

BACHELORTHESIS
Frank Matthiesen

Evaluation einer Audio- und Videobasierten M2M Kommunikation

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Frank Matthiesen

Evaluation einer Audio- und Videobasierten M2M Kommunikation

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Technische Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Martin Becke
Zweitgutachter: Prof. Dr. Andreas Meisel

Eingereicht am: 27. August 2021

Frank Matthiesen

Thema der Arbeit

Evaluation einer Audio- und Videobasierten M2M Kommunikation

Stichworte

Evaluation, M2M, Audio, Video, Kommunikation, schnurlos, QR-Code, Frequenzumtastung

Kurzzusammenfassung

Sollen zwei Systeme kommunizieren, kann das stationär oder mobil geschehen. Stationär ist es oft möglich mit einer festen Verkabelung eine stabile und schnelle Kommunikation zu implementieren, hingegen muss mobil auf andere Techniken gesetzt werden. Mögliche Techniken sind hierbei die Übertragung mittels Radiowellen, Schallwellen oder über optische Systeme.

Die Übertragung mittels Radiowellen ist bereits durch viele Standards, wie bspw. WiFi oder Bluetooth, umgesetzt und in vielen Geräten implementiert. Tritt ein Stromausfall in der Infrastruktur auf, dann kann nicht mehr darüber kommuniziert werden und es muss eine direkte Kommunikation stattfinden. Für diese direkte Verbindung ist aber ein sogenannter Handshake notwendig um Informationen für Kommunikation auszutauschen. Ein weiteres Szenario kann eine Störung der Radiospektren sein, wodurch gar keine Kommunikation via Radiowellen möglich ist.

Aus diesem Grund wird in dieser Arbeit die Stabilität der Übertragung mittels Schallwellen und optischen Systemen betrachtet. Gemessen wird diese bei der Übertragung mittels Schallwellen anhand der auftretenden Bitfehler, bei optischen Systemen anhand fehlerhafter Frames. Ziel soll es sein, einen Überblick zu geben, mit welchen Verlusten bei der Übertragung zu rechnen ist, damit auf diesen Erkenntnissen ein entsprechend stabiles Protokoll entwickelt werden kann.

Für die audiobasierte Übertragung wurde eine offene Bibliothek genutzt die auf Frequenzumwandlung setzt. Die videobasierte Übertragung wurde mithilfe von QR-Codes gelöst. Die Ergebnisse waren sehr heterogen mit geringen bis sehr hohen Verlusten bei der audiobasierten Übertragung. Bei der videobasierten Übertragung kam es zu geringeren Verlusten.

Frank Matthiesen

Title of Thesis

Evaluation of an audio and video based M2M communication

Keywords

Evaluation, M2M, Audio, Video, Communication, Wireless, QR-Code, Frequency-shift keying

Abstract

If two systems are to communicate, this can be done stationary or mobile. In stationary systems, it is often possible to implement stable and fast communication with fixed cabling, whereas mobile systems must use other techniques. Possible techniques are transmission via radio waves, sound waves or optical systems. Transmission via radio waves has already been implemented in many standards, such as WiFi or Bluetooth, and in many devices. If a power failure occurs in the infrastructure, it is no longer possible to communicate via it and direct communication must take place. For this direct connection, a so-called handshake is necessary to exchange information for communication. Another scenario can be a disturbance of the radio spectrum, whereby no communication via radio waves is possible at all. For this reason, the stability of transmission via sound waves and optical systems is considered in this work. In the case of transmission via sound waves, this is measured on the basis of the bit errors that occur, in the case of optical systems on the basis of faulty frames. The aim is to give an overview of the losses to be expected during transmission, so that a correspondingly stable protocol can be developed on the basis of this knowledge. For the audio-based transmission, an open library was used that relies on frequency conversion. The video-based transmission was solved with the help of QR codes. The results were very heterogeneous with low to very high losses for audio-based transmission. The video-based transmission had lower losses.

Inhaltsverzeichnis

Abbildungsverzeichnis	viii
1 Einleitung	1
1.1 Motivation	1
1.2 Aufbau der Arbeit	2
2 Grundlagen	4
2.1 Kommunikation von Systemen	4
2.1.1 Kabelgebunden	4
2.1.2 Kabellos	5
2.1.3 Protokolleigenschaften	6
2.1.4 Dienste	7
2.2 Probleme	9
2.2.1 Stromausfall	9
2.2.2 Frequenzstörung	10
3 Derzeitiger Entwicklungsstand	11
3.1 Audio	11
3.1.1 Frequenzumtastung	11
3.1.2 Schnelle Fourier-Transformation	13
3.2 Video	15
3.2.1 QR-Code	15
4 Konzept	18
4.1 Designkriterien	18
4.1.1 Experimentablauf	18
4.1.2 Anforderungen an den Traffic Generator	19
4.1.3 Anforderungen an das Protokoll	20
4.1.4 Anforderungen an den Testaufbau	20

4.2	Verwendete Techniken	21
4.2.1	Software Bibliotheken	21
4.2.2	Protokoll	22
4.2.3	Parameter	24
4.2.4	User Interface des Traffic Generators	25
4.2.5	Daten sichern	26
4.3	Mögliche Fehlerquellen	28
4.3.1	Data-Over-Sound	29
4.3.2	QR-Codes	29
5	Umsetzung	30
5.1	Software	30
5.1.1	Android MVC	30
5.1.2	Configuration Object	31
5.1.3	Object Passing in Android	32
5.1.4	Sending Activity Architektur	33
5.1.5	Speicherform der Daten	35
5.2	Experimentaufbau	36
5.2.1	Smartphones	36
5.2.2	Experimentaufbau	36
5.2.3	Parameter	39
6	Ergebnisse	40
6.1	Parameterauswahl	40
6.2	Erwartungen	40
6.2.1	Data-over-Sound Übertragung	41
6.2.2	QR-Codes Übertragung	41
6.3	Aufbereitung der Daten	42
6.4	Datenübertragung mit Data-over-Sound	43
6.4.1	Beobachtung	43
6.4.2	Diskussion	46
6.5	Datenübertragung mit QR-Codes	47
6.5.1	Beobachtung	47
6.5.2	Diskussion	51
7	Fazit	53
7.1	Ausblick und Verbesserungen	54

Literaturverzeichnis	55
Selbstständigkeitserklärung	58

Abbildungsverzeichnis

2.1	Beispiele für mögliche Ethernet-Topologien	5
2.2	WiFi Scan in der Staatsbibliothek Hamburg	6
2.3	Beispielhaft das OSI Schichtenmodell	9
3.1	Bildung eines binären Frequenzumtastung-Signals. Quelle: [16]	12
3.2	Ein zeitkontinuierliches Signal und dessen Abbildung auf ein Frequenzspektrum	13
3.3	Ein QR-Code mit markierten Elementen seines Aufbaus	16
4.1	Wireframe der zu entwickelnden Anwendung	25
4.2	Ablauf einer Übertragung	27
4.3	Komponenten der Übertragung	28
5.1	Aufbau der Sending Activity	33
5.2	Sequenz einer Übertragung mittels QR-Codes	34
5.3	Sequenz einer Übertragung mittels Data-over-Sound	35
5.4	Testaufbau mit beiden Smartphones bei der Übertragung mittels QR-Codes	37
5.5	Spektrometer kurz nach einer Übertragung	38
6.1	Bitfehler über die verschiedenen Konfigurationen	43
6.2	Verschiedene Audiospektren bei 15 cm Abstand	44
6.3	Verschiedene Audiospektren bei 100 cm Abstand	44
6.4	Verschiedene Audiospektren bei 150 cm Abstand	45
6.5	Verschiedene Audiospektren bei 300 cm Abstand	45
6.6	Frameverluste über die verschiedenen Konfigurationen	47
6.7	Verschiedene Geschwindigkeiten bei 15 cm Abstand und maximal 25 Zeichen pro QR-Code	48
6.8	Verschiedene Geschwindigkeiten bei 15 cm Abstand und maximal 40 Zeichen pro QR-Code	49

6.9	Verschiedene Geschwindigkeiten bei 25 cm Abstand und maximal 25 Zeichen pro QR-Code	50
6.10	Verschiedene Geschwindigkeiten bei 25 cm Abstand und maximal 40 Zeichen pro QR-Code	51

1 Einleitung

1.1 Motivation

Übliche kabelgebundene oder kabellose Kommunikationsstrecken sind in der Informatik weit verbreitet, sind aber nicht immer zugänglich oder stabil. Daher lohnt es sich durchaus über alternative Kommunikationsarten zu diskutieren. Alternative Kommunikationsarten können zum Beispiel mit Audio- und Videoübertragung benannt werden. Diese Übertragungsarten sollten über verschiedene Eigenschaften verfügen zum Beispiel Robustheit und den Aufbau stabiler Verbindungen.

Kommunikation ist ein Teilgebiet der verteilten Systeme wie es in Tannenbaum [17] diskutiert wird. Natürlich spielen auch andere Bereiche eine Rolle, wie die Architektur oder verteilte Algorithmik, aber in dieser Arbeit wird sich auf eine reine Ende-zu-Ende-Kommunikation konzentriert. Wobei man sich stark an der Nomenklatur der Referenzliteratur orientiert.

Die Referenzliteratur kennt in der Kommunikation verschiedene Eigenschaften, wie zum Beispiel verbindungslose und verbindungsorientierte Kommunikation oder transienten und persistenten Datenaustausch. Weiter wird auch in den Referenzwerken diskutiert über verschiedene Dienste, wie zum Beispiel Multiplexing, Datenintegrität, vollständige Übertragung, Reihenfolge Sicherung und Verschlüsselung. Natürlich müssen in dieser Arbeit diese Punkte auch adressiert werden.

Diese Arbeit konzentriert sich zunächst auf die Ende-zu-Ende-Kommunikation zweier Maschinen in einer geografischen Nähe. Die Maschinen sind definiert als Endgeräte mit einem Display, Lautsprecher, Mikrofon und einer Kamera. Für diese Art der Maschinen gibt es noch keine standardisierte Art der Kommunikation. Ein starkes Motiv dieser Arbeit ist es eine Lösung dafür anzubieten.

Die wesentliche Herausforderung der Arbeit ist es ein Protokoll zu spezifizieren und an die Anforderungen anzupassen. Um den Funktionsumfang des Protokolls zu diskutieren, soll nachrangig ein Experimentaufbau zur Evaluation der Ergebnisse erstellt werden. Hierzu ist es zwingend notwendig Werkzeuge wie einen Traffic Generator bereitzustellen, welcher die zu messenden Übertragungen generiert. Ziel ist es mit den alternativen Kommunikationswegen einen hohen Durchsatz zu ermöglichen, daher liegt eine weitere Motivation dieser Arbeit darin, Optimierungsvorschläge zu anzubieten. Des Weiteren wird sich auf Unicast Kommunikation konzentriert, ohne Multicast oder Broadcast Kommunikation zu betrachten. Zusammengefasst ist die Hauptmotivation eine effiziente Alternative zu dem üblichen Ethernet Stack anzubieten und zu evaluieren.

Für den Datenaustausch mit Audiosignalen wird Frequenzumtastung angewendet. Dafür wird eine definierte Anzahl an Bits als eine Sequenz übermittelt. Danach wird der Datenverlust dieser Bits betrachtet.

Bei der videobasierten Übertragung werden Datensätze mithilfe von QR-Codes ausgetauscht. Da jeder QR-Code einen Frame des Datensatzes darstellt, wird im Anschluss der Verlust dieser Frames in verschiedenen Szenarien betrachtet.

1.2 Aufbau der Arbeit

Die Arbeit besteht aus folgenden Kapiteln:

Einleitung

Grundlagen

Es wird betrachtet wie Computer miteinander kommunizieren. Welche Ansätze es bei kabelgebundener sowie schnurloser Kommunikation gibt und welche Probleme dabei auftreten können.

Derzeitiger Entwicklungsstand

Hier wird betrachtet was für Ansätze es bereits im Bereich der Übertragung auf audio- und videobasierten Kommunikationen gibt.

Konzept

Es werden die Kriterien, verwendeten Techniken und möglichen Fehlerquellen des gewählten Konzepts vorgestellt.

Umsetzung

Es wird gezeigt mit welcher Hardware und in welchem Versuchsaufbau die Messergebnisse erhoben werden

Ergebnisse der Evaluation

Die Ergebnisse der Umsetzung werden betrachtet und bewertet.

Fazit und Ausblick

Die Ergebnisse werden zusammengefasst und ein Ausblick gegeben wie diese Arbeit weiter geführt werden kann.

2 Grundlagen

Zunächst wird ein Überblick davon gegeben mithilfe welcher Techniken verteilte Systeme und Maschinen miteinander kommunizieren. Danach werden die Anforderungen beschrieben, welche an eine Kommunikation gestellt werden. Zuletzt wird grob angerissen was für Probleme bei der Kommunikation auftreten können.

2.1 Kommunikation von Systemen

Wenn Maschinen miteinander verbunden werden sollen, stellt sich die Frage, ob diese miteinander verkabelt werden sollen oder kabellos miteinander kommunizieren. Beide Arten haben sowohl Vor- als auch Nachteile, weswegen keine generelle Aussage darüber getroffen werden kann welcher Typ besser geeignet ist. Kabelgebundene Verbindungen können einen höheren Datendurchsatz und eine stabilere Verbindung ermöglichen, jedoch kann das Verlegen der Kabel unter Umständen kostspielig oder gar nicht möglich sein. Kabellose Verbindungen sind bei manchen Technologien flexibel, da nur eine Stromversorgung der Maschine benötigt wird. Dafür benötigen Sie teilweise extra Infrastruktur. Bei WiFi wird beispielsweise meist mithilfe von Access Points kommuniziert, welche für den Betrieb Strom und oft auch einen Netzwerkzugang benötigen. Möglicherweise ist die zentrale Anforderung an eine Maschine, dass sie mobil sein soll, sodass die Einschränkung durch Kabel inakzeptabel ist. Es folgen Beispiele für beide Techniken wie sie geläufig genutzt werden.

2.1.1 Kabelgebunden

Ethernet

Der Standard IEEE 802.3 und dessen Unterpunkte stellt die Hauptbasis für Netzwerke seit den 80ern dar. Es sind Übertragungsraten von bis zu 10 Gbit möglich (IEEE 802.3ae),

Maschinen können mit Strom versorgt werden (IEEE 802.af und IEEE 802.at) und Duplex Verbindungen (IEEE 802.3x) sind möglich. Verbunden werden die Maschinen bei Ethernet direkt, über einen Bus, über Geräte wie Switches oder vermischt. Es sind verschiedene Netzwerk Topologien möglich, wie in Abb. 2.1 zu sehen, was bei dem Aufbau Flexibilität bietet.

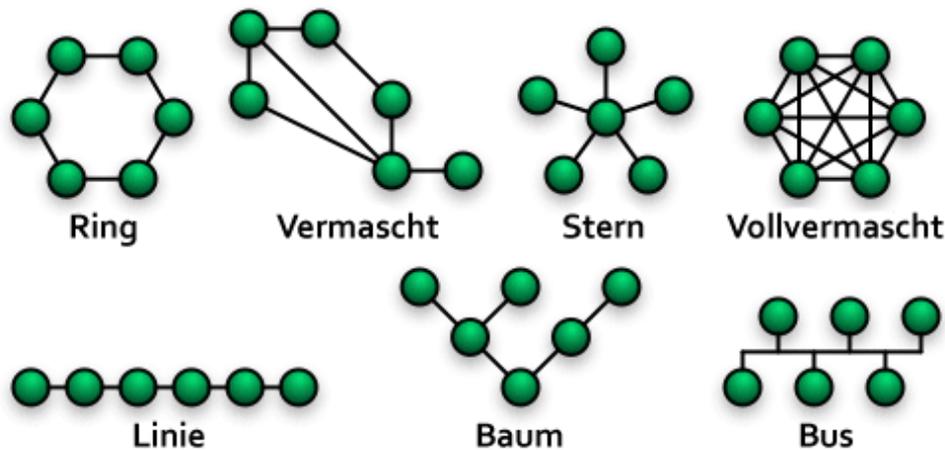


Abbildung 2.1: Beispiele für mögliche Ethernet-Topologien

2.1.2 Kabellos

WiFi

Seit Ende der 90er Jahre existiert der Standard IEEE 802.11. Anfangs wurden Computer noch mit Erweiterungskarten ausgestattet, um per Funk Daten zu übertragen. Heutzutage besitzen bereits kleinste Mikrocontroller die Fähigkeit nach dem Standard zu senden und zu empfangen.

Topologien wie Direktverbindungen zwischen zwei Geräten sind im Standard vorgesehen. Meist kommunizieren Geräte mittels sogenannter Access Points miteinander.

WiFi sendet für gewöhnlich auf Kanälen im 2,4 GHz und 5 GHz Band. Es sind aber bereits Standards geplant, die im 60 GHz oder 900 MHz Band senden. Ein Frequenzband wird in verschiedene Kanäle aufgeteilt, auf denen die Geräte kommunizieren. Da sich die Maschinen eine Frequenz teilen, sind verschiedene Techniken entwickelt worden, um die Kanäle für verschiedene Sender aufzuteilen. Um die Übertragungen vor unbefugten

Dritten zu sichern wurden Standards wie zum Beispiel WPA entwickelt, die bereits in mehreren Versionen existieren.

Abb. 2.2 zeigt wie das 5 GHz Frequenzband an einem belebten Ort aussehen kann. Zu sehen ist wie sich diverse Access Points die Kanäle teilen.

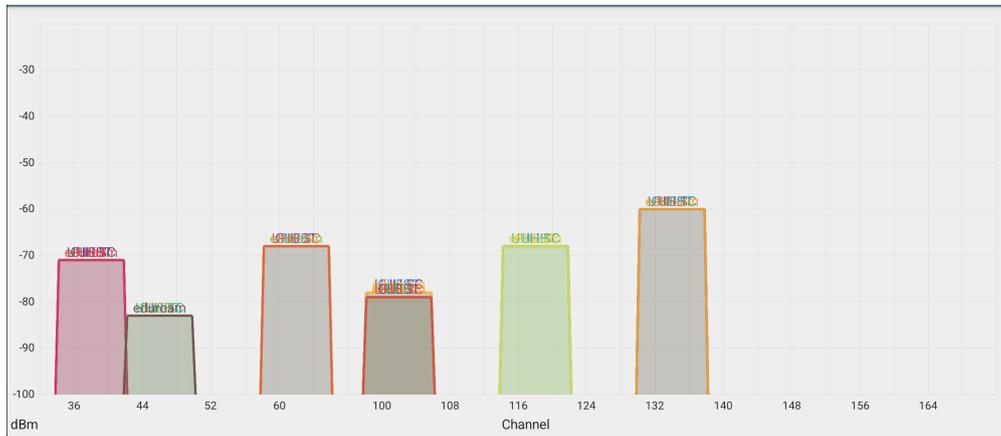


Abbildung 2.2: WiFi Scan in der Staatsbibliothek Hamburg

2.1.3 Protokolleigenschaften

Verbindungslos / Verbindungsorientiert

Tanenbaum definiert eine Kommunikation als verbindungslos, wenn der Sendende seine Daten einfach versendet [17] ohne vorher mit dem Empfänger Informationen, über die bevorstehende Übertragung ausgetauscht zu haben.

Bei der verbindungsorientierten Kommunikation wird zunächst eine Verbindung mit dem Empfänger aufgebaut. Durch einen Handshake werden Eigenschaften der folgenden Übertragung mitgeteilt.

Beispiele sind UDP für eine verbindungslose und TCP für eine verbindungsorientierte Kommunikation.

Transient / Persistent

Als Beispiel für ein persistente Kommunikation lässt sich ein E-Mail System sehen. Eine versendete Nachricht wird zuerst auf einer Middleware gespeichert und gelangt nicht

direkt zum Empfänger. Dadurch ergibt sich, dass der Empfänger nicht durchgängig mit einer Nachricht rechnen und diese sofort abarbeiten muss. Das erfordert aber Systeme, welche die zu übermittelnden Informationen speichern können.

Ein transientes System hingegen besitzt keine Middleware wodurch ein Datenaustausch nur möglich ist, wenn Sender und Empfänger sich im entsprechenden Zustand befinden.

2.1.4 Dienste

Multiplexing

Nehmen wir das Beispiel TCP so muss jede Anwendung eine Verbindung eröffnen, um Daten an einen Empfänger zu senden. Das kann durch die verbindungsorientierte Übertragung zu einem Overhead bei der Berechnung und der zu Versenden Nachrichten führen.

Beim Multiplexing teilen sich verschiedene Anwendungen eine Verbindung, in der dann die Daten alle verschickt werden. Das kann besonders bei schwachen Maschinen zu einer ressourcenschonenden Verarbeitung führen.

Datenintegrität

Die Integrität beschreibt die Korrektheit der Daten. Es geht darum, dass die Daten so beim Empfänger ankommen, wie sie beim Sender abgeschickt worden sind. Das kann je nach Anwendung unterschiedlich wichtig sein. Spielt ein Fehler bei einem Videostream vielleicht keine Rolle, so sind falsche Bits bei einem Banküberweisung wahrscheinlich eher ungewünscht.

Reihenfolge Sicherung

In manchen Szenarien können Pakete einer Übertragung vor anderen bei dem Empfänger ankommen. Das Ergebnis ist ein Datenstrom der nicht dem entspricht wie er versendet wurde. Damit die Integrität der Daten nicht verletzt wird, sollte darauf reagiert werden.

Vollständige Übertragung

Damit nach einer erfolgreichen Übertragung alle Daten bei dem Empfänger zu finden sind, sollte dies sichergestellt werden. Beispielsweise sendet TCP bei dem Verbindungsaufbau die Anzahl der folgenden Frames mit, damit die Vollständigkeit gesichert werden kann.

Verschlüsselung

Die Verschlüsselung der Daten hat das Ziel, dass die Daten, die vom Sender verschickt werden, ohne Veränderung beim Empfänger ankommen und auf ihrem Weg auch nicht von Dritten gelesen werden können. Mögliche Veränderungen wären Manipulationen an den Daten, sowohl an einzelnen Bereichen als auch der gesamten Nachricht. Der Bereich Datenschutz und Manipulation ist ein großes Feld und wird in dieser Bachelorarbeit nicht weiter betrachtet.

Schichtenmodell

Im Bereich der verteilten Systeme werden verschiedene Architekturmodelle eingesetzt wie beispielsweise die objektbasierten, die datenzentrierten, die ereignisbasierten oder die geschichteten Architekturen [17].

Bei der geschichteten Architektur werden Komponenten in Schichten zusammengefasst und kommunizieren nur mit der oberhalb oder unterhalb von ihr angrenzenden Schicht. Dadurch kann eine Abstraktion verwendet werden, welche es ermöglicht Anwendungen auf verschiedenen Geräten zu implementieren, weil teilweise nur die Schicht ausgetauscht werden muss, die für die Kommunikation im physikalischen Netzwerk notwendig ist.

Abb. 2.3 zeigt das OSI-Schichtenmodell. TCP sitzt in diesem Model in Schicht vier der Transportschicht und Ethernet in der Sicherung und in der Bitübertragungsschicht.

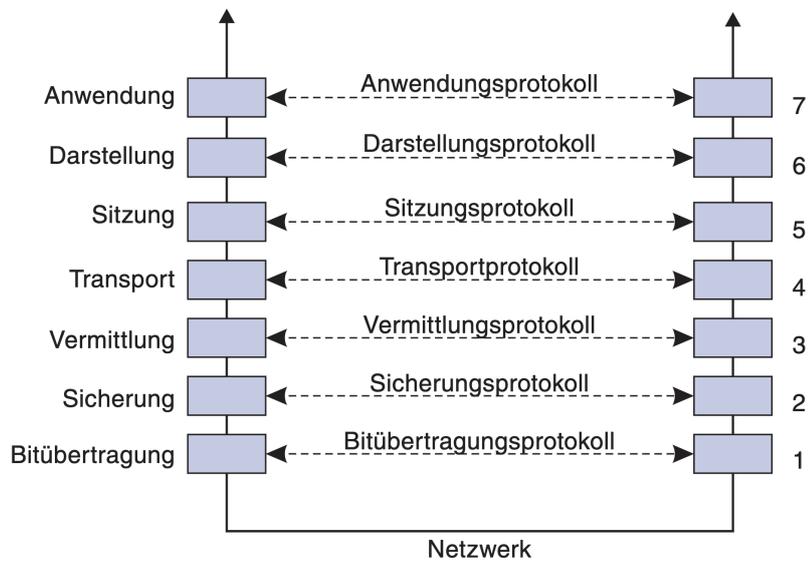


Abbildung 2.3: Beispielhaft das OSI Schichtenmodell

2.2 Probleme

Bei verteilten Systemen muss der Fehlerfall bereits in der Designphase berücksichtigt werden, denn von solchen Systemen wird erwartet, dass sie ein möglichst hohes Maß an Fehlertoleranz besitzen und lange funktionieren [17]. Für einen Ausfall kann es verschiedene Ursachen geben. Die, für die vorliegende Arbeit relevanten Störursachen sind der Stromausfall und die Frequenzstörung.

2.2.1 Stromausfall

Sowohl bei der kabellosen als auch bei der kabelgebundenen Kommunikation ist häufig Infrastruktur für den Datenaustausch notwendig. Wifi und Ethernet ermöglichen zwar die Maschine zu Maschine Kommunikation, aber dies ist oft zu aufwendig. Bei Ethernet müssen Kabel von jeder Maschine zu jeder anderen Maschine gelegt werden und für jede Verbindung eine Netzwerkschnittstelle für jede Verbindung verbaut werden. Es wird meist auf Topologien gesetzt, die auf Netzwerkverteiler (Switch) setzen. Fallen diese Verteiler aus, ist keine Kommunikation mehr möglich.

Die schnurlose Kommunikation via WiFi setzt auf Access Points, die sich ähnlich wie die Verteiler im Ethernet verhalten.

Ist nun die Infrastruktur ausgefallen, so wäre es denkbar eine direkte Funkverbindung zwischen zwei Geräten zu etablieren. Dafür ist aber häufig ein sogenannter Handshake notwendig bei dem die notwendigen Informationen für einen Verbindungsaufbau ausgetauscht werden müssen. Durch die fehlende Infrastruktur ist aber keine Möglichkeit mehr gegeben diese Informationen zu übertragen. Für diese Aushandlung kann eine Alternative zu Ethernet oder WiFi interessant sein.

2.2.2 Frequenzstörung

Funkverbindungen oder auch kabelgebundene Netzwerke, wie der Token-Ring, kommunizieren über ein geteiltes Medium. Verschiedene Maschinen kommunizieren auf gemeinsamen Frequenzen, wodurch es notwendig ist, dass diese unter den Teilnehmern aufgeteilt werden. Für diesen Zweck gibt es verschiedene Multiplextechniken, die das Ziel haben den Frequenzraum möglichst gleichmäßig aufzuteilen. Dabei setzen sie beispielsweise auf zeitbasierte (time-division multiplexing) oder frequenzbasierte (frequency-division multiplexing) Techniken.

Wenn eine Maschine einen Fehler hat oder sich nicht an die Standards hält, ist keine Kommunikation der anderen Teilnehmer mehr möglich. Auch kann es passieren, dass Geräte durch eine Fehlfunktion von ihrer zugewiesenen Funkfrequenz abweichen und die Kommunikation stören.

Eine alternative Art der Kommunikation ist interessant, um beispielsweise unter den Teilnehmern einen neuen Funkkanal auszuhandeln.

3 Derzeitiger Entwicklungsstand

Im folgenden Kapitel soll der aktuelle Forschungsstand betrachtet und dessen Grundlagen beschrieben werden.

3.1 Audio

Den Ansatz Daten über mobile Geräte zu versenden ist ein Themenfeld, welches bereits untersucht wird. Es wurden die mögliche Datenrate anhand der Verluste [3, 6] oder aber der Produktiveinsatz betrachtet. Beispielsweise haben Forschende als alternative Möglichkeit der Lokalisierung innerhalb eines Museums untersucht [4]. Auch wird bereits untersucht, ob die Übertragung von Daten von Lautsprecher zu Mikrofon mithilfe von maschinellem Lernen verbessert werden kann [12].

Diesen Arbeiten ist der Ansatz gemein Daten mittels Frequenzumtastung auf Frequenzen zu übertragen, welche vom menschlichen Gehör nicht wahrgenommen werden können.

3.1.1 Frequenzumtastung

Die Frequenzumtastung ist eine Art der Modulation, bei der digitale Signale über eine Trägerfrequenz übertragen werden [7].

Abb. 3.1 zeigt beispielhaft die Generierung eines Signals. Der Einfachheit halber wird hier ein binäres Datensignal gewählt. Für die möglichen Zustände des Datensignals (High oder Low) werden Frequenzen definiert. Die mittlere Trägerfrequenz steht in diesem Beispiel für ein High im Datensignal. Das untere Signal ist nun das modulierte Ergebnis. Zu sehen sind die beiden verschiedenen Frequenzen je nach Position im Datensignal.

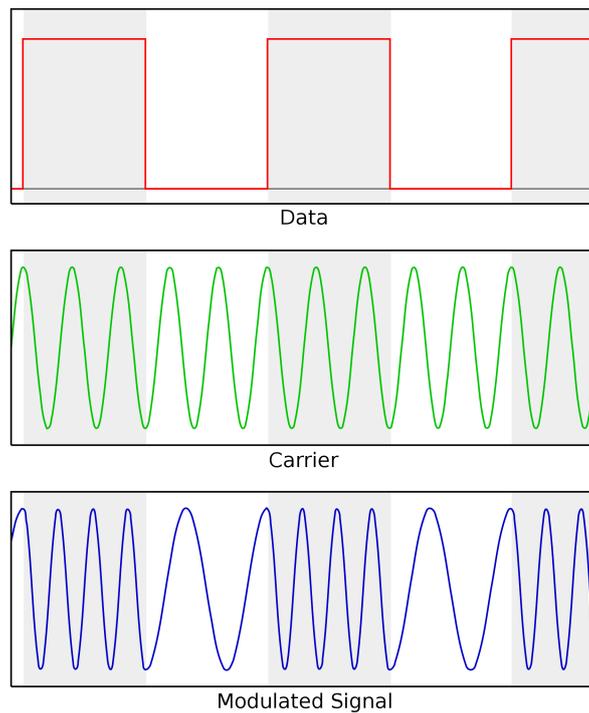


Abbildung 3.1: Bildung eines binären Frequenzumtastungssignals. Quelle: [16]

Es gilt zu definieren, welche Frequenzen für welche Symbole bei der Übertragung stehen. Benötigt wird also für eine Menge an Symbolen S und eine Menge an Frequenzen F . Es gilt: $|S| = |F|$.

Die Menge der Signale bei Abb. 3.1 ist 2 da 1 Bit übertragen werden kann. Sollen nun pro Ton n Bit übertragen werden gilt: $|S| \geq 1 + \sqrt{n}$. Es werden aber auch Steuersignale übertragen, somit kann S größer sein als die notwendige Menge Frequenzen nur für den Datenstrom.

Die Frequenzen werden wie folgt festgelegt:

Wurde eine Startfrequenz f_{start} und eine Endfrequenz f_{ende} definiert, wird anschließend der Abstand von Frequenz zu Frequenz berechnet. Dieser wird in der Literatur manchmal Frequenzhub genannt.

$$f_{\text{hub}} = \frac{|f_{\text{ende}} - f_{\text{start}}|}{|S|}$$

Nun kann von der Startfrequenz f_{start} an für jedes Symbol von S eine Frequenz von F berechnet werden.

Anschließend kann ein moduliertes Signal erzeugt werden und beispielsweise über die Lautsprecher eines Smartphones ausgegeben werden.

3.1.2 Schnelle Fourier-Transformation

Wenn ein Mikrofon eines Smartphones ein moduliertes Signal empfängt, muss dieses wieder demoduliert werden. Eine Möglichkeit dafür ist die Schnelle Fourier-Transformation.

Sie ist ein Algorithmus zur Berechnung einer diskreten Fourier-Transformation, deren Ziel es ist ein zeitdiskretes endliches Signal, das sich periodisch wiederholt, abzubilden auf ein Frequenzspektrum [15].

Abb. 3.2 zeigt wie ein Signal aus verschiedenen einzelnen periodischen Signalen mit verschiedenen Frequenzen aufgebaut sein kann. Dies ist im linken Graphen im zeitlichen Verlauf dargestellt. Rechts daneben ist dasselbe Signal nach Anwendung der Fourier-Transformation. Übereinstimmend zu den Frequenzen und deren Amplituden sind Spitzen im Frequenzraum zu erkennen.

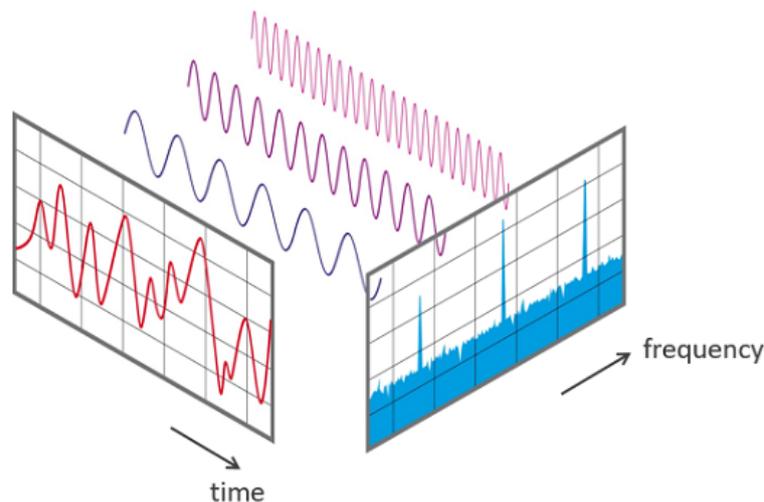


Abbildung 3.2: Ein zeitkontinuierliches Signal und dessen Abbildung auf ein Frequenzspektrum

Die Fourier-Transformation ist definiert als:

$$(\mathcal{F}f)(y) = \frac{1}{\sqrt{2\pi}^n} \int_{\mathbb{R}^n} f(x) e^{-iy \cdot x} dx$$

Mit dem Computer können wir aber keine zeitkontinuierlichen Signale verarbeiten. Aus diesem Grund müssen wir die Eingangssignale zunächst diskretisieren.

Es wird eine Abtastrate f_{abtaast} gewählt wobei gilt, dass diese mindestens doppelt so hoch sein muss wie die höchste zu messende Frequenz [20]. Ist die Abtastrate kleiner, kann es zu Alias-Effekten kommen. Berechnet werden kann die Bandbreite des zu messenden Frequenzspektrums (Nyquist-Frequenz) durch:

$$f_{\text{bandbreite}} = \frac{1}{2} \cdot f_{\text{abtaast}}$$

Zusätzlich benötigt es noch eine Samplingrate SR . Die Frequenzauflösung gibt an welche Abstände von Frequenz zu Frequenz im Ergebnis berechnet werden können. Sie wird berechnet mit:

$$df = f_{\text{abtaast}}/SR$$

Eine höhere Samplingrate bedeutet einen geringeren Abstand und damit eine höhere Frequenzauflösung.

Da aber für die Messdauer gilt:

$$D = SR/f_{\text{abtaast}}$$

benötigt eine höhere Auflösung auch eine längere Messdauer.

Liegen nun die diskreten Werte $a = (a_0, \dots, a_{N-1})$ vor, können diese mit der Diskreten Fourier-Transformation transformiert werden.

$$\hat{a}_k = \sum_{j=0}^{N-1} e^{-2\pi i \cdot \frac{jk}{N}} \cdot a_j$$

Eine effizientere Berechnung bietet die Schnelle Fourier-Transformation. Diese basiert auf den arbeiten von Cooley und Tukey [5].

Die Implementierung von Cooley und Tukey funktioniert nach dem Teile-und-herrsche-Verfahren. Ihnen ist aufgefallen, dass sich eine Diskrete Fourier-Transformation der Größe $2n$ aufteilen lässt in zwei Transformationen der Größe n , die danach wieder zu $2n$ zusammgeführt werden können.

Der Vorteil ist, dass sich durch dieses Verfahren die Komplexität von $\mathcal{O}(n^2)$ zu $\mathcal{O}(n \log n)$ reduziert und der Algorithmus somit schneller abgearbeitet werden kann. Damit dieser Vorteil genutzt werden kann, sollte also eine Samplingrate $SR = 2^n$ wobei gilt $n \in \mathbb{N}$.

3.2 Video

Im Bereich der optischen Übertragung (visible light communication) wird meist über eine Lampe oder LED das zu übertragende Signal ausgesendet [14, 21, 1]. Es gibt aber auch Varianten der Kamera zu Bildschirm Übertragung.

Hierbei steht häufig im Fokus die Daten so dezent wie möglich in die Inhalte, dessen was auf dem Bildschirm, ist zu platzieren [9, 18].

Die Nutzung des Bildschirms als Fläche der Darstellung von komplexen Mustern scheint aber ein Gegenstand großer Diskussion zu sein [19, 11, 2]. Zur Kodierung für Daten werden QR-Codes oder erweiterte Formen ebendieser eingesetzt, beispielsweise mit Farben für mehr Informationen.

3.2.1 QR-Code

Der QR-Code (kurzform für Quick Response Code) wurde 1994 von der Firma Denso entwickelt, da ein Label zur Markierung von Gegenständen entwickelt werden sollte ¹. Er ist ein zweidimensionales Label, welches von Maschinen lesbar ist und Informationen über den Inhalt und weiterführende Informationen über den Gegenstand enthält, an dem es angebracht ist. Inzwischen werden QR-Codes aber nicht mehr nur als Label an Maschinen verwendet. Mit dem Aufkommen von Smartphones werden Sie immer häufiger zur Übertragung von kurzen Datensätzen benutzt, wie zum Beispiel einer URL, einer Visitenkarte oder einer Anmelde Nummer. Im Zuge der Corona-Krise und der Kontaktverfolgung mittels Warn-Apps begegnet man ihnen mehrmals täglich. Die Verwendung liegt auch an der eingebauten Fehlerkorrektur.

¹<https://www.denso-wave.com/en/technology/voll.html>

Aufbau

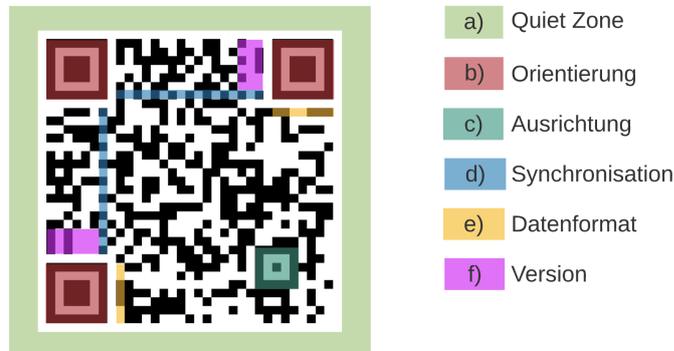


Abbildung 3.3: Ein QR-Code mit markierten Elementen seines Aufbaus

Abb. 3.3 zeigt einen beispielhaften QR-Code mit markierten Bereichen, die keine Nutzdaten enthalten und vorgeschrieben sind.

Das Schachbrettmuster des Codes ist umgeben von einer Quiet Zone (a). Dieser Bereich dient als Pufferzone, um den Kontrast zur Umgebung zu erhöhen. Es wird empfohlen diesen Bereich mindesten 4-mal so breit zu halten wie ein Datenpixel im Datenbereich.

In den oberen Ecken und in der unteren linken Ecke befinden sich die Orientierungsblöcke (b). Sie dienen dazu die Ausrichtung der Kamera auf den Code zu normalisieren. Da eine Kamera selten parallel zum Code gehalten wird, können Winkelunterschiede und Drehungen somit ausgerechnet werden. Sie sind umgeben von einer Pufferzone von der Breite eines Datenpixels zur besseren Erkennung.

Auf dieser Abbildung im unteren rechten Bereich ist der Bereich für die Ausrichtung (c). Wie auch die Orientierungsblöcke soll dieser helfen, die Verzerrungen des Bildes zu erkennen und unterstützt die Algorithmen bei der Decodierung. In Version 1 des Standards war dieser Bereich noch nicht vorhanden und wurde hinzugefügt, um bei zunehmender Datenkapazität das Decodieren zu unterstützen, indem es weitere Merkmale zum Ausgleich von Verzerrungen bietet. Es können mehr als nur ein Block zur Ausrichtung in einem QR-Code enthalten sein.

Der Synchronisationsbereich (d) ist ein Muster aus alternierenden Schwarzen und Weißen Pixeln. Er dient dem Algorithmus dazu, die Breite und Höhe der Datenblöcke zu erfassen.

Das Datenformat (e) ist ein 15-Bit großer Bereich, in den Informationen über den Datenstrom codiert sind. In ihm ist zu finden welche Fehlerkorrekturstufe verwendet wird und mit welchem Datenmuster der Datenbereich zu lesen ist.

Im Bereich der Version (f), ist die verwendete Version zu finden. Primär legt die Version die Auflösung des QR-Codes fest. Beispielsweise hatte Version 1 noch eine Auflösung von 21×21 Pixel, wohingegen Version 40 schon 177×177 Pixel bietet.

Fehlerkorrektur

Ein Vorteil der QR-Codes ist die variabel implementierte Fehlerkorrektur. Der Datensatz nutzt den Reed-Solomon-Code über eine endliche Menge [13]. Auf eine Folge von 8-Bit folgt ein Codewort, mithilfe dessen es möglich ist fehlerhafte Bits zu korrigieren. Jeder Code benutzt ein gewisses Level für die Fehlerkorrektur. Level L (Low) ermöglicht es 7% eines Datensatzes wiederherzustellen. Level M (Medium) 15%, Level Q (Quartile) 25% und Level H (High) 30%.

Datensatz

Abhängig davon was im Bereich für das Datenformat steht, wird die Folge aus Bits entweder als Folge von Ziffern, alphanumerischen Zeichen, Binär nach ISO 8859-1 oder als Kanji decodiert. Je nach Format wird eine unterschiedliche Anzahl an Bits für die Codierung verwendet, wodurch die maximale Kapazität eines QR-Codes je nach Format variiert.

Ein QR-Code in Version 40 mit 177×177 Pixeln mit Fehlerkorrekturstufe Low bietet eine Datenkapazität von 23.624 Bits. Damit sind 7.089 Ziffern, 4.296 alphanumerische Zeichen oder 2.953 Bytes möglich.

4 Konzept

Um mit dieser Arbeit die Übertragungen zu evaluieren, gilt es einen Experiment durchzuführen. Dazu ist es erforderlich ein Traffic Generator, ein Protokoll und einen Versuchsaufbau zu entwerfen. Die Designkriterien und Techniken, welche für diese gelten und genutzt werden sollen, werden im folgenden Kapitel betrachtet.

4.1 Designkriterien

4.1.1 Experimentablauf

Es sollen zwei Smartphones miteinander Daten austauschen. Diese Daten sollen zum einen über einen audiobasierten und zum anderen über einen videobasierten Weg übertragen werden. Dafür ist es notwendig einen Traffic Generator zu schreiben, der auf beiden Wegen Daten versenden und empfangen kann.

Bei der audiobasierten Übertragung sollen die Daten vom Lautsprecher des einen Geräts zum Mikrofon des anderen über eine Frequenzumtastung versendet werden. Bei der videobasierten Übertragung sollen die Daten als QR-Codes auf dem Bildschirm des einen Geräts dargestellt und mithilfe der Kamera des anderen Geräts abgescannt werden. QR-Codes werden bei hoher Datenlast immer höher aufgelöst und somit feiner auf den Bildschirmen der Smartphones. Große Datensätze sollten daher in kleinere aufgeteilt werden.

Pro Testablauf sollen verschiedene Parameter je nach Übertragungsgrundlage variiert werden. Damit ein Testdurchlauf aussagekräftig werden kann, soll jeder Durchlauf wiederholt werden können.

Später sollen die Verluste dieser Übertragung betrachtet werden, dafür müssen die Daten gesichert werden. Allgemein sollen die Experimente dabei mit so wenig äußeren Einflüssen

wie möglich ablaufen. Um Bedienfehler ausschließen zu können, sollen die Testdurchläufe automatisiert durchgeführt werden.

Ursprünglich sollten die Tests mit den Loomo Robotern der Firma Segway durchgeführt werden. Aufgrund der Corona-Krise und der damit begrenzten Verfügbarkeit der Räume der HAW-Hamburg wurde allerdings davon abgesehen und entschieden, die Tests mit Smartphones durchzuführen. Da der Kern der entwickelten Software später auch auf diesen nutzbar sein soll, muss bei der Implementierung darauf geachtet werden, dass von den Loomos unterstützte Android APIs genutzt werden.

4.1.2 Anforderungen an den Traffic Generator

Es ergeben sich die folgenden Anforderungen an den Traffic Generator:

Anforderung 01-001

Er muss auf der Architektur der Smartphones und der Loomo Roboter lauffähig sein. Beide Plattformen basieren auf Android.

Anforderung 01-002

Für die audiobasierte Übertragung müssen Daten mithilfe einer Frequenzumtastung versendet werden können.

Anforderung 01-003

Für die audiobasierte Übertragung müssen Daten mithilfe der schnellen Fourier-Transformation empfangen werden können.

Anforderung 01-004

Für die videobasierte Übertragung müssen QR-Codes erstellt werden können.

Anforderung 01-005

Für die videobasierte Übertragung müssen QR-Codes gelesen werden können.

Anforderung 01-006

Zur späteren Evaluation der Übertragung sollen die versendeten und empfangenen Daten gespeichert werden.

Anforderung 01-007

Die Testdurchläufe müssen wiederholt übertragen werden.

Anforderung 01-008

Testdurchläufe möglichst automatisieren.

Anforderung 01-009

Die Übertragung parametrisieren.

4.1.3 Anforderungen an das Protokoll

Es ergeben sich folgende Anforderungen an das Protokoll:

Anforderung 02-001

Datenübertragungen müssen eindeutig identifizierbar sein.

Anforderung 02-002

Erneute Übertragungen sollen zuordenbar sein.

Anforderung 02-003

Datensätze müssen zerlegt übertragen und wieder zusammgefügt werden können.

4.1.4 Anforderungen an den Testaufbau

Anforderung 03-001

Bei der Video-basierten Übertragung: gleichmäßige Lichtverhältnisse und keine Veränderungen.

Anforderung 03-002

Bei der Video-basierten Übertragung: keine Beleuchtungen die Reflexionen im Display verursachen können.

Anforderung 03-003

Bei der Video-basierten Übertragung: eine neutrale Umgebung ohne Ablenkungen.

Anforderung 03-004

Bei der Audio-basierten Übertragung: keine störenden Geräusche bei den Aufnahmen.

Anforderung 03-005

Bei der Audio-basierten Übertragung: einen Raum der keinen Schall reflektiert.

Anforderung 03-006

Die Geräte dürfen sich während der Tests, nicht bewegen und müssen erschütterungsfest montiert sein.

4.2 Verwendete Techniken

4.2.1 Software Bibliotheken

Bei dieser Arbeit wird auf etablierte Lösungen gesetzt, um die Fähigkeiten dieser zu betrachten. Es werden hier Lösungen, genutzt die unter freier Lizenz auf der Plattform GitHub zu finden sind.

Es wurden Bibliotheken ausgewählt, die in Java geschrieben sind und für die Android-Plattform entwickelt wurden (Anforderung 01-001). Zusätzlich war wichtig, die Übertragung mittels Parameter zu variieren, um einen Einfluss verschiedener Kombinationen zu messen.

Data-over-Sound

Im Bereich der Datenübertragungen von Lautsprechern zu Mikrofon gibt es nur wenige Bibliotheken, die meist nicht frei verfügbar sind. In dieser Arbeit wird die Data-Over-Sound Bibliothek ¹ eingesetzt.

Die Daten werden mittels Frequenzumtastung mittels Schallwellen übertragen (Anforderung 01-002 und 01-003). Sollen Daten empfangen werden, werden die Audiodaten mittels einer schnellen Fourier-Transformation auf ihr Frequenzspektrum abgebildet und ausgewertet.

Der Quellcode ist offen und bietet Möglichkeiten zur Anpassung. Es lässt sich das Frequenzband der Übertragung und die Anzahl der zu übertragenen Bits pro Sound konfigurieren.

¹<https://github.com/MrLaki5/Data-over-sound>

ZXing Android Embedded

Für das Erstellen und Lesen (Anforderung 01-004 und 01-005) von QR-Codes soll die ZXing Android Embedded Bibliothek ² genutzt werden. Sie nutzt zur Decodierung die ZXing Bibliothek, die in vielen Anwendungen genutzt wird und diverse Barcodes lesen kann. Da bei der Wahl der Bibliothek auch auf Industrielle Standards gesetzt werden soll, wurde sich für diese entschieden, weil sie auch von der Corona-Warn-App³ genutzt wird (31 Millionen Downloads [10]).

Die Bibliothek ermöglicht es ein Vorschaubild von dem Bereich darzustellen, auf den die Kamera des Geräts gerade ausgerichtet ist. Das soll die Anwendung vereinfachen und Fehler vermeiden. Zudem bietet sie die Möglichkeit die Fehlerkorrekturstufe zu variieren.

4.2.2 Protokoll

Verbindungslose Kommunikation

Es wird eine verbindungslose Kommunikation stattfinden. Die Anwendung wird auf einem Smartphone in einen Empfänger Modus versetzt und wartet darauf, dass Übertragungen neue Daten, liefern die abgespeichert werden sollen. Eine Bestätigung über den Empfang wird nicht gesendet.

Transiente Übertragung

Eine versendete Information muss in dem Moment empfangen werden in dem sie versendet wird. Ein Speichern dieser, um sie später zu übertragen, ist nicht vorgesehen. Bevor ein Gerät die Übertragung beginnt, muss das Empfängergerät in den Empfangsmodus wechseln.

²<https://github.com/journeyapps/zxing-android-embedded>

³<https://github.com/corona-warn-app>

Datenverlust

Im Fokus der Beobachtungen dieser Arbeit steht der Datenverlust in der Kommunikation zwischen den beiden Smartphones. Mit Verlusten wird gerechnet, da nur diese für die Auswertung von Bedeutung sind, wurde eine Re-Transmission, ähnlich wie in TCP, nicht implementiert.

Übertragungsgeschwindigkeit

Die Geschwindigkeit in der Daten versendet werden legt die Anwendung, welche sich im Sendemodus befindet, fest. Sie basiert auf der Konfiguration, welche Grundlage für die Kommunikation ist.

Datenaufbau

In den Experimenten wird gemessen wie hoch der Datenverlust bei den Übertragungen ist. Es ist davon auszugehen, dass es bei ihnen zu Fehlern kommen, wird die eine spätere Evaluation erschweren. Um die Auswertung zu erleichtern werden die Datenpakete auf eine Art aufgebaut, die der Auswertung helfen soll.

Ein Datensatz soll mit einer eindeutigen ID erkennbar sein (Anforderung 02-001) in Form einer Data-ID. Da jeder Datensatz wiederholt übertragen werden soll (Anforderung 001-008), muss eine Möglichkeit geschaffen werden jede Wiederholung eindeutig zuzuordnen (Anforderung 02-002). Dies soll geschehen, indem jede Übertragung eine eindeutige Sequence-ID erhält. Es soll die Möglichkeit gegeben sein, Datensätze in mehreren Sequenzen zu übertragen. Dafür muss mithilfe einer Sequence-Nummer erkennbar sein welcher Teil dieser Sequence übertragen wird und mit einer Sequence-Length aus wie vielen Teilen sie besteht (Anforderung 02-003).

Daraus ergibt sich der Aufbau wie in Tabelle 4.1 zu sehen.

<—— 80 Bit ——>

Sequence-ID	Sequence-Nummer
Sequence-Length	Data-ID
Data	
Data	
...	

Tabelle 4.1: Aufbau eines zu versendenden Datenpakets

4.2.3 Parameter

Es soll untersucht werden, inwiefern einzelne Faktoren bei der Übertragung Auswirkungen auf den Verlust von Datenpaketen oder einzelne Bits haben (Anforderung 01-009). Zusätzlich werden alle Übertragungen in verschiedenen Abständen durchgeführt. Da noch nicht abzusehen ist, wie stark der Datenverlust bei den verwendeten Geräten sein wird, werden die finalen Parameter erst im Laufe der Versuche ermittelt.

Data-Over-Sound

Die Bibliothek bietet die Möglichkeit folgende Parameter zu verändern:

Bits pro Ton

Es kann definiert werden wie viele Bits in einen Ton codiert werden.

Dauer eines Tons

Die Töne werden eine bestimmte Zeit gespielt und müssen in dieser Zeit vom Empfänger erkannt werden.

Frequenzband

Eine Start- und Endfrequenz können definiert werden. Sie bestimmen wie breit das Spektrum der Übertragung ist und wie nah die Töne aneinander liegen.

QR-Code

Folgende Parameter sollen bei der Übertragung mittels QR-Codes variiert werden:

Fehlerkorrektur

Es wird erwartet, dass die QR-Codes zuverlässiger gelesen werden können. Eine höhere Fehlerkorrektur erhöht aber auch den Datensatz und somit auch die Auflösung des QR-Codes, weil das Display nur eine begrenzte Fläche zur Darstellung bietet.

Datenmenge

Die codierte Datenmenge erhöht die Auflösung des QR-Codes. Es ist zu befürchten, dass die Kameras der Geräte nicht jede Auflösung erkennen können.

Sendegeschwindigkeit

Das Sendegerät zeigt neue QR-Codes in verschiedenen Frequenzen.

4.2.4 User Interface des Traffic Generators

Zweck der Anwendung ist es Daten zu generieren und zu verschicken. Das soll so weit wie möglich von selbst geschehen. Sie soll aber auch Informationen darüber geben was gerade geschieht und es soll möglich sein, einzelne Konfigurationen innerhalb der Anwendung zu generieren. Abb. 4.1 zeigt einen Entwurf wie die Anwendung aussehen und funktionieren soll.

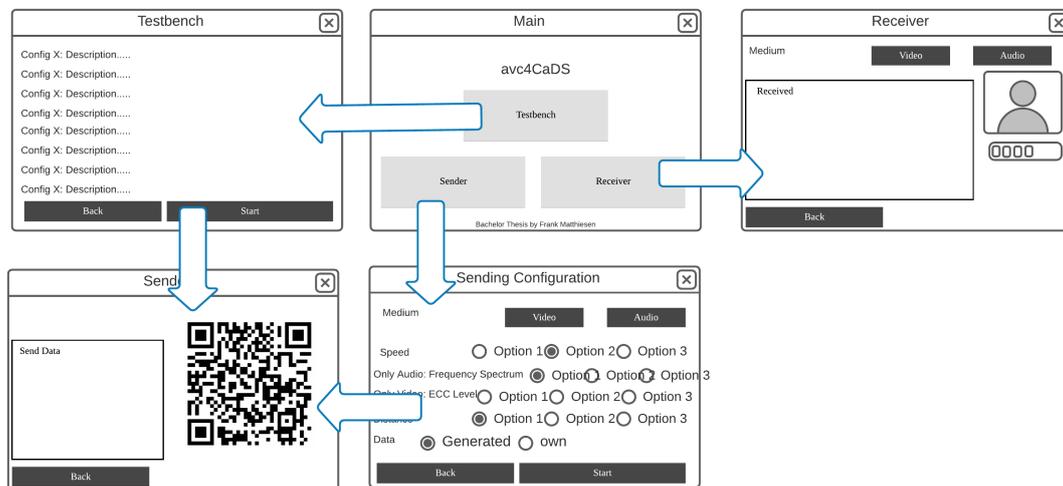


Abbildung 4.1: Wireframe der zu entwickelnden Anwendung

Main

Die Startseite der Anwendung soll die Wahl geben Daten zu senden oder in den Empfangsmodus zu wechseln. Der Button Testbench führt zu den automatisierten Tests (Anforderung 01-008).

Receiver

Befindet sich die Anwendung im Empfangsmodus, soll die Kamera aktiviert werden und zeigen welcher Bereich gescannt wird. Ein Textfeld soll Debug Informationen und empfangene Daten anzeigen.

Testbench

Definierte Sende-Konfigurationen sollen auswählbar sein und anschließend versendet werden.

Sender

Der Sendemodus muss einen QR-Code darstellen können und in einem Textfeld Informationen über den aktuellen Zustand der Übertragung darstellen.

Sending Configuration

Die Parameter für eine Übertragung und den eventuellen Datensatz soll hier konfigurierbar sein.

4.2.5 Daten sichern

Der Traffic Generator soll zufällig erstellte Datenmengen übertragen, um eine möglichst ausgewogene Übertragung sicherzustellen. Wenn beispielsweise immer nur der gleiche Datensatz übertragen wird, kann nicht ausgeschlossen werden, dass die Fehlerrate nicht repräsentativ wird.

Die Datensätze müssen für die spätere Auswertung gesichert werden (Anforderung 01-006). Dabei dürfen die Daten nicht verändert oder verlustbehaftet komprimiert werden.

Sender

Die sendende Applikation soll die Daten, die Sie versendet, in einer Textdatei speichern. Es ist wichtig, dass die Daten in einer Form gespeichert werden, die es ermöglicht jeden zu versendenden Datensatz mit den empfangenen Daten zu vergleichen. Idealerweise ist

die Konfiguration der Übertragung auch gespeichert. Die Konfiguration ist nicht für die Übertragung notwendig, aber sollte bei der späteren Evaluation die Arbeit erleichtern.

Empfänger

Die empfangende Applikation soll alle empfangenen Daten in einer Textdatei abspeichern. Alle Daten sollen mit den versendeten Daten verglichen werden. Es ist also notwendig einen Aufbau zu wählen, welcher dies ermöglicht.

Software-Architektur

Die Anwendung dient der Übertragung von Daten von einem Gerät zum Anderen, so wie in Abb. 4.2 dargestellt.

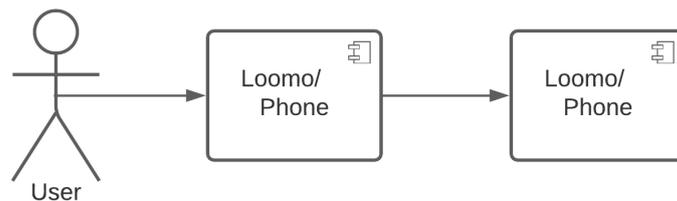


Abbildung 4.2: Ablauf einer Übertragung

Der interne Ablauf der Anwendung sollte dabei wie in Abb. 4.3 aussehen.

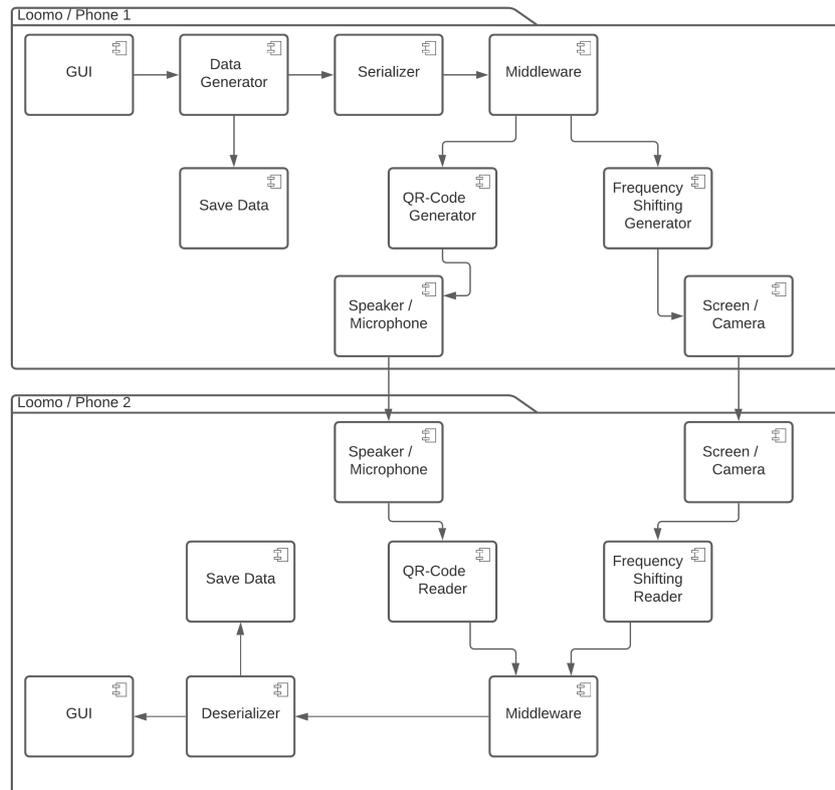


Abbildung 4.3: Komponenten der Übertragung

4.3 Mögliche Fehlerquellen

Im Vorfeld gilt es zu betrachten, welche Fehler bei der Messung auftreten können, damit diese gezielt umgangen werden können.

Wenngleich beide Übertragungsarten für sich besondere Anforderungen stellen, so sollten bei allen Experimenten die Smartphones so positioniert sein, dass sie nicht bewegt werden können. Bei beiden Smartphone sollten zudem Hintergrundaktivitäten vom Betriebssystem, wenn möglich deaktiviert werden, damit die Übertragungen nicht gestört werden.

4.3.1 Data-Over-Sound

Die Daten werden über Schallwellen, also durch Druck- und Dichteschwankungen, in der Luft bewegt. Da die Luft ein geteiltes Medium ist, kann nicht ausgeschlossen werden, dass die Übertragung von anderen Geräten gestört wird. Insbesondere da Frequenzen genutzt werden, welche von Menschen nicht erfasst werden können. Aufgrund dieser Einschränkung des menschlichen Hörvermögens kann eine Störquelle bei den Experimenten gar nicht auffallen.

Es ist deshalb ratsam, ein Frequenzspektrometer parallel zu den Experimenten zu betreiben. Dabei handelt es sich, um ein Werkzeug, das Töne grafisch darstellt. Anwendungen, die solche Funktionen bieten, gibt es inzwischen auf diversen Smartphone Plattformen.

4.3.2 QR-Codes

Ist ein QR-Code auf dem sendenden Smartphone erstellt, muss dieser von dem empfangenden Smartphone mithilfe der Kamera gescannt werden. Es ist zu erwarten, dass dieser Prozess gestört wird, wenn die Beleuchtung sich rasch verändert und den Kontrast der Datenblöcke verändert. Zudem kann nicht ausgeschlossen werden, dass Bewegungen im Sichtfeld der Kamera das Smartphone dazu bewegt den Fokus neu zu setzen.

Beim Durchführen der Experimente muss darauf geachtet werden, dass eine gleichmäßige Beleuchtung und eine möglichst gleichmäßige Umgebung ohne Bewegungen im Hintergrund sichergestellt sind.

5 Umsetzung

5.1 Software

5.1.1 Android MVC

Für die Entwicklung einer Anwendung, die auf Geräten mit Android lauffähig ist, gibt es verschiedene Sprachen und Ansätze. Für diese Arbeit wurde sich für Java mit dem Android SDK entschieden.

Das Android SDK enthält alle Klassen, welche der Anwendung den Zugriff auf das Android Framework ermöglichen. Insbesondere solche die, den Android Ansatz des Model-View-Controller Architekturmusters ermöglichen. Hierfür gibt es insbesondere die **Activity** Klasse. Mit ihr, und einem im XML-Format beschriebenen Layout, ist es möglich Oberflächen (Views) auf dem Smartphone darzustellen.

Basierend auf Abb. 4.1 wurden folgende Oberflächen und Layouts implementiert:

Main Activity

Wird als erstes von Android geladen. Sie bietet die Möglichkeit die einzelnen Funktionen der Anwendung zu starten.

Receiver Activity

Greift durchgehend auf Kamera und Mikrofon zu, um Datensätze empfangen und sichern zu können.

SendConfig Activity

Bietet die Möglichkeit ein Configuration-Objekt für die Übertragung zu konfigurieren.

Sending Activity

Muss mit einem Configuration-Objekt initialisiert werden und überträgt dessen Daten abhängig von dessen Parametern.

Testbench Activity

Stellt eine Reihe von Konfigurationen bereit, die je nach Auswahl übertragen werden können.

5.1.2 Configuration Object

Jede Übertragung wird definiert durch ein Configurations Objekt. Dieses beinhaltet alle notwendigen Parameter für eine Übertragung und den Datensatz der übertragen werden soll.

medium

Gibt an, ob eine audio- oder videobasierte Übertragung durchgeführt werden soll.
Bsp.: AUDIO, VIDEO.

speed

Legt die Aktualisierungsrate der QR-Codes fest. Bsp: Speed1, Speed2.

spectrum

Breite des Audiospektrums. Beispiel: FULL, INAUDIBLE.

eccLevel

Fehlerkorrekturstufe der QR-Codes. Beispiel: LEVELQ, LEVELM.

distance

Dient nur der Evaluation. Beispiel: DISTANCE1, DISTANCE2.

data

Der zu übermittelnde Datensatz.

loops

Legt die Anzahl der Wiederholungen für die Experimente fest.

uuid

Jede Konfiguration wird mit einem Universally Unique Identifiert initialisiert

5.1.3 Object Passing in Android

Die Sending Activity benötigt ein Configuration-Objekt, in dem die zu sendenden Daten stehen und ebenso Informationen über die Art und Weise, wie diese übertragen werden. Hierfür wurde nicht auf ein globales Objekt gesetzt, sondern der Activity wird das Objekt übergeben.

Im Android Framework wird der Wechsel von einem View zu einem anderen mit einer Instanz der **Intent** Klasse durchgeführt. Es ist allerdings mit ihr nicht möglich komplexe Datentypen zu übergeben. Das Android SDK bietet jedoch die Möglichkeit **Bundle** Objekte zu übergeben.

Damit die Configuration-Klasse übergeben werden kann, muss sie das **Parcelable** Interface implementieren. Das Android Framework ruft nun beim Übergeben die `void writeToParcel(Parcel dest, int flags)` Methode auf, in der alle Werte der Konfiguration dem Parcel Objekt angehängt werden. Das Objekt wird so für die Übergabe gemarshalled, wie in Listing 5.1 zu sehen ist.

```
1 public void writeToParcel(Parcel dest, int flags) {
2     dest.writeString(getMedium().name());
3     dest.writeString(getSpeed().name());
4     dest.writeString(getSpectrum().name());
5     dest.writeString(getEccLevel().name());
6     dest.writeString(getDistance().name());
7     dest.writeString(getData());
8     dest.writeInt(getLoops());
9     dest.writeString(getUuid());
10 }
```

Listing 5.1: Marshalling der Configuration-Klasse

Die Configuration-Klasse wurde um einen Konstruktor erweitert, welcher ein Parcel-Objekt erhält und die Werte in der Reihenfolge, wie sie angehängt wurden, liest und ein neues Objekt konstruiert, wie zu sehen in Listing 5.2.

```
1 protected Configuration(Parcel in) {
2     this.medium = Medium.valueOf(in.readString());
3     this.speed = Speed.valueOf(in.readString());
4     this.spectrum = Spectrum.valueOf(in.readString());
5     this.eccLevel = EccLevel.valueOf(in.readString());
6     this.distance = Distance.valueOf(in.readString());
```

```

7   this.data = in.readString();
8   this.loops = in.readInt();
9   this.uuid = in.readString();
10  }

```

Listing 5.2: Unmarshalling der Configuration-Klasse

5.1.4 Sending Activity Architektur

Die Sending Activity, welche für das versenden der Daten zuständig ist, hat den in Abb. 5.1 dargestellten Aufbau.

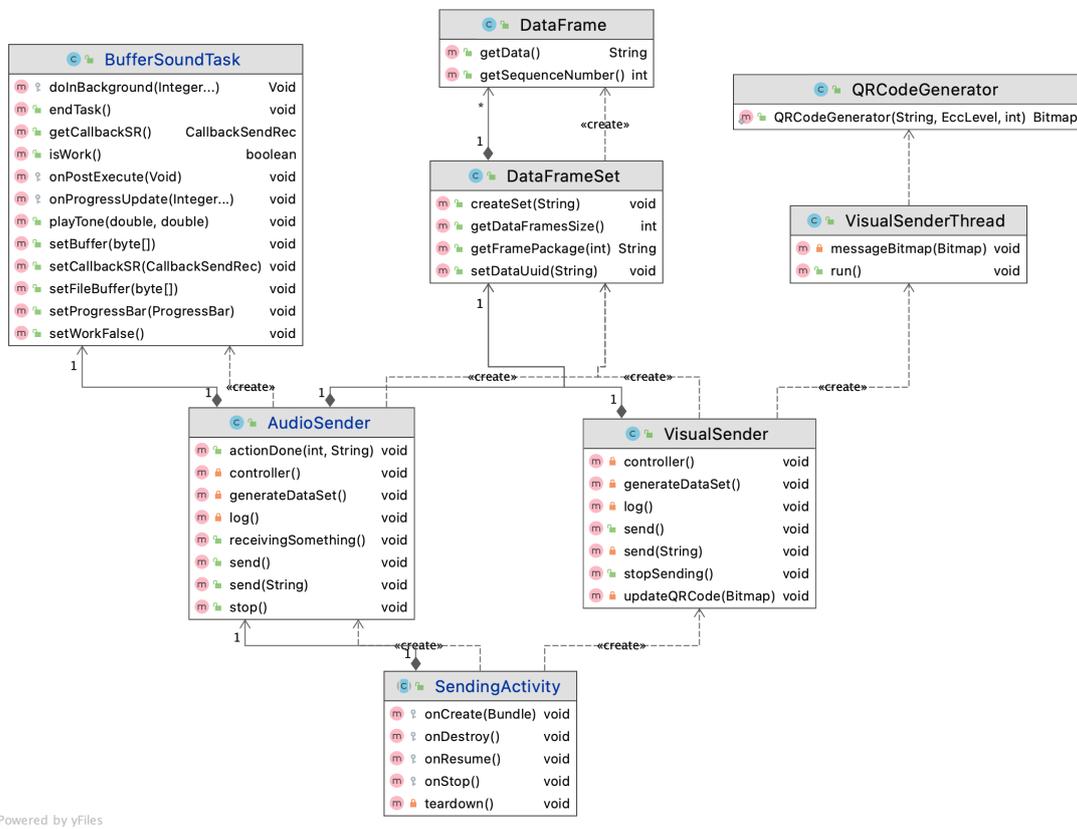


Abbildung 5.1: Aufbau der Sending Activity

Ablauf QR-Codes Übertragung

Wenn sich im Configuration-Objekt die Information befindet, dass eine videobasierte Übertragung stattfinden soll, wird ein Objekt der VideoSender-Klasse initialisiert. Damit die QR-Codes eine Obergrenze der Datenmenge haben und nicht zu hochauflösend für Displays sind, werden die zu übermittelnden Daten in einzelne DataFrame-Objekte aufgeteilt. Die dadurch entstehende Sequenz wird dann nacheinander als QR-Code auf dem Display des Smartphones dargestellt.

Würde dies ohne Unterbrechung geschehen, wäre es aufgrund der Framerate der Kamera und der Rechengeschwindigkeit nicht möglich alle QR-Codes zu scannen. Wenn ein QR-Code auf dem Display dargestellt ist, wird der Thread eine definierte Zeit pausiert, ehe der Nächste gezeigt wird. Damit die Anwendung dennoch weiterhin Eingaben verarbeiten kann, werden die QR-Codes in jeweils einem Thread generiert. Abb. 5.2 stellt diesen Ablauf dar.

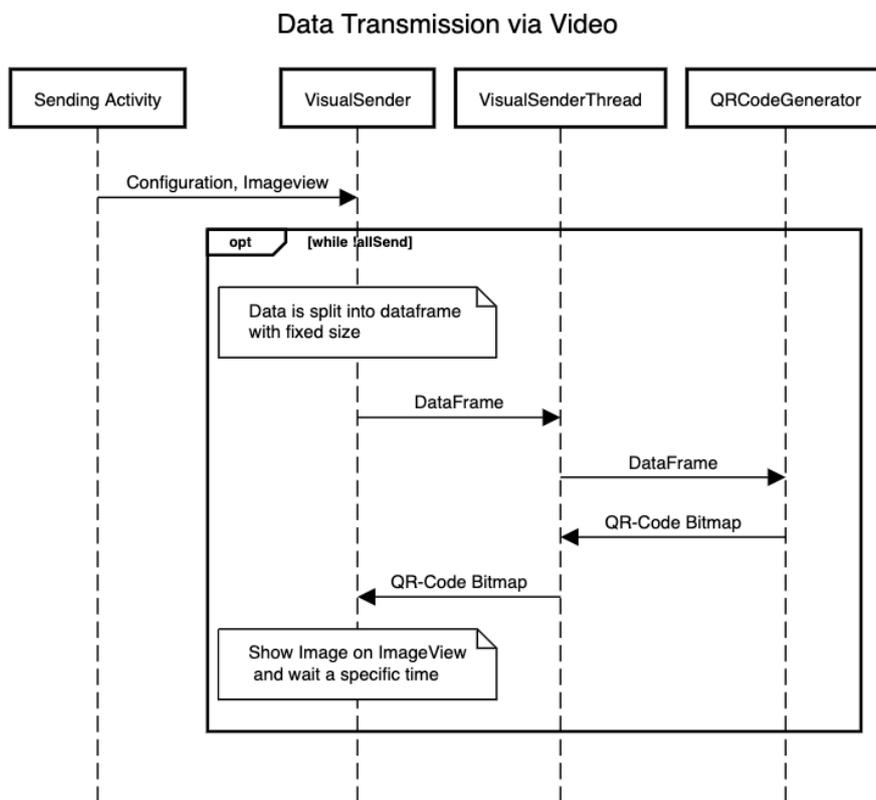


Abbildung 5.2: Sequenz einer Übertragung mittels QR-Codes

Ablauf Data-over-Sound Übertragung

Bei der Datenübertragung mittels Data-over-Sound müssen die Töne eine bestimmte Zeit gehalten werden, damit das empfangende Smartphone die Information erhalten kann. Die Data-over-Sound Bibliothek hat diesen nebenläufigen Algorithmus bereits implementiert, weshalb nur ein `BufferSoundTask` Objekt initialisiert werden muss.

Dieses erfordert die Frequenzen der Übertragung, die zu übermittelnden Daten und wieviele Informationen pro Ton übertragen werden sollen. In Abb. 5.3 ist dieser Ablauf dargestellt.

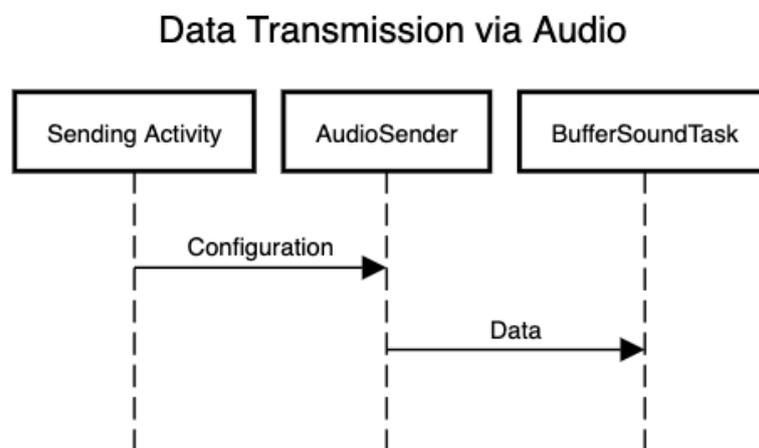


Abbildung 5.3: Sequenz einer Übertragung mittels Data-over-Sound

5.1.5 Speicherform der Daten

Da mit fehlenden oder fehlerhaften Daten auf der Empfängerseite gerechnet wird, werden die Daten in einer Form gespeichert, die es ermöglicht, ohne manuelle Korrektur die Übertragungen zu evaluieren.

Jede Übertragung wird in einer Textdatei als Zeile angehängt. Der Aufbau ist identisch mit dem Aufbau eines Datenpakets, siehe Tabelle 4.1. Wenn es bei der Übertragung zu keinen Fehlern kommt, so sehen die Zeilen auf Sender und Empfängerseite identisch aus.

Ein Auszug aus der Textdatei ist in Listing 5.3 zu sehen.

```
1 nY5Kn000010000160b69AUDIO, SPEED1, INAUDIBLE, LEVELM, DISTANCE1
  , GENERATED, , 10, 60b699VyUT
2 bIMpH000010000160b69AUDIO, SPEED1, INAUDIBLE, LEVELM, DISTANCE1
  , GENERATED, , 10, 60b699VyUT
3 Er1ub000010000160b69AUDIO, SPEED1, INAUDIBLE, LEVELM, DISTANCE1
  , GENERATED, , 10, 60b699VyUT
```

Listing 5.3: Gespeicherte Datenpakete

5.2 Experimentaufbau

5.2.1 Smartphones

Für die Experimente wurden zwei Smartphones ausgewählt. Da davon ausgegangen wird, dass unterschiedliche Geräte einen Einfluss auf das Ergebnis der Experimente haben, ist Wert darauf gelegt worden identische Geräte zu benutzen. Die Wahl fiel auf zwei Samsung Galaxy S7 Edge Smartphones. Diese liegen der Forschungsgruppe CaDS HAW Hamburg vor und ihre technische Ausstattung ließ darauf schließen, dass sie aussagekräftige Ergebnisse liefern werden.

Ausgestattet sind sie mit einem Snapdragon 820 System-on-Chip. Dieser bietet 4 Kerne, die mit bis zu 2.3GHz getaktet werden und 4GB Arbeitsspeicher. Betrieben wird das Smartphone von Android 7.0 mit einer angepassten Oberfläche von Samsung.

Interessant für die Experimente sind das Display, die Front-Kamera, die Lautsprecher und die Mikrofone. Das Display hat eine Diagonale von 5.5Zoll und bietet dabei eine Auflösung von 1440×2560 Pixeln, was einer Pixeldichte von ungefähr 534Dpi entspricht. Die Frontkamera bietet im kontinuierlichen Modus (Video) eine Auflösung von 1920×1080 bei 30 Bildern pro Sekunde. Beim Sampling ist eine Wiedergabe von $44,1\text{kHz}$ und 48kHz möglich. Das Sampling bei der Aufnahme soll bis zu 192GHz mit 24bit sein ¹.

5.2.2 Experimentaufbau

Aufgrund der Covid-19 Pandemie und der damit einhergehenden Beschränkungen, konnten die Experimente nicht in den Räumen der HAW-Hamburg durchgeführt werden. Sie

¹<https://www.qualcomm.com/products/snapdragon-820-mobile-platform>

finden in den Räumlichkeiten des Autors statt, wurden aber mit so wenigen Störungen wie möglich durchgeführt.

Abb. 5.4 zeigt den Experimentaufbau. Die Smartphones sind parallel mit ihren Fronten zueinander gewandt aufgestellt.

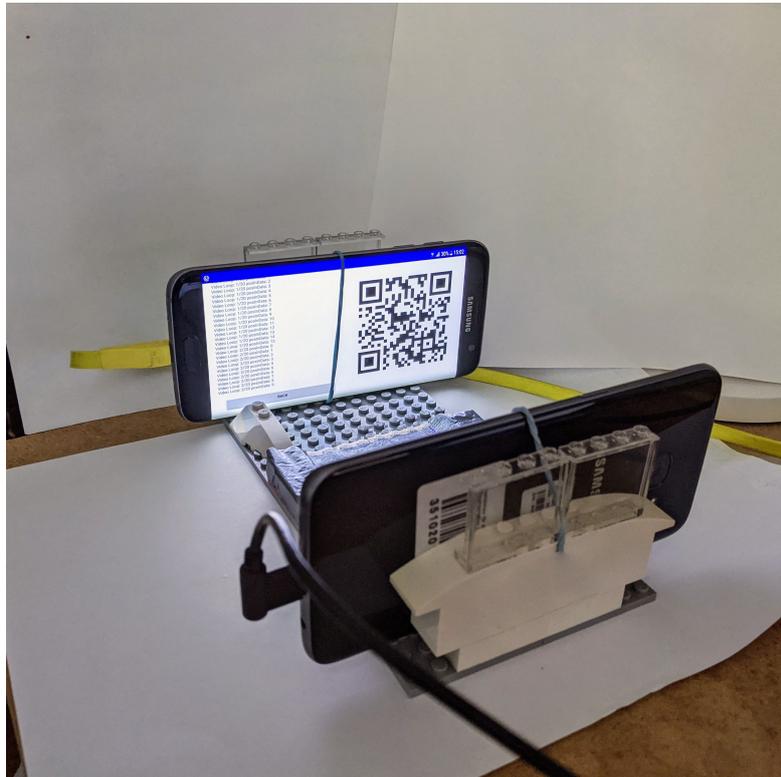


Abbildung 5.4: Testaufbau mit beiden Smartphones bei der Übertragung mittels QR-Codes

Der Aufbau erfolgte wie beschrieben, sodass die Kamera optimal auf den jeweiligen QR-Code gerichtet ist. Eine Konstruktion aus Klemmbausteinen und Gummibändern half dabei die Geräte in eine aufrechte Position zu bringen und zu fixieren (Anforderung 03-006).

Die Beleuchtung im Raum wurde reduziert und die Fenster verschlossen, damit es zu keinen Lichtreflexionen auf den Bildschirmen der Geräte kam (Anforderung 03-001 und 03-002). Damit die Kamera keine Veränderungen des Fokus durchführt oder vermeintlich Muster erkannt werden, wurde die Umgebung des Experiments mit weißem Papier ausgekleidet (Anforderung 03-003).

Bei den Übertragungen mittels Data-over-Sound wurden die Halterungen weiter verwendet, um einen sicheren Stand zu bieten. Die Fenster waren auch bei diesen Experimenten geschlossen und es wurde darauf geachtet, dass keine Hintergrundgeräusche stören. Nebenbei wurde während der Experimente mit einem Spektrometer überprüft, dass auch auf den Frequenzen, die Menschen nicht hören können, keine Störungen auftreten (Anforderung 03-004). Beispielhaft ist dies in Abb. 5.5 zu sehen. Der Screenshot zeigt die Audiosignale der letzten ca. 22 Sekunden. Die X-Achse stellt dabei die Zeit dar und die Y-Achse den Frequenzraum von 0 kHz–20 kHz. Die Intensität von Signalen wird über die Farbe dargestellt, von blau (sehr gering) bis rot (sehr stark). Die Übertragung ist im oberen Spektrum zu erkennen und zu sehen ist, dass keine anderen Geräusche in diesem Spektrum auftraten.

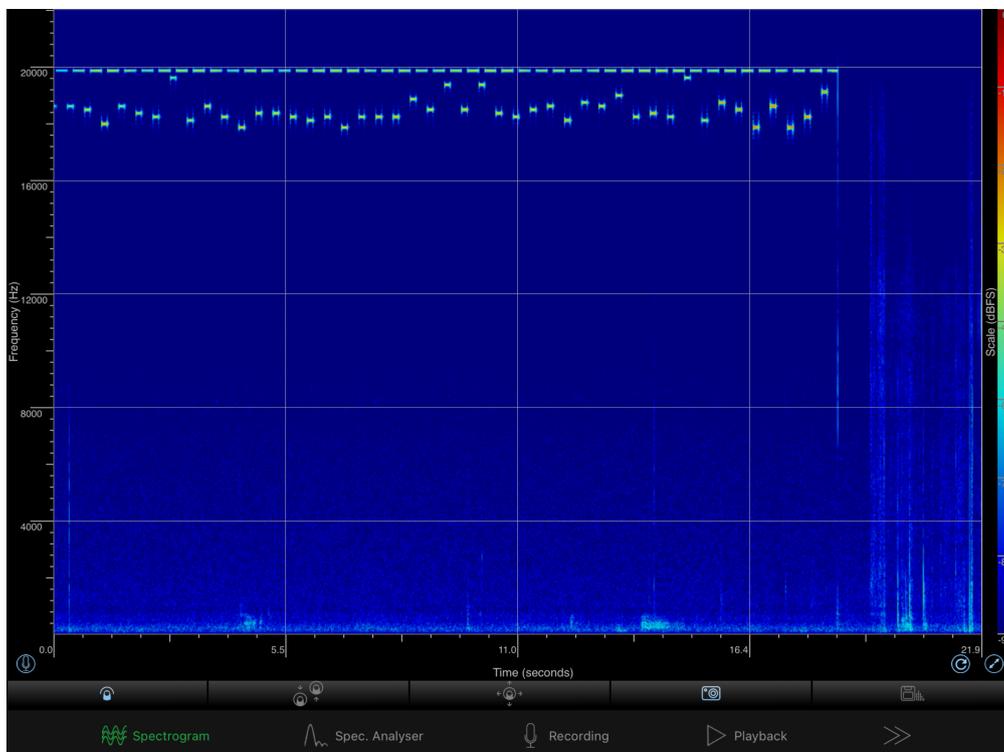


Abbildung 5.5: Spektrometer kurz nach einer Übertragung

5.2.3 Parameter

Im Zuge der Implementierung des Traffic-Generators wurden eine Reihe von Parametern konfigurierbar implementiert, um deren Einfluss auf die Ergebnisse beurteilen zu können.

Zusätzlich soll beobachtet werden wie sich die Übertragungsarten bei verschiedenen Distanzen verhalten.

Data-over-Sound Übertragung

Bei der Data-Over-Sound Bibliothek sind es:

Geschwindigkeit

Die Dauer wie lange ein Ton gehalten wird.

Bits pro Ton

Die Anzahl an Bits, die pro Übertragung versendet werden.

Start und Endfrequenz

Das sich dadurch ergebene Audiospektrum und der Abstand der Töne während der Übertragung.

QR-Codes Übertragung

Bei der QR-Codes Übertragung sind es:

Geschwindigkeit

Die Dauer der Darstellung eines QR-Codes.

Fehlerkorrektur

Die damit hervorgehende Fehlertoleranz.

Obergrenze der Daten

Eine Obergrenze der Daten pro QR-Code und die damit einhergehende Auflösung des QR-Codes.

6 Ergebnisse

6.1 Parameterauswahl

In kleinen Testreihen wurden vorab die Parameter der Übertragungsarten betrachtet.

Bei Data-over-Sound ist es möglich zu konfigurieren wie viele Bytes pro Ton übertragen werden sollen. Allerdings scheint die Bibliothek fehlerhaft zu sein, denn ab *5Byte/Ton* wurden keine Töne mehr erzeugt. Die Übertragung von *2Byte/Ton* und *4Byte/Ton* wurde untersucht, aber da abgesehen von der längeren Übertragungszeit keine Unterschiede zu erkennen waren, wurde dieser Parameter nicht ausführlicher betrachtet.

Bei QR-Codes erschien die Fehlerkorrekturstufe vorab interessant, jedoch wirkte sich diese nur auf die Größe und dementsprechende Auflösung des QR-Codes aus. Da eine höhere Auflösung zu mehr nicht gelesenen Codes führte, wurde dieser Parameter nicht weiter betrachtet.

Alle weiteren Parameter wurden durch Experimente ausgewählt. Beispielsweise wäre bei QR-Codes ein größerer Abstand als 25 cm wünschenswert gewesen, jedoch wurden bei größeren Distanzen kaum noch Codes erkannt.

6.2 Erwartungen

Erfolgreiche Übertragungen zeichnen sich durch verschiedene Eigenschaften wie die Bandbreite oder das Delay. Im Folgenden wird betrachtet, was für Werte im Idealfall erwartet werden.

6.2.1 Data-over-Sound Übertragung

Betrachtet werden Parameter von denen erwartet werden, dass sie die Stabilität der Übertragung, mit blick auf die Fehlerrate, beeinflussen. Aus diesem Grund wird mit der gleichen Bandbreite, bei allen Übertragungen, gerechnet. Die Bandbreite der Übertragung ergibt sich aus der Anzahl der Bytes die pro Ton übertragen werden und wie lange dieser Ton gehalten wird. Konfiguriert ist die Übertragung so, das pro Ton 4 Byte übertragen werden und jeder Ton 0,270s gehalten wird. Daraus ergibt sich eine mögliche Übertragungsrate von $14,81 \text{ Byte/sekunde}$.

Das entstehende Delay basiert auf der Dauer, die für die Aufbereitung der Daten benötigt, der Geschwindigkeit der Schallwellen und der Decodierung des Signals. Der Geschwindigkeit der Schallwellen beträgt $343,2 \text{ m} \times \text{s}$. Daraus ergibt sich bei einem Abstand von 1 m ein Delay von mindestens 29,13 ms.

6.2.2 QR-Codes Übertragung

Die Bandbreite der QR-Codes Übertragung ergibt sich aus Größe eines Datensatzes pro QR-Code und der Zeit die jeder dargestellt wird. 6.1 zeigt die möglichen Bandbreiten. Es handelt sich hierbei um die Netto-Bandbreite, ohne die Informationen die zusätzlich durch Fehlerkorrektur und Codierungsinformationen generiert werden.

Tabelle 6.1: Mögliche Übertragungsraten bei verschiedenen Konfigurationen der QR-Code Übertragung

QR-Code Dauer	QR-Code Größe	Bandbreite
100ms	200 Bit	2 kbit/s
75ms	200 Bit	2,66 kbit/s
50ms	200 Bit	4 kbit/s
100ms	320 Bit	3,2 kbit/s
75ms	320 Bit	4,26 kbit/s
50ms	320 Bit	6,4 kbit/s

Das zu erwartende Delay beruht, bei der Übertragung, auf der Aufarbeitung der Daten der Generierung und lesen der QR-Codes. Da die Übertragung optisch geschieht, ist die

Zeit, die das Licht für kurze Distanz zurücklegt, durch seine hohe Geschwindigkeit, zu vernachlässigen.

6.3 Aufbereitung der Daten

Die übermittelten Daten wie sie in Listing 5.3 zu sehen sind, galt es zu analysieren.

Durch den homogenen Aufbau und die Informationen zur Übertragung in jeder Zeile, war es möglich die Daten auszuwerten.

Für die Auswertung wurde ein Python-Skript entwickelt, welches zunächst alle gesendeten und empfangen Zeilen einliest. Aus den Informationen über die Übertragung (siehe Tabelle 4.1) konnten die einzelnen Übertragungen, die Wiederholungen und die übermittelten Daten extrahiert und strukturiert werden.

Für die Auswertung der Übermittlung von Data-over-Sound wurden die Daten als Byte-Stream geladen. Jedes Byte wurde zu einem Bit-Array umgewandelt. Anschließend wurde die Bitfolge der gesendeten Daten mit denen der empfangenen verglichen, der Hamming Abstand [8] gebildet und im Verhältnis zur Länge der Folge gesetzt.

Bei der Übertragung mittels QR-Codes konnte durch die Information, aus wie vielen Teilen die Übertragung besteht, geschlossen werden wie viele Frames bei der Übertragung verloren gegangen sind.

6.4 Datenübertragung mit Data-over-Sound

6.4.1 Beobachtung

Die Übertragung mittels Data-over-Sound wurde mit verschiedenen Konfigurationen durchgeführt. Eine Zusammenfassung der aufgetretenen Bitfehler ist in Abb. 6.1 zu sehen. In dem nun folgenden Teil werden die Konfigurationen im Einzelnen betrachtet.

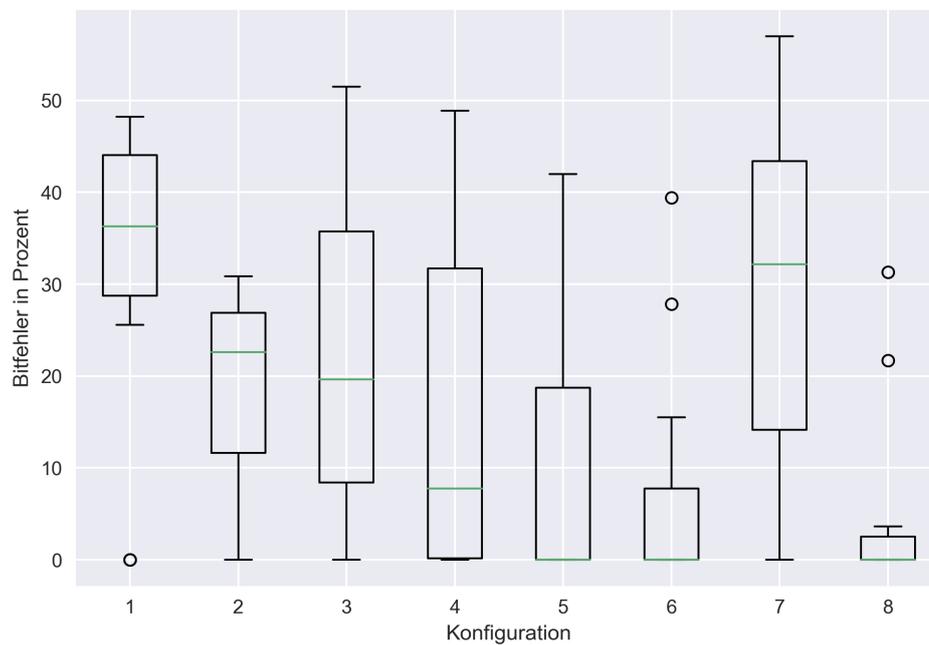


Abbildung 6.1: Bitfehler über die verschiedenen Konfigurationen

Konfiguration 1 - 2: Abstand 15 cm, mit verschiedenen Audiospektren

Bei 15 cm Abstand und einem Spektrum von 2,5 kHz waren zwei Übertragungen fehlerfrei. Alle anderen hatten Fehlerraten von 24 %–48 %. Bei einem Spektrum von 5 kHz waren zwei Übertragungen fehlerfrei. Alle anderen hatten Fehlerraten von 5 %–31 %. Wie in Abb. 6.2 zu sehen.

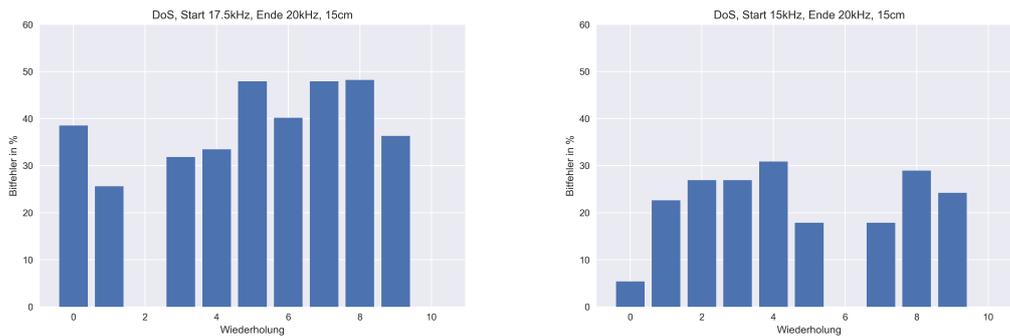


Abbildung 6.2: Verschiedene Audiospektren bei 15 cm Abstand

Konfiguration 3 - 4: Abstand 100 cm, mit verschiedenen Audiospektren

Bei 100 cm Abstand und einem Spektrum von 2,5 kHz war eine Übertragung fehlerfrei und eine mit einer Fehlerrate von 1 %. Alle anderen hatten Fehlerraten von 15 %–51 %. Bei einem Spektrum von 5 kHz waren drei Übertragungen fehlerfrei und eine mit einer Fehlerrate von 1 %. Alle anderen hatten Fehlerraten von 7 %–49 %, wie in Abb. 6.3 zu sehen ist.

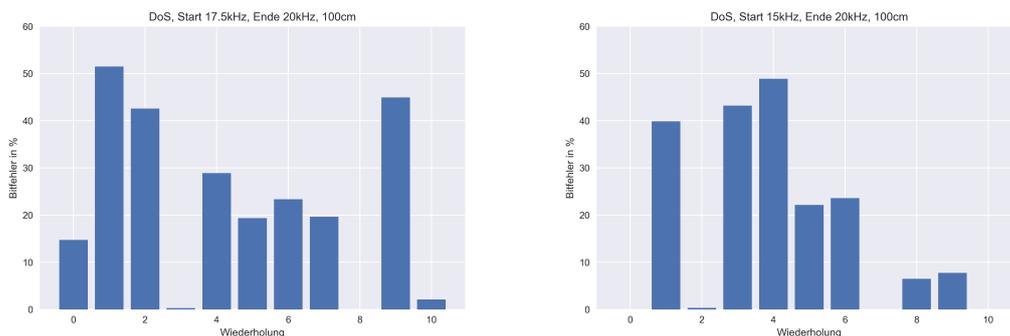


Abbildung 6.3: Verschiedene Audiospektren bei 100 cm Abstand

Konfiguration 5 - 6: Abstand 150 cm, mit verschiedenen Audiospektren

Bei 150 cm Abstand und einem Spektrum von 2,5 kHz waren sechs Übertragungen fehlerfrei. Alle anderen hatten Fehlerraten von 18 %–42 %. Bei einem Spektrum von 5 kHz waren sieben Übertragungen fehlerfrei. Alle anderen hatten Fehlerraten von 16 %–39 %, wie in Abb. 6.4 zu sehen ist.

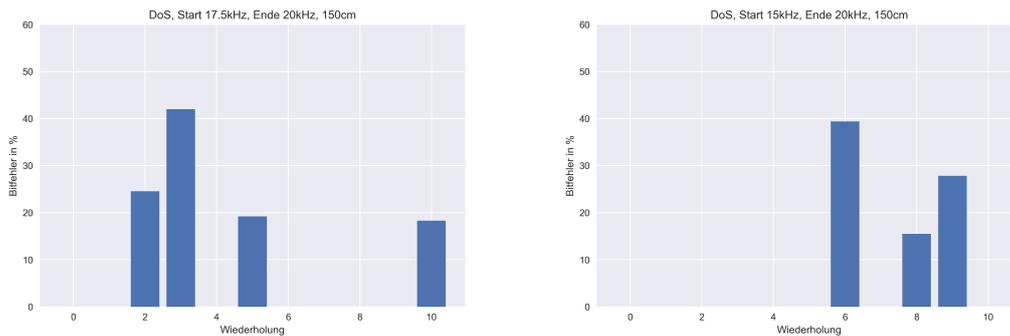


Abbildung 6.4: Verschiedene Audiospektren bei 150 cm Abstand

Konfiguration 7 - 8: Abstand 300 cm, mit verschiedenen Audiospektren

Bei 300 cm Abstand und einem Spektrum von 2,5 kHz war eine Übertragung fehlerfrei und zwei lagen unter 5 %. Alle anderen hatten Fehlerraten von 26 %–57 %. Bei einem Spektrum von 5 kHz waren fünf Übertragungen fehlerfrei und drei lagen bei unter 5 %. Alle anderen hatten Fehlerraten von 22 %–31 %, wie in Abb. 6.5 zu sehen.

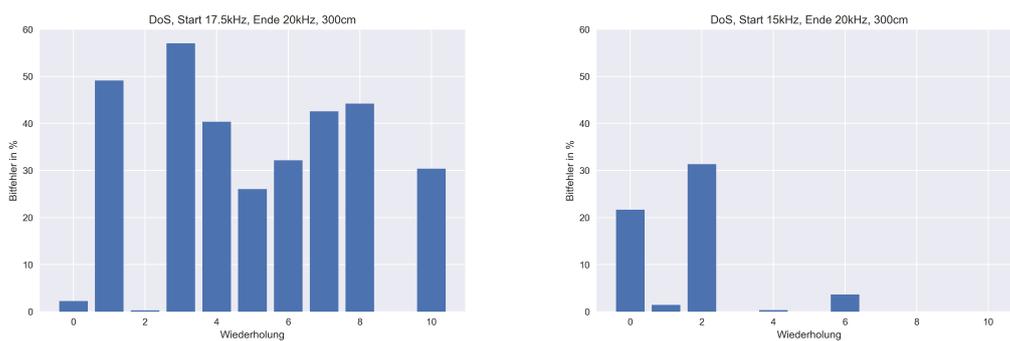


Abbildung 6.5: Verschiedene Audiospektren bei 300 cm Abstand

6.4.2 Diskussion

Insgesamt gab es eine hohe Varianz bei den Fehlerraten der Übertragungen.

Abstand

Ein größerer Abstand der Geräte zueinander scheint die Übertragungen bis zu einer gewissen Distanz zu verbessern.

Bei allen Übertragungen waren die Smartphones mit 80 % ihrer maximalen Lautstärke eingestellt. Es wird vermutet, dass bei zu geringem Abstand Störungen auftreten können, wenn die Signale zu laut sind und übersteuern. Ist der Abstand zu groß, sind die Signale hingegen zu schwach um noch richtig erkannt zu werden.

Audiospektrum

Ein größeres Audiospektrum scheint die Übertragung zu verbessern. Durchweg waren die Fehlerraten bei einem 5 kHz Spektrum geringer als bei 2,5 kHz. Allerdings waren bei einer Startfrequenz von 15 kHz auch Geräusche der Übertragung für das menschliche Ohr zu hören.

6.5 Datenübertragung mit QR-Codes

6.5.1 Beobachtung

Die Übertragung mittels QR-Code wurde mit verschiedenen Konfigurationen durchgeführt. Eine Zusammenfassung der aufgetretenden Verluste ist in Abb. 6.6 zu sehen.

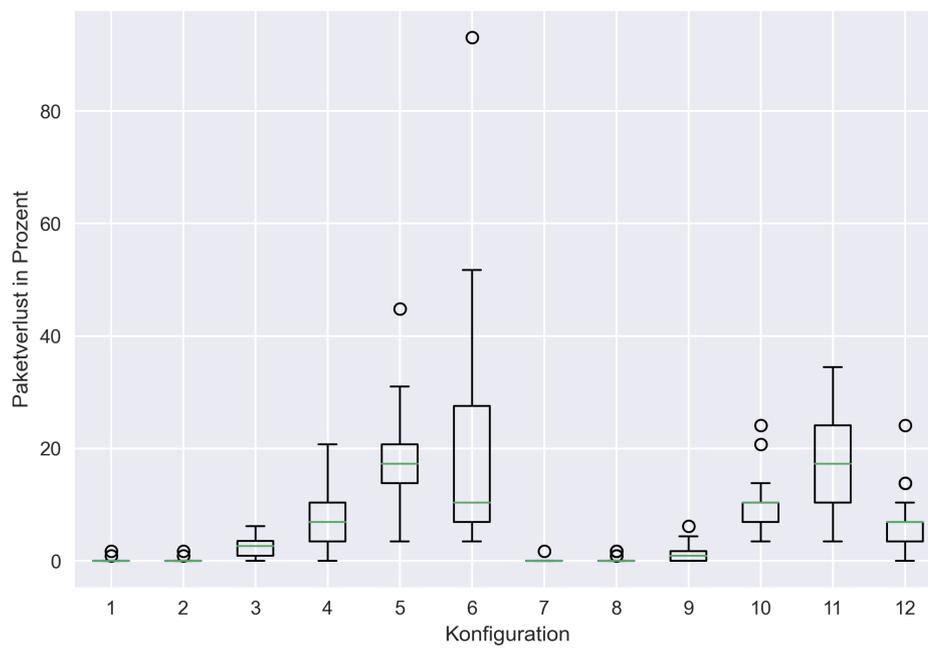


Abbildung 6.6: Frameverluste über die verschiedenen Konfigurationen

Es folgt eine detaillierte Betrachtung der einzelnen Konfigurationen.

Konfiguration 1 - 3: Abstand 15 cm, 25 Zeichen und verschiedene Geschwindigkeiten

Bei 15 cm Abstand und maximal 25 Zeichen, traten bei einer Darstelldauer von 100 ms bei 18 Übertragungen keine Verluste von Frames auf. Bei zwei Übertragungen lag die Verlustrate bei 0,88 %–1,67 %. Bei 75 ms traten bei zwei Übertragungen Fehler von 0,88 %–1,67 % auf und alle anderen waren fehlerfrei. Bei 50 ms war eine Übertragung fehlerfrei und alle anderen hatten Verluste zwischen 0,88 %–6,14 %. Wie in Abb. 6.7 zu sehen.

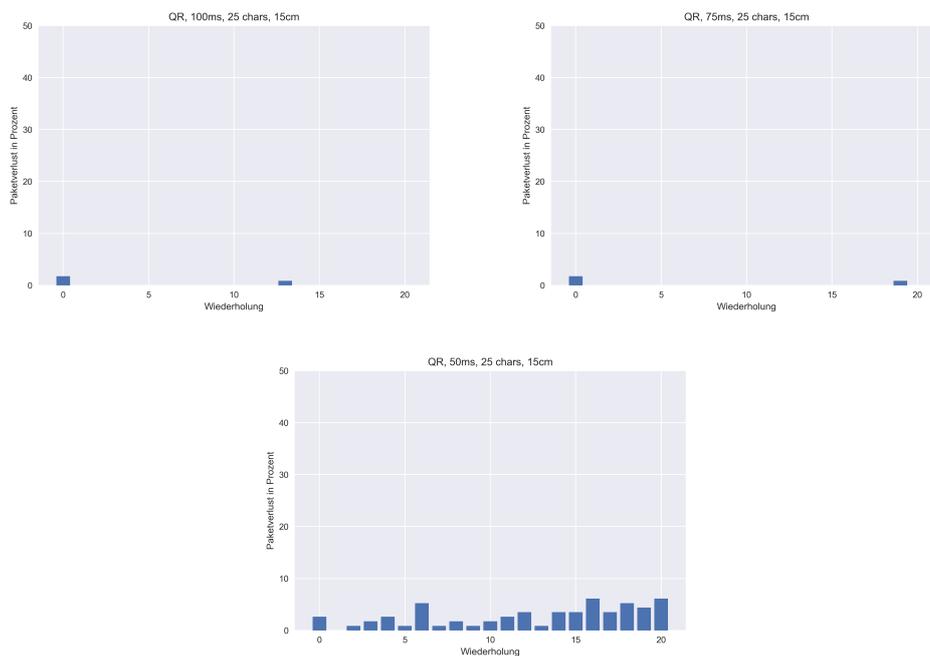


Abbildung 6.7: Verschiedene Geschwindigkeiten bei 15 cm Abstand und maximal 25 Zeichen pro QR-Code

Konfiguration 4 - 6: Abstand 15 cm, 40 Zeichen und verschiedene Geschwindigkeiten

Bei 15 cm Abstand und maximal 40 Zeichen traten bei einer Darstelldauer von 100 ms bei zwei Übertragungen keine Verluste von Frames auf. Bei den anderen lag die Verlustrate bei 0,88 %–20,69 %. Bei 75 ms traten bei zwei Übertragungen Fehler von 3,44 %–44,82 % auf und alle anderen waren fehlerfrei. Bei 50 ms war eine Übertragung fehlerfrei und alle anderen hatten Verluste zwischen 3,44 %–93,10 %, wie in Abb. 6.8 dargestellt.

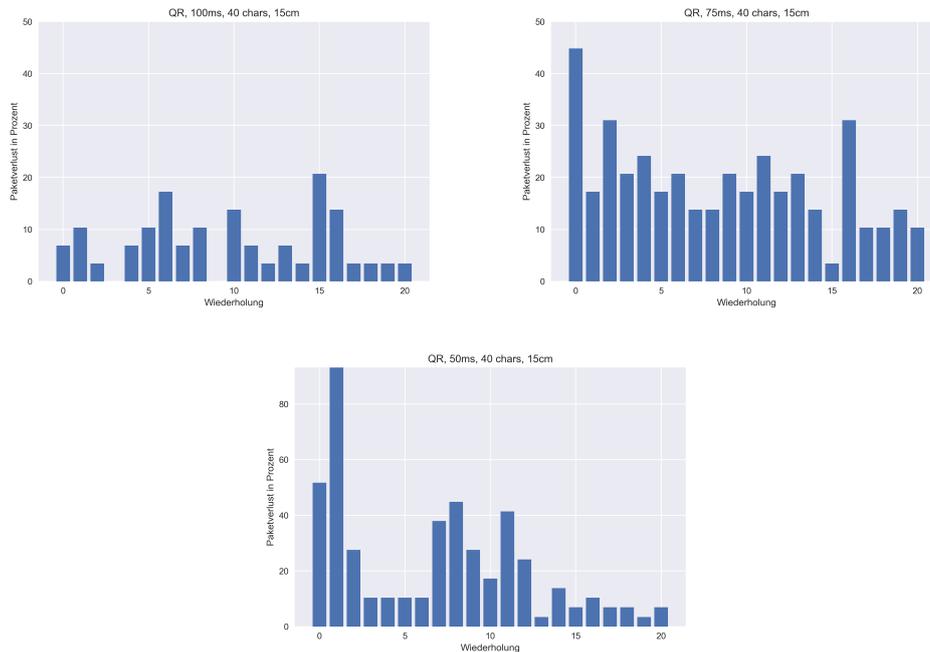


Abbildung 6.8: Verschiedene Geschwindigkeiten bei 15 cm Abstand und maximal 40 Zeichen pro QR-Code

Konfiguration 7 - 9: Abstand 25 cm, 25 Zeichen und verschiedene Geschwindigkeiten

Bei 25 cm Abstand und maximal 25 Zeichen, traten bei einer Darstelldauer von 100 ms, bei einer Übertragung Verluste in Höhe von 1,75% auf. Alle anderen Übertragungen waren fehlerfrei. Bei 75 ms traten bei 5 Übertragungen Fehler von 0,87%–1,75% auf und alle anderen waren fehlerfrei. Bei 50 ms waren zehn Übertragungen fehlerfrei und alle anderen hatten Verluste von 0,87%–6,14%.

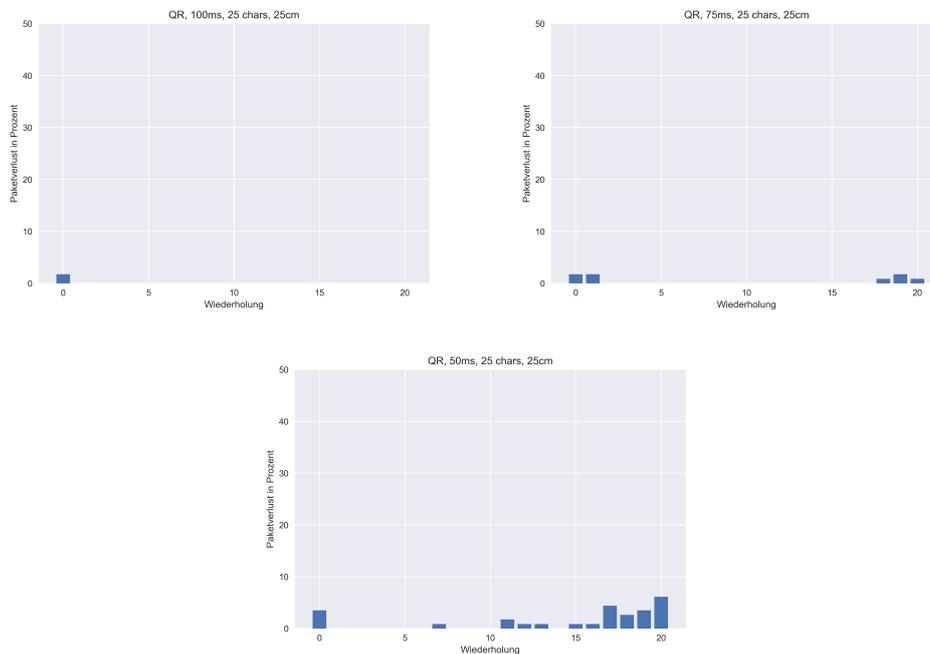


Abbildung 6.9: Verschiedene Geschwindigkeiten bei 25 cm Abstand und maximal 25 Zeichen pro QR-Code

Konfiguration 10 - 12: Abstand 25 cm, 40 Zeichen und verschiedene Geschwindigkeiten

Bei 25 cm Abstand und maximal 40 Zeichen, traten bei einer Darstelldauer von 100 ms Verluste von Frames in Höhe von 3,44 %–24,13 % auf. Alle anderen Übertragungen waren fehlerfrei. Bei 75 ms traten Verluste von 3,4 %–34,48 % auf. Bei 50 ms waren fünf Übertragungen fehlerfrei und alle anderen hatten Verluste von 3,44 %–24,13 %.

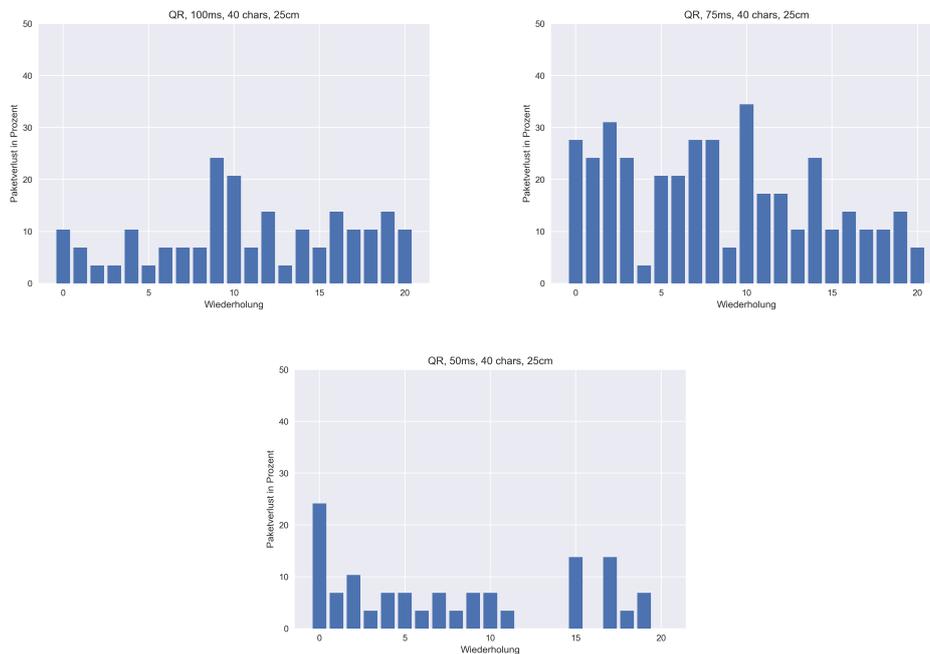


Abbildung 6.10: Verschiedene Geschwindigkeiten bei 25 cm Abstand und maximal 40 Zeichen pro QR-Code

6.5.2 Diskussion

Auch bei der Übertragung mithilfe von QR-Codes schwanken die Ergebnisse, jedoch weniger stark.

Geschwindigkeit

Bei der Geschwindigkeit fällt auf, dass bei der Reduzierung von 100 ms auf 75 ms pro QR-Code die Fehlerrate steigt. Das war zu erwarten, da die Smartphones weniger Zeit zur Verfügung haben, um den Code zu erkennen. Was allerdings auffällt ist, dass bei einer weiteren Reduzierung auf 50 ms die Fehlerrate geringer wird.

Es wird vermutet, dass es einen Zusammenhang der Bildwiederholrate von der Kamera mit der Dauer der QR-Codes besteht. Möglicherweise findet eine Überlagerung der Frequenzen statt, wodurch einzelne Codes nicht gescannt werden können.

Abstand

Bei der Vergrößerung des Abstandes von 15 cm auf 25 cm wurde vorab davon ausgegangen, dass sich die Verlustrate erhöht. Jedoch ist das Gegenteil eingetreten und die Verluste wurden geringer. Die Verbesserung ist allerdings nur gering, abgesehen von der letzten Konfiguration.

Die QR-Codes wurden mit der Front-Kamera abgelesen. Diese Kamera wird von den Herstellern manchmal auch als Selfie-Kamera bezeichnet, da dies ihr primärer Einsatzzweck ist. Vielleicht liegt der optimale Fokuspunkt deshalb weiter weg als 15 cm wodurch die QR-Codes dadurch besser gescannt werden konnten.

Größe des QR-Codes

Die maximale Größe eines QR-Codes pro Datenübertragung hat die Verlustrate der Übertragungen negativ beeinflusst. So war mit 40 Zeichen pro QR-Code beinahe keine Übertragung fehlerfrei und Verlustraten wurden erhöht.

Je mehr Informationen in einem QR-Code codiert werden, um so höher ist die Pixeldichte, da die Fläche des QR-Codes nicht vergrößert wird. Die Kamera bietet bei Videoaufnahmen eine Auflösung von 1920×1080 Pixel. Wird nun die Pixeldichte zu hoch, können nicht mehr alle Daten des Codes gescannt werden. Es wird vermutet, dass dies der Grund für die Verluste ist.

7 Fazit

Ziel dieser Arbeit war es audio- und videobasierte Kommunikationen zu untersuchen und die Schwächen anhand der Verlustrate zu messen. Für diesen Zweck wurde zunächst ein Protokoll entwickelt. Das Protokoll erfüllt die Anforderungen, die Daten auf eine Weise zu übertragen, die eine Speicherung der übertragenen Daten beim Empfänger für eine gute Evaluation erlaubt. Nach diesen Vorgaben wurde ein Traffic Generator entwickelt, welcher in der Lage ist Daten über die Data-over-Sound Bibliothek und via QR-Codes zu übertragen. Anschließend wurden zwei Smartphones in Experimenten gegenüber gestellt und mithilfe des Traffic Generators Daten übertragen.

Die Evaluation der Daten hat ergeben, dass die Übertragungen in Ihren Fehlerraten stark schwanken. Insbesondere bei der Übertragung mit der audiobasierten Data-over-Sound Bibliothek waren diese Schwankungen zu sehen. War ein Übertragungsdurchlauf zuerst fehlerfrei, so kam es vor, dass der nächste Durchlauf Fehlerraten von 50% erreichte. Bei der videobasierten Übertragung mithilfe von QR-Codes waren diese Schwankungen geringer, aber auch zu erkennen.

Trotz sehr hoher Fehlerquoten bei der audiobasierten Übertragung fiel auf, dass diese bei steigender Entfernung ein wenig abnahmen. Bei der videobasierten Übertragung fiel auf, dass die Fehlerrate stark von der Auflösung des QR-Codes abhängig ist. Zu beachten ist auch, dass die gemessenen Werte in optimaler Umgebung entstanden sind. Im Praxiseinsatz muss damit gerechnet werden, dass die Fehlerquoten noch größer ausfallen werden.

Die audiobasierte Übertragung erreichte eine geringere Bandbreite als die videobasierte und besitzt ein größeres Delay, dafür aber sind größere Distanzen mit ihr möglich. Die videobasierte Übertragung erreichte bessere Werte, aber dafür muss eine stabile Sichtverbindung zwischen beiden Geräten existieren. Es wird erwartet, dass Erschütterungen die Verbindung stört. Sollen Daten mit einem geringen Delay und höherer Bandbreite übertragen werden, dann sollte die videobasierte Übertragung gewählt werden. Ist eine

Sichtverbindung allerdings nicht möglich und eine geringere Bandbreite mit höherem Delay akzeptabel, dann sollte die audiobasierte Übertragung mit ihrer größeren Reichweite genutzt werden.

7.1 Ausblick und Verbesserungen

Die Evaluation hat gezeigt, dass die gewählten Ansätze noch Potenzial für Verbesserungen haben.

Die Data-over-Sound Bibliothek hat im Laufe der Arbeit einige Schwächen gezeigt. Es scheint zielführender zu sein, hier eine eigene Lösung zu entwickeln. Hierbei ist es auch interessant andere Ansätze als eine Frequenzumtastung zu evaluieren.

Da bei der videobasierten Lösung mithilfe von QR-Codes Übertragungen auch zufriedenstellend übertragen werden konnten, gilt es hier die Übertragung robuster zu gestalten. Ein starker Faktor für die Verluste von Frames scheint die Auflösung des QR-Codes und in diesem Zusammenhang die Fähigkeiten der Kamera zu sein. Hier scheint es sinnvoll eine Aushandlung zu entwickeln, der die Grenzen der Übertragungen ausmisst.

Allgemein sollte im nächsten Schritt eine bidirektionale Kommunikation entwickelt werden. Eine Kommunikation zwischen Sender und Empfänger würde Möglichkeiten, wie eine variable Geschwindigkeit oder Wiederholungen von Teilen des Datensatzes zu wiederholen, ermöglichen. Es wird weiterhin mit Fehlern in der Übertragung zu rechnen sein, somit ist es hilfreich, wenn der Empfänger beim Sender eine erneute Übertragung von fehlerhaften Übertragungen anfordern kann.

Literaturverzeichnis

- [1] ARNON, Shlomi: *Visible light communication*. Cambridge University Press, 2015
- [2] ASHOK, Ashwin ; JAIN, Shubham ; GRUTESER, Marco ; MANDAYAM, Narayan ; YUAN, Wenjia ; DANA, Kristin: Capacity of pervasive camera based communication under perspective distortions. In: *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2014, S. 112–120
- [3] BANG, Green ; CHUNG, Myoungbeom ; KO, Ilju: Data communication method based on inaudible sound at near field. In: *2016 18th International Conference on Advanced Communication Technology (ICACT)*, 2016, S. 639–642
- [4] BIHLER, Pascal ; IMHOFF, Paul ; CREMERS, Armin B.: SmartGuide – A Smartphone Museum Guide with Ultrasound Control. In: *Procedia Computer Science* 5 (2011), S. 586–592. – URL <https://www.sciencedirect.com/science/article/pii/S1877050911004017>. – The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011) / The 8th International Conference on Mobile Web Information Systems (MobiWIS 2011). – ISSN 1877-0509
- [5] COOLEY, James W. ; TUKEY, John W.: An algorithm for the machine calculation of complex Fourier series. In: *Mathematics of Computation* 19 (1965), Mai, Nr. 90, S. 297–297. – URL <https://doi.org/10.1090/s0025-5718-1965-0178586-1>
- [6] DESHOTELS, Luke: Inaudible Sound as a Covert Channel in Mobile Devices. In: *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. San Diego, CA : USENIX Association, August 2014. – URL <https://www.usenix.org/conference/woot14/workshop-program/presentation/deshotels>
- [7] GEORGE KENNEDY, Bernard D.: *Electronic communication systems*. 4th ed. Tata McGraw-Hill, 1999. – ISBN 9780074636824,0074636820

- [8] HAMMING, R. W.: Error Detecting and Error Correcting Codes. In: *Bell System Technical Journal* 29 (1950), April, Nr. 2, S. 147–160. – URL <https://doi.org/10.1002/j.1538-7305.1950.tb00463.x>
- [9] HU, Wenjun ; GU, Hao ; PU, Qifan: LightSync. In: *Proceedings of the 19th annual international conference on Mobile computing & networking - MobiCom '13*, ACM Press, 2013. – URL <https://doi.org/10.1145/2500423.2500437>
- [10] INSTITUT, Robert K.: *KENNZAHLEN ZUR CORONA-WARN-APP*. 2021. – URL https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/WarnApp/Archiv_Kennzahlen/Kennzahlen_23072021.pdf?__blob=publicationFile. – Zugriffsdatum: 2021-07-24
- [11] LIU, Wenkai ; WANG, Biqi ; LI, Yan ; WU, Menglong: Screen-Camera Communication System Based on Dynamic QR Code. In: *IOP Conference Series: Materials Science and Engineering* 790 (2020), apr, S. 012012. – URL <https://doi.org/10.1088/1757-899x/790/1/012012>
- [12] OZEKI, Kosei ; AOKI, Naofumi ; DOBASHI, Yoshinori: Development of an Inaudible Sound Communications System Based on a Machine Learning Approach. In: *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, 2020, S. 079–083
- [13] REED, I. S. ; SOLOMON, G.: Polynomial Codes Over Certain Finite Fields. In: *Journal of the Society for Industrial and Applied Mathematics* 8 (1960), Juni, Nr. 2, S. 300–304. – URL <https://doi.org/10.1137/0108018>
- [14] SAGOTRA, Rajan ; AGGARWAL, Reena: Visible light communication. In: *International Journal of Engineering Trends and Technology* 4 (2013), Nr. 3, S. 403–405
- [15] STOLERMAN, Ian P. (Hrsg.) ; PRICE, Lawrence H. (Hrsg.): *Fast Fourier Transformation*. S. 676–676. In: STOLERMAN, Ian P. (Hrsg.) ; PRICE, Lawrence H. (Hrsg.): *Encyclopedia of Psychopharmacology*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2015. – URL https://doi.org/10.1007/978-3-642-36172-2_200157. – ISBN 978-3-642-36172-2
- [16] USER:KTIMS, Wikimedia C.: *Bildung eines binären FSK-Signals. Oben: Quelldaten als eine Folge von logisch-1 und logisch-0. Mitte: Unmodulierte Trägerfrequenz Unten: Moduliertes FSK-Signal*. 2006. – URL <https://de.wikipedia.org/>

- [wiki/Frequenzumtastung#/media/Datei:Fsk.svg](#). – Zugriffsdatum: 2021-06-21. – File: LambdaPlaques.jpg
- [17] VAN STEEN, Maarten ; TANENBAUM, Andrew S.: *Distributed systems*. Maarten van Steen Leiden, The Netherlands, 2017
- [18] WANG, Anran ; LI, Zhuoran ; PENG, Chunyi ; SHEN, Guobin ; FANG, Gan ; ZENG, Bing: InFrame++: Achieve Simultaneous Screen-Human Viewing and Hidden Screen-Camera Communication. In: *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. New York, NY, USA : Association for Computing Machinery, 2015 (MobiSys '15), S. 181–195. – URL <https://doi.org/10.1145/2742647.2742652>. – ISBN 9781450334945
- [19] WANG, Tongyu ; HAN, Hao ; WANG, Zijie: FareQR: Fast and Reliable Screen-Camera Transfer System for Mobile Devices using QR Code. In: *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, 2020, S. 364–369
- [20] WHITTAKER, J. M.: The “Fourier” Theory of the Cardinal Function. In: *Proceedings of the Edinburgh Mathematical Society* 1 (1928), Nr. 3, S. 169–176
- [21] YEH, Chien-Hung ; LIU, Yen-Liang ; CHOW, Chi-Wai: Real-time white-light phosphor-LED visible light communication (VLC) with compact size. In: *Optics express* 21 (2013), Nr. 22, S. 26192–26197

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Evaluation einer Audio- und Videobasierten M2M Kommunikation

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort Datum Unterschrift im Original