2017 amending Decision 2006/771/EC on harmonisation of the radio spectrum for use by short-range devices and repealing Decision 2006/804/EC. [Online]. Available: <https://eur-lex.europa.eu/legal-content/GA/TXT/?uri=CELEX%3A32017D1483>
- [18] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," vol. 55, no. 9, pp. 34–40.
- [19] LoRa Data rate Calculator | LoRa Bit rate Calculator. [Online]. Available: <https://www.rfwireless-world.com/calculators/LoRa-Data-Rate-Calculator.html>
- [20] E. B. LoRa — LoRa documentation. [Online]. Available: <https://lora.readthedocs.io/en/latest/>
- [21] K. Mikhaylov, "Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology," in *European Wireless 2016; 22th European Wireless Conference*, pp. 1–6.
- [22] Understanding ADR | DEVELOPER PORTAL. [Online]. Available: <https://lora-developers.semtech.com/documentation/tech-papers-and-guides/understanding-adr/>
- [23] ETSI, "Etsi ts 103 357 v1.1.1 (2018-06) short range devices; low throughput networks (ltn);protocols for radio interface a," ETSI, Tech. Rep., June 2018.
- [24] Funksensornetzwerke: Rezept für mioty-fähige sensoren - software - elektroniknet. [Online]. Available: <https://www.elektroniknet.de/embedded/software/rezept-fuer-mioty-faehige-sensoren.184502.html>
- [25] F. I. F. I. C. IIS, *MIOTYTM – MAC AND HIGHER LAYERS*. [Online]. Available: https://www.iis.fraunhofer.de/content/dam/iis/de/img/Forschungsfelder/LV/VernetzungundIdentifikation/Technologien/mioty/Mioty-Mac_and_Higher_Layers_final.pdf

- [26] Bundesnetzagentur - mobiles breitband. [Online]. Available: <https://www.bundesnetzagentur.de/DE/Fachthemen/Telekommunikation/Breitband/MobilesBreitband/start.html>
- [27] Bundesnetzagentur - campusnetze. [Online]. Available: <https://www.bundesnetzagentur.de/DE/Fachthemen/Telekommunikation/Nummerierung/Campusnetze/start.html>
- [28] C. Cox, *An Introduction to LTE: LTE, LTE-Advanced, SAE, VoLTE and 4G Mobile Communications, 2nd Edition*. [Online]. Available: <https://learning.oreilly.com/library/view/an-introduction-to/9781118818015/>
- [29] 3GPP, “3gpp ts 22.261 – service requirement for the 5g system ((3gpp ts 22.261 version 16.14.0 release 16)),” 12 2015. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3107>
- [30] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology* -. Amsterdam, Boston: Academic Press, 2018.
- [31] G. A. for Connected Industries and Automation, “5g for industrial internet of things (iiot): Capabilities, features, and potential,” 11 2021. [Online]. Available: https://5g-acia.org/wp-content/uploads/Whitepaper_5G-ACIA_5G-for-Industrial-Internet-of-Things.pdf
- [32] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, “A Primer on 3GPP Narrowband Internet of Things,” vol. 55, no. 3, pp. 117–123.
- [33] M. Chen, Y. Miao, Y. Hao, and K. Hwang, “Narrow Band Internet of Things,” vol. 5, pp. 20 557–20 577.
- [34] M. Lauridsen, R. Krigslund, M. Rohr, and G. Madueno, “An Empirical NB-IoT Power Consumption Model for Battery Lifetime Estimation,” in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pp. 1–5.
- [35] B. Martinez, F. Adelantado, A. Bartoli, and X. Vilajosana, “Exploring the Performance Boundaries of NB-IoT,” vol. 6, no. 3, pp. 5702–5712.
- [36] T. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, “A survey of various propagation models for mobile communication,” vol. 45, no. 3, pp. 51–82.
- [37] A. Sikora, E. J. Sebastian, A. Yushev, E. Schmitt, and M. Schappacher, “Automated Physical Testbeds for Emulation of Wireless Networks,” vol. 75, p. 06006. [Online]. Available: https://www.matec-conferences.org/articles/mateconf/abs/2016/38/mateconf_icmie2016_06006/mateconf_icmie2016_06006.html
- [38] nsnam. Ns-3. ns-3. [Online]. Available: <https://www.nsnam.org/>
- [39] OMNeT++ Discrete Event Simulator. [Online]. Available: <https://omnetpp.org/>

- [40] Riverbed modeler | discrete event simulator for network simulation | riverbed. [Online]. Available: <https://www.riverbed.com/de/products/riverbed-modeler>
- [41] J. Horneber and A. Hergenröder, “A Survey on Testbeds and Experimentation Environments for Wireless Sensor Networks,” vol. 16, no. 4, pp. 1820–1838.
- [42] J. Wu, W. Jiang, Y. Mei, Y. Zhou, and T. Wang, “A Survey on the Progress of Testing Techniques and Methods for Wireless Sensor Networks,” vol. 7, pp. 4302–4316.
- [43] H. Hellbrück, M. Pagel, A. Krölller, D. Bimschas, D. Pfisterer, and S. Fischer, “Using and operating wireless sensor network testbeds with WISEBED,” pp. 171–178.
- [44] OneLab - Home. [Online]. Available: <https://www.onelab.eu/>
- [45] FIT IoT-LAB. FIT IoT-LAB. [Online]. Available: <https://iot-lab.github.io/>
- [46] Testbeds. [Online]. Available: <https://www.smartsantander.eu/index.php/testbeds>
- [47] ETSI. Testing and Test Control Notation Version 3 (TTCN-3). TTCN-3. [Online]. Available: <http://www.ttcn-3.org/>
- [48] Eclipse Titan 8.2.0. GitLab. [Online]. Available: <https://gitlab.eclipse.org/eclipse/titan/titan.core/-/tree/8.2.0>
- [49] Test automation platform - ttworbench - spirent. [Online]. Available: <https://www.spirent.com/products/test-automation-platform-ttworbench>
- [50] *VDI/VDE 2185 Blatt 4 - Funkgestützte Kommunikation in der Automatisierungstechnik - Messtechnische Performancebewertung von Funklösungen für industrielle Automatisierungsanwendungen*, Mar. 2019.
- [51] *VDI/VDE 2185 Blatt 1 - Funkgestützte Kommunikation in der Automatisierungstechnik - Anforderungen und Grundlagen*, Aug. 2020.
- [52] Taktilus – Taktiler Internet für sichere und zeitsensitive Anwendungen der Industrie- und Prozessautomation. [Online]. Available: <https://taktilus-projekt.de/>
- [53] Nb-iot. [Online]. Available: <https://www.nsnam.org/wiki/NB-IOT>
- [54] A. K. Sultania, C. Delgado, and J. Famaey, “Implementation of NB-IoT Power Saving Schemes in ns-3,” in *Proceedings of the 2019 Workshop on Next-Generation Wireless with Ns-3*. ACM, pp. 5–8. [Online]. Available: <https://dl.acm.org/doi/10.1145/3337941.3337944>
- [55] P. Jörke, T. Gebauer, and C. Wietfeld, “From LENA to LENA-NB: Implementation and Performance Evaluation of NB-IoT and Early Data Transmission in ns-3,” in *Proceedings of the 2022 Workshop on Ns-3*, ser. WNS3 '22. Association for Computing Machinery, pp. 73–80. [Online]. Available: <https://doi.org/10.1145/3532577.3532600>

- [56] P. Jörke, T. Gebauer, S. Böcker, and C. Wietfeld, "Scaling Dense NB-IoT Networks to the Max: Performance Benefits of Early Data Transmission," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–7.
- [57] TTCN-3 Standards. [Online]. Available: <http://www.ttcn-3.org/index.php/downloads/standards>
- [58] Eclipse Projects / Eclipse Titan / titan.TestPorts.HTTPmsg · GitLab. GitLab. [Online]. Available: <https://gitlab.eclipse.org/eclipse/titan/titan.TestPorts.HTTPmsg>
- [59] Eclipse Projects / Eclipse Titan / titan.TestPorts.Serial · GitLab. GitLab. [Online]. Available: <https://gitlab.eclipse.org/eclipse/titan/titan.TestPorts.Serial>
- [60] Eclipse Projects / Eclipse Titan / titan.TestPorts.TCPasp · GitLab. GitLab. [Online]. Available: <https://gitlab.eclipse.org/eclipse/titan/titan.TestPorts.TCPasp>
- [61] Eclipse Projects / Eclipse Titan / titan.TestPorts.UDPasp · GitLab. GitLab. [Online]. Available: <https://gitlab.eclipse.org/eclipse/titan/titan.TestPorts.UDPasp>
- [62] B. Mishra and A. Kertesz, "The Use of MQTT in M2M and IoT Systems: A Survey," vol. 8, pp. 201 071–201 086.
- [63] Lightweight machine to machine technical specification. [Online]. Available: https://www.openmobilealliance.org/release/LightweightM2M/V1_1_1-20190617-A
- [64] V. Abramova and J. Bernardino, "NoSQL databases: MongoDB vs cassandra," in *Proceedings of the International C* Conference on Computer Science and Software Engineering*, ser. C3S2E '13. Association for Computing Machinery, pp. 14–22. [Online]. Available: <https://doi.org/10.1145/2494444.2494447>
- [65] Power profiler kit ii. [Online]. Available: <https://www.nordicsemi.com/Products/Development-hardware/Power-Profiler-Kit-2>
- [66] "ETSI TS 136 104 V14.3.0 (2017-04) LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception (3GPP TS 36.104 version 14.3.0 Release 14)." [Online]. Available: https://www.etsi.org/deliver/etsi_ts/136100_136199/136104/14.03.00_60/
- [67] M. Hata, "Empirical formula for propagation loss in land mobile radio services," vol. 29, no. 3, pp. 317–325.
- [68] Fading_urban_3kmph.png (PNG-Grafik, 583 × 438 Pixel). [Online]. Available: https://www.nsnam.org/docs/models/html/_images/fading_urban_3kmph.png
- [69] 3GPP, "3gpp tr 45.820 – cellular system support for ultra-low complexity and low throughput internet of things (ciot)," 12 2015. [Online]. Available: <https://itectec.com/archive/3gpp-specification-tr-45-820/>

- [70] A. P. Matz, J.-A. Fernandez-Prieto, J. Cañada-Bago, and U. Birkel, "A Systematic Analysis of Narrowband IoT Quality of Service," vol. 20, no. 6, p. 1636. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7146203/>
- [71] iperf - iperf3 and iperf3 user documentation. [Online]. Available: <https://iperf.fr/iperf-doc.php>
- [72] Lorawan airtime calculator. [Online]. Available: <https://www.thethingsnetwork.org/airtime-calculator>
- [73] Airtime calculator for lorawan. [Online]. Available: <https://avbentem.github.io/airtime-calculator/ttn/eu868/32>
- [74] J. E. Sebastian and A. Sikora, "Systematic test environment for narrowband IoT technologies," in *Technologien für die intelligente Automation*. Springer Berlin Heidelberg, 2022, pp. 233–244. [Online]. Available: https://doi.org/10.1007/978-3-662-64283-2_17
- [75] J. A. Sikora, F. Sowieja, M. Schappacher, and W. Hadrich, "Extended automated physical testbed for wireless iot testing & performance measurements," in *IEEE International Conference on Industrial Informatics, INDIN'23 (17th - 20th July 2023, Lemgo, Germany, 2023)*.