Institute of Architecture of Application Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

Bluetooth Broadcasting Revisited: Experiments and Comparison to 2009

Marc Grütter

Course of Study: Medieninformatik

Examiner: Dr. I. Georgievski

Supervisor: Dr. I. Georgievski

Commenced: October 17, 2022

Completed: April 17, 2023

Acknowledgements

I would like to express my deepest gratitude to all the individuals who have played a crucial role in bringing this thesis to fruition. Special thanks go to my esteemed supervisor and examiner, Dr. Ilche Georgievski, for his invaluable guidance and feedback throughout this journey. I am also grateful to Prof. Dr. Marco Aiello for facilitating my contact with Joel de Nes and Remko de Jong, who generously shared their original theses with me.

Furthermore, I am indebted to the study participants from the university for their participation and contribution to my research. I am grateful to my oldest sister for her diligent proofreading and editing assistance, as well as to my family and friends for their unwavering support and encouragement. Last but not least, I am immensely grateful to my cat for her constant emotional support and motivation to keep me going.

Abstract

The world of technology is rapidly changing and advancing, including Bluetooth. In 2009 an experimental study at the university of Groningen was conducted to evaluate Bluetooth technology on the premise of broadcasting data to a multitude of devices. The authors especially had a look at the performance and scalability. They concluded that it was suitable for broadcasting data although there are limitations in the technology, such as the limitation of seven simultaneous devices connected to a single dongle (Bluetooth antenna).

In 2023, a review was conducted to examine how this statement has held up and to assess how Bluetooth, with its new development versions, can contribute to broadcasting today. Hence, this thesis aims to investigate the evolution in Bluetooth technology, including changes in its functioning, hardware compatibility, and current market share. Furthermore, experimental studies have been conducted similar to 2009 to compare the performance and scalability with real-life data.

In 2009 the authors used Bluetooth version 2.1 for their experiment, whereas in 2023 version 5.4 was used. Over the years many improvements have been added with each version of Bluetooth. Especially with the better data rates and better reliability of Bluetooth, the technology has great promises. A promising change is the addition of Bluetooth Low Energy (LE), which can be beneficial in the aspect of broadcasting data. While Bluetooth LE has the ability to reach an unlimited number of devices with greater distances using just one dongle, it is important to note that it also has limitations. Specifically, it is only compatible with apps and does not work natively on a phone.

In our experimental studies, it was discovered that Bluetooth technology performed better in terms of data transmission success rates and distance coverage when connecting with compatible devices. However, despite the increasing number of Bluetooth-enabled devices in 2023 compared to 2009, Bluetooth was able to locate fewer devices. This is mainly due to the fact that devices need to be in their Bluetooth settings to be detected, and there are many other non-phone devices, such as TVs, smartwatches, headphones, and more that are also utilizing the software. Additionally, Bluetooth technology is mainly compatible with Android devices using the open standard profile Object Push Profile (OPP). This poses an issue for the many iPhones in the market, as they are not compatible and can only work with an app or their own Airdrop standard.

The conclusion is that Bluetooth broadcasting is possible, yet Bluetooth Classic is not that suitable with all its current limitations. The recommendation for broadcasting data is, that people must have an app where they want to receive data and use Bluetooth LE, since it is better in distance, and has no limitations on the number of devices.

Keywords: Bluetooth, Broadcasting, Experimental Study, Bluetooth Low Energy, market share, Bluetooth history, compatibility, app, scalability

Contents

List of Figures

List of Tables

Acronyms

ACL Asynchronous Connection-Less. 20

ATT Attribute Protocol. 24

BR Basic Rate. 17

DoS Denial of Service. 25

EATT Enhanced Attribute Protocol. 15

EDR Enhanced Data Rate. 15

eSCO enhanced-SCO. 20

FHSS Frequency Hopping Spread Spectrum. 19

GAP Generic Access Profile. 24

GATT Generic Attribute Profile. 24

GOEP General Object Exchange Profile. 22

HCI Host Controller Interface. 18, 19

HS High Speed. 15

IoT Internet of Things. 15

ISM Industrial, Scientific and Medical. 19

L2CAP Logical Link Control and Adaptation Layer Protocol. 20

LE Low Energy. 4, 13, 69

MANETs mobile ad hoc networks. 28

OBEX General Object Exchange. 22

OPP Object Push Profile. 4, 22, 69

PAwR Periodic Advertising with Responses. 28

PSM Protocol Service Multiplexer. 22

RF radio frequency. 19

SCO Synchronous Connection Oriented. 20

SDP Service Discovery Protocol. 21

SIG Bluetooth Special Interest Group. 13

UART Universal Asynchronous Receiver and Transmitter. 19

USB Universal Serial Bus. 19

UUID Universally Unique ID. 21

1 Introduction

Joel de Nes [Joe12] and Remko de Jong [Rem09] collaborated on their research on the viability of Bluetooth broadcasting in 2009. A shortened version of their study can be found in a 2009 publication titled "Bluetooth Broadcasting: How far can we go? An experimental study"[AJN09], which was authored by Prof. Dr. M. Aiello. The broadcast topology is a communication method that enables one-to-many (1:m) device communication, which is ideal for sharing data in the nearby vicinity. [KHC23] They carried out an experiment utilizing Bluetooth version 2.1. While Bluetooth is suited for mobile location-based broadcasting, the study's authors found that technological barriers precluded it from being fully practicable. [Joe12]
This bachelor thesis aims to reevaluate the feasibility of Bluetooth broadcasting using current technology. The study will focus on the technological advancements in Bluetooth and how they impact broadcasting. A series of experiments will be conducted to examine several parameters, including the number of simultaneous connections per module, scalability, device responsiveness, and transmission success rate. Subsequently, the findings will be compared with the results from the previous study. [Joe12][Rem09][AJN09]
Bluetooth technology is commonly found in various mobile devices, such as smartphones, laptops, and wearable devices. The Bluetooth Special Interest Group (SIG) reported in their annual Market Update [SIG23a] that over 4.9 billion devices with Bluetooth capability were shipped in 2022, and this number is projected to increase to 7.6 billion devices by 2027 alone. This can be seen in Figure 1.1. Bluetooth technology allows for wireless connectivity between devices such as headphones, smartwatches, and other wearable technology. The technology is utilized for various purposes such as audio streaming, data transfer, proximity marketing, indoor navigation, and many more. Due to the increasing use of such devices and services, people tend to leave Bluetooth enabled even when not in use. Theoretically, having a large number of potential recipients should increase the likelihood of successfully transmitting messages, which could be advantageous for broadcasting purposes. Bluetooth technology has evolved over the years, with new modes like Bluetooth Low Energy (LE) being introduced [SIG23a].
Chapter 2 will give the reader an introduction to the Bluetooth technology, including its history, applications and challenges. Chapter 3 presents the latest developments in Bluetooth, including the new Bluetooth 5.4 standard and Auracast, as well as three related projects. Chapter 4 goes into the overall architecture of the software project, while Chapter 5 explains how the project was implemented in terms of specific software and hardware. Chapter 6 provides detailed descriptions of the experiments, which are then evaluated and compared with the results from 2009 in Chapter 7. Finally, Chapter 8 concludes with a discussion of the findings and possible future work.

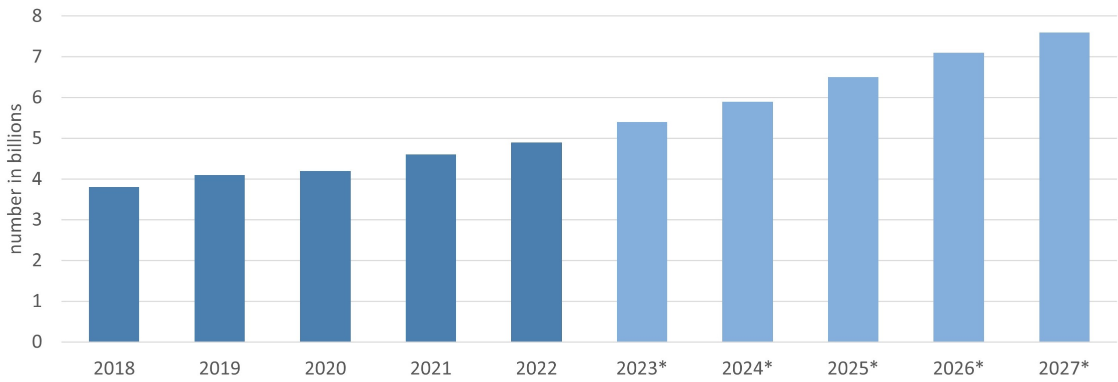


Figure 1.1: Total Annual Device shipments (*estimated). Adapted from [SIG23a]

2 Background Information

Bluetooth is a well-known technology to connect and pair devices, which allows communication between them. This is done without the use of wires and cables. [Moh22]
In this chapter the background with history, use cases, functionality and problems will be described.

2.1 Bluetooth history

The name Bluetooth derives from the two words “blue tooth”, which is the translated nickname of the Danish King Harald Blatand, who ruled in the 10th century. He was able to unite Denmark and Norway. Like him, the technology Bluetooth unites computers and phones. [Moh22]
The technology traces its origins back to 1994 when it was developed by the company Ericsson. This company conceptualized the idea of connecting devices like headphones to a mobile device. Further development was done in the next years to also allow other devices to communicate over this wireless technology. The company wanted to make it an open industry standard and allow further development. So, in February 1998 Ericsson, Nokia, IBM, Toshiba and Intel formed a SIG, which would gain more members over the years. [Moh22] Currently there are over 38,000 members [SIG23b].
The first official version 1.0B was introduced in December 1999. Over the next three years further versions (1.1 / 1.2) got developed to fix identified bugs and improve the speed of the connection. [Sau22]
The original Bluetooth specification was slower than other wireless technologies like Wi-Fi at the time. With Bluetooth 2.0 in 2004, SIG improved the speed by adding an Enhanced Data Rate (EDR). [Moh22]
With the version 2.1 in 2007 the security was improved and Secure Simple Pairing was added, which makes pairing easier. In 2009 Bluetooth 3.0 introduces the possibility to use Wi-Fi to send data with faster speeds. This is called High Speed (HS) Mode. Bluetooth LE was introduced in 2010 with version 4.0 and was further improved with versions 4.1 and 4.2 by adding for example more security and allowing bigger data packets. With Bluetooth 5.0 the newest major version has been released in 2016. [Sau22]
This version is focused on optimizing the technology for Internet of Things (IoT) devices. Compared to version 4.2, numerous features were refined. This included doubling the data transfer speed, boosting the data broadcast capacity by eight times, and more than quadrupling the range of Bluetooth compared to the previous version. In 2019, the ability to determine the direction of a signal was added with version 5.1 of Bluetooth. This feature is called Angle of Arrival and requires multiple Antennas. [Moh22]
Version 5.2 added LE Audio support and improved Enhanced Attribute Protocol (EATT) for multiple applications simultaneously working with Bluetooth. [Sau22]
Data processing and reliability is improved in Bluetooth 5.3 since its release in 2021. Version 5.4

released in 2023 is the most current version at the time of the creation of this paper. [SIG23d] Table 2.1 shows an overview of all major Bluetooth versions and their transfer speeds and range.

Factors	Bluetooth 1	Bluetooth 2	Bluetooth 3	Bluetooth 4	Bluetooth 5
Speed	732.2 kb/s to 1 Mbps	2.1 Mbps	24 Mbps (via Wi-Fi)	1 Mbps (LE)	2 Mbps (LE)
		EDR		25 Mbps (EDR)	50 Mbps (EDR)
Range	10 meters (33 feet)	30 meters (100 feet)	30 meters (100 feet)	60 meters (200 feet)	240 meters (800 feet)

Table 2.1: Speed and range of each Bluetooth version [AG21]

2.2 Statistics and applications of Bluetooth Technology

According to the Bluetooth SIG platform, around two billion devices equipped with Bluetooth are shipped per year. This number is estimated to stay constant in the upcoming years, whereas peripheral devices are predicted to grow even more over the next years. This can be seen in Figure 2.1. In the forecast for 2027, peripheral devices are estimated to even make up around 74% of all Bluetooth enabled devices shipped in that year. An example for platform devices are phones, tablets and PCs. [SIG23a]

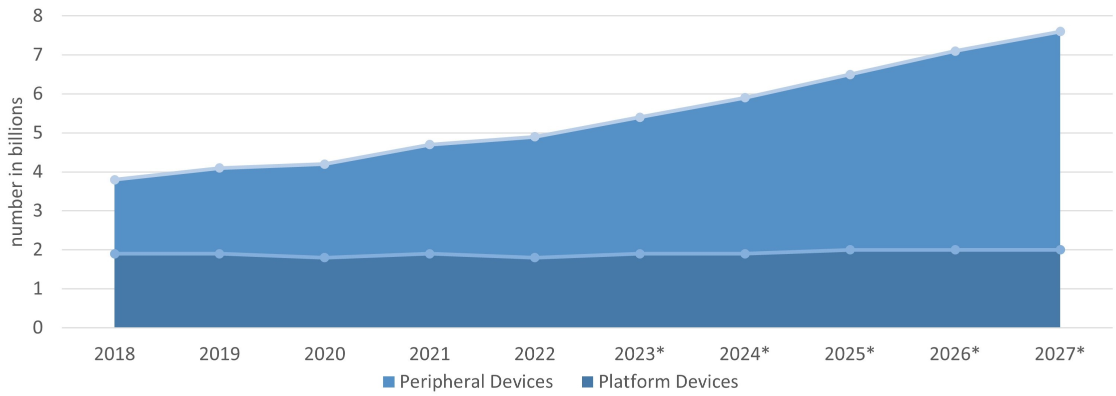


Figure 2.1: Device shipment of platform and peripheral devices (*estimated). Adapted from [SIG23a]

Figure 2.2 illustrates the growth of individual Bluetooth modes, showcasing the number of shipped devices supporting each mode per year. All platform devices support both Classic and Bluetooth LE modes, which is useful for developers to know that the individual device supports both of them. In addition, the are many Bluetooth LE single mode peripheral devices gaining market share. In total, by 2027 it is expected that 97% of all shipped devices will support Bluetooth LE. According to the Bluetooth SIG, peripherals are projected to be the main driver of Bluetooth device shipments' growth in the years that follow. [SIG23a]

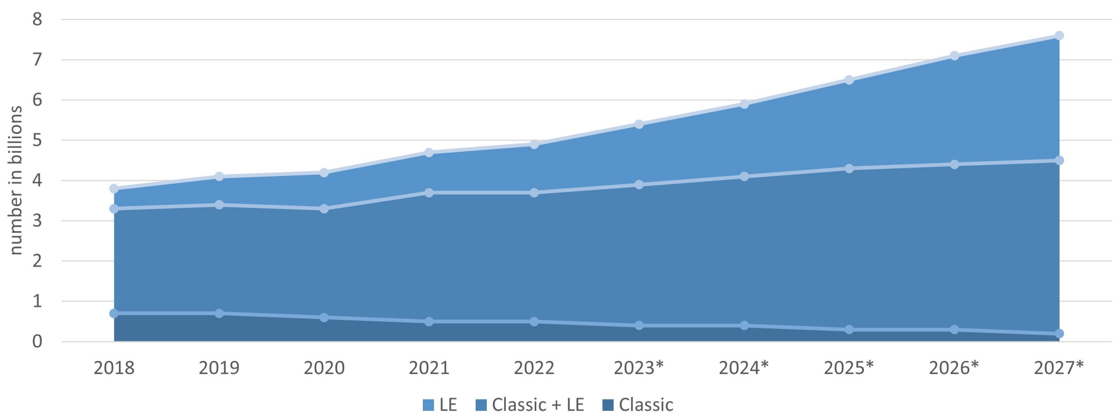


Figure 2.2: Device shipment for each version of Bluetooth (*estimated). Adapted from [SIG23a]

Bluetooth technology can be used for different purposes. It can be categorized into four distinct categories: audio streaming, data transfer, location services and device networks.
The first category is Audio Streaming. It enables devices like headphones or speakers to be used wirelessly indoors, outdoors, and even in the car. The second category is data transfer. This is used to make IoT devices, health or fitness trackers, toys or PC peripherals like keyboards or mice work wirelessly as well. The third category is Location Services, which is especially useful to navigate indoors, track objects or people, locate items with a tag, or as a digital key. The fourth category is Device Networks. It is used for monitoring (e.g., temperature, humidity), control (lighting), and electronic shelf labels, in for example, a retail environment. For instance, all this can be used to optimize the energy consumption or improve the life span of a system. Often in this category, a multitude of devices are communicating with each other. [SIG23a]

2.2.1 Bluetooth Classic

Bluetooth Classic can either use the Basic Rate (BR) or EDR. It was developed to allow short range point-to-point wireless communication. It is reliable for continuous communication as audio and data transfer. Classic mode is not originally meant to be used for one-to-many devices communications. [KHC23]

2.2.2 Bluetooth LE

Bluetooth LE is as the name suggests, a less energy consuming version of Bluetooth to wirelessly communicate with devices. This lower power means that it can increase battery life of the device. Often small IoT devices prefer to have a lower power consumption. Bluetooth LE has slower data rates, which is not suitable for sending big data files. But for small data it is sufficient.

Bluetooth LE can set up in three different network configurations. These configurations are point to point (one-to-one device), broadcast (one-to-many devices), and mesh (many-to-many devices). [KHC23]

2.3 Functional principle of Bluetooth stacks

Every Bluetooth device has Bluetooth radio and software to enable the use of the technology. [Moh22] Each mode of Bluetooth has a different stack, which allows an application to use the technology. The differences can be seen in Figure 2.3. All three versions (Classic, Dual and LE) can be separated into two parts. These are the controller and host. The lower layers of the stack are part of the controller and are responsible for managing operations that require a timely execution. The upper layers are located in the host and are tasked with executing more complex and high-level operations. The Host Controller Interface (HCI) acts as an intermediary in communicating between the two components, allowing the host to control the lower layers by transmitting commands while the controller transfers data to the upper layers. The application interfaces with the topmost layers of the host component situated in the stack. [CPST22]

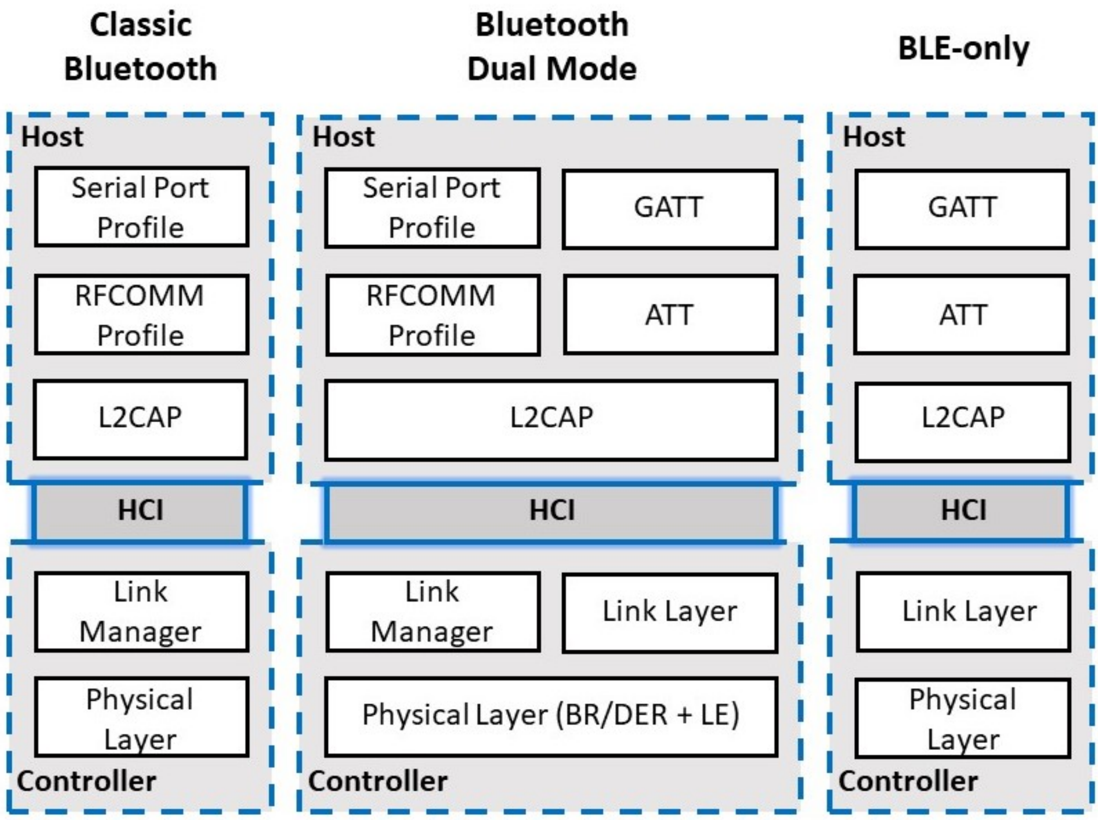


Figure 2.3: Simplified Bluetooth stack for the three versions. Adapted from [YPMN19]

2.3.1 Joint layers of Dual mode (Classic + LE)

The Dual Mode stack is the combined stack of Bluetooth Classic and LE. Some layers exist in both versions of Bluetooth. These layers are described in the following paragraphs.

Physical layer

The capabilities and properties of the technology itself are defined in the physical layer. Both Bluetooth Classic and LE are operating in the Industrial, Scientific and Medical (ISM) band from 2.402 GHz to 2.480 GHz. This frequency is also used in Wi-Fi and other wireless communication protocols. [CPST22] Even if they are in the same band, it does not mean they are compatible with each other. Bluetooth Classic and LE use Frequency Hopping Spread Spectrum (FHSS) [Sau22], to lower the chance of disturbing other traffic on the same frequency band. FHSS means that if a connection is established the Bluetooth frequency is changed with every packet that is sent to a different device. [CPST22]
Bluetooth Classic separates the ISM band into 79 radio frequency (RF) channels with 1 MHz spacing. With multiple channels, it is possible to operate multiple Bluetooth connections at the same time in the vicinity of each other. The minimum requirement is that each network of multiple Bluetooth enabled devices, called a piconet, has its individual hopping sequence. Each piconet has one master and up to seven slave devices. Every device can be a master, but often the device that establishes the connection takes this role. The master device decides when data is allowed to be sent. [Sau22] The supported data rates range from 1 Mb/s to 3 Mb/s. [KHC23]
The ISM band is separated in Bluetooth LE as well. The band splits up into 40 RF channels with a spacing of 2 MHz. Three equally distributed channels of those RF channels are primary advertising channels. These are responsible for the first advertising and connection. The remaining channels can be used for other tasks. [Sau22] Data rates between 125 Kbps and 2 Mbps can be supported by Bluetooth LE. [KHC23]
Both Bluetooth Classic and LE have three power levels for different devices. Class 3 is 1mW and typically used in mobile phones. Class 2 reaches 2.5mW and Class 1 up to 100mW of power. Type 1 is commonly used for PCs since energy consumption is not a critical factor. Varying levels of power consumption can result in different transmission ranges. [KHC23]

Host Controller Interface (HCI)

HCI is the interface between the controller and the host. As an interface it allows the host to send commands to the lower layers and the controller to forward data to the host. The host primarily runs on the main CPU of the device. The HCI is implemented in different ways, depending on the hardware. The controller can be integrated into the main CPU or externally on a separate chip. The integrated version is often found in smaller devices. [CPST22]
There are two interfaces declared in the standard. The Universal Serial Bus (USB) interface and the Universal Asynchronous Receiver and Transmitter (UART) interface. Small device interfaces are often implemented as UART and the bigger devices like PCs as USB. The commands and data packets are handled differently depending on which interface is used. In the UART, packets are identified with the header. USB is identified with different endpoints. [KHC23]

Logical Link Control and Adaptation Layer Protocol (L2CAP)

L2CAP is above the HCI and mostly relevant for the Classic Bluetooth version. The main function of this layer is multiplexing. [CPST22] Multiplexing makes it possible to support multiple simultaneous data streams over a single physical Bluetooth connection. Additionally, it can assign the data to the specific protocols and channels in the upper layers. This layer has also the important task of segmenting the data packets into the correct size for the lower layers to process. [KHC23]

2.3.2 Bluetooth Classic stack

The following layers are only present in the Bluetooth Classic version. In Figure 2.4 shows a more detailed version of the Bluetooth Classic stack.

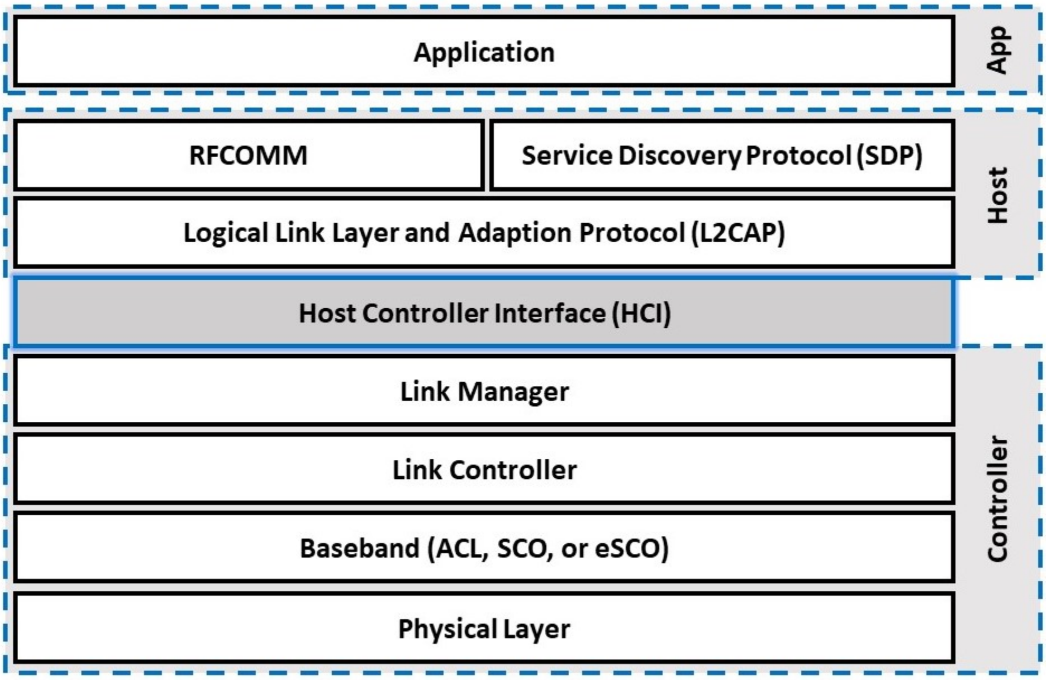


Figure 2.4: More detailed depiction of the Bluetooth Classic stack. Adapted from [Sau22]

Baseband

The baseband layer is on top of the physical layer. This layer frames the data packets for the data transfer. There are three sorts of frame types for transmitting data: Asynchronous Connection-Less (ACL), Synchronous Connection Oriented (SCO), and enhanced-SCO (eSCO). ACL is asynchronous and has no guarantee for the bandwidth, which is not ideal for real time data. SCO is synchronous and has a fixed speed of 64 kbit/s, which is better suited for real time scenarios. However, it has no error detection for data, which can propagate in the higher layers. eSCO is also synchronous and can be set up with a higher data rate (up to 288 kbit/s in both directions) at the beginning of a connection. eSCO data packets also have a checksum for error detection, which allows for resending data if other data packets have not arrived yet. There are other packet types in the baseband used for establishing and maintaining a connection, which will not be further discussed as they do not contribute in the understanding of this thesis. [Sau22]

Link Controller

The Link Controller layer builds on the baseband layer and is responsible for establishing, maintaining, and terminating connections. A state model is used at this layer to manage the connections. A device is in the inquiry state if it wants to search for new devices. The state change is ordered by the higher layers. Devices in the inquiry state send ID packets on two different frequencies in every time slot. A device must also periodically switch to an inquiry scan state and search for ID packets on alternating frequencies to allow connection requests from unknown devices. To conserve power, devices do not constantly search for ID packets. The Bluetooth standard recommends a scan time of 11.25ms per 1.28 s interval, which has a 90% chance of finding all devices in the vicinity in around 10s. The paging procedure is similar to inquiry. It sends swiftly many ID packets on different frequencies. And the packet contains the device id of the other device instead of a general address. This allows devices to communicate only with paired devices whose device addresses are known. The inquiry scan function can be turned off to prevent contact with unknown devices. The connection active state is achieved after a successful paging procedure. In this state data can be exchanged. There are three additional connected states that are defined in the Bluetooth standard for devices that want to save energy:

• Connection-Hold - master and slave choose a time in which they are both inactive,
• Connection-Sniff - master and slave choose interval and time in which they are actively listening for packets and
• Sniff-Subrating - similar to Connection-Sniff but with an additional timeout if no data is received. [Sau22]

Link Manager

The previously discussed Link Controller layer sends and receives data packets depending on the connection state. The task of the Link Manager is to establish, maintain, terminate, and configure connections. Possible configurations are, for example, turning on the authentication or encryption. [Sau22]

Service Discovery Protocol (SDP)

Theoretically, communication could start with all layers presented above already. However, Bluetooth devices have a multitude of possible services, which could not be addressed this way yet. Therefore, all services a device supports are saved in a service database with their own Universally Unique ID (UUID). A service is also called a profile in the application layer. The SDP enables a device to query another device for all the services they support and how they can be addressed. This is called SDP ServiceSearchRequest and uses the UUID of the Public Browse Group (0x1002). It is also possible to query a specific service with its own UUID to determine if it is supported and how it

can be addressed, if available. [Sau22]
If the way to address a device with its specific service is already known, it is possible to skip the query. It is important to note that Bluetooth services can change the address during the runtime. Connection parameters, for example, the RFCOMM channel number are flexible. [Sau22]

RFCOMM

The L2CAP layer is used to multiplex numerous data streams over a physical connection in Bluetooth. The Service Database is an application that is addressed via the L2CAP Protocol Service Multiplexer (PSM) 0x0001, and other services can be addressed through other PSMs. Some services use a shared layer called RFCOMM, which provides virtual serial interfaces to services and simplifies data transmission. RFCOMM is addressed via PSM 0x0003. [Sau22]
The RFCOMM layer simulates not only the send and receive lines but also the status lines of a complete UART module. Other services, such as the General Object Exchange (OBEX) file transfer service, also use the RFCOMM layer. Different RFCOMM channel numbers can be used to select which service to address during connection setup. The correct channel number can be obtained by querying the Service Database. [Sau22]

Bluetooth Profiles

Bluetooth is a versatile technology that works across various devices made by different manufacturers. To ensure seamless communication between them, Bluetooth profiles are specified in the Bluetooth standard. A profile describes how a client (slave) and a server (master) must behave while communicating with each other. The server and client side in Bluetooth applications usually have different responsibilities. Both devices must support the profile to work properly. [Sau22]
The General Object Exchange Profile (GOEP) is used as a foundation for other profiles, such as the Object Push Profile (OPP). GOEP defines the connection between two devices during the transfer of objects. The connection established between two devices using these profiles is temporary and lasts only for the duration of transmitting one or more consecutive objects, after which it is immediately terminated. [Sau22]
The OPP is commonly used for transferring calendar events, address book entries, and files between devices. It does not support directory operations or file deletion to simplify the process. The received object is stored in a buffer and only copied to the device after confirmation by the user. The OPP requires calendar events and address book entries to have the same format to ensure compatibility between devices. File name extensions can be used to identify other objects that are transferred, like images as a .jpeg file. The OPP can be used in most devices without authorization and encryption. [Sau22]

Security modes

In Bluetooth, different security modes exist:
• Security mode 1 does not require authentication or encryption and is suitable for simple data transfer between devices.
• Security mode 2 allows users to determine whether authentication, encryption, and authorization

are necessary for a connection, and can be configured for each service individually.
• Security mode 3 automatically establishes authentication and encryption for each connection.
• Security mode 4 was specified for the new pairing mechanism Secure Simple Pairing in Bluetooth 2.1. This mode allows a service to choose the security category.
[Sau22]

2.3.3 Bluetooth LE stack

The following layers are only present in the Bluetooth LE version. Figure 2.5 shows more detailed version of the Bluetooth LE stack.

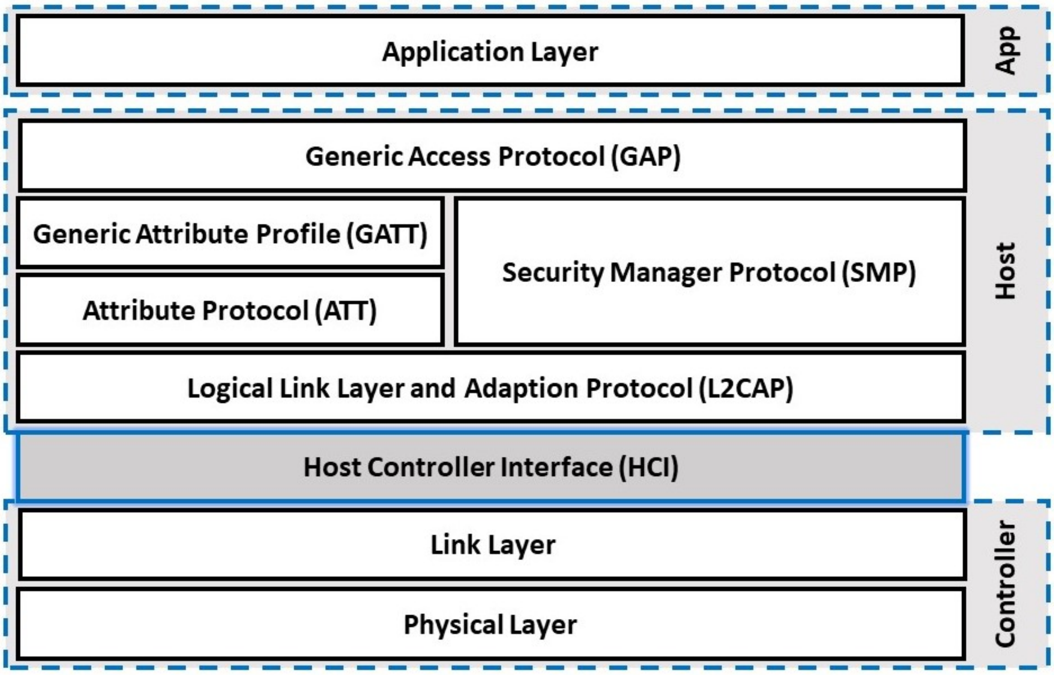


Figure 2.5: More detailed depiction of the Bluetooth LE stack. Adapted from [CPST22]

Link Layer

The link layer is responsible for tasks such as handling protocol data units, performing time-critical functions like encryption and decryption, and maintaining a unique device identifier called Bluetooth device address. The link layer also uses a state-machine to manage different connection states, including standby, advertising, scanning, initiating, connection, synchronization, and isochronous broadcasting. [CPST22]
Connection-oriented and connection-less communication is supported in Bluetooth LE. In the connection-less scenario, the advertiser becomes a broadcaster and sends non-connectable and non-scannable packets, which nearby devices can listen to but not connect to. [CPST22]

Security manager

The security manager layer is responsible for all security aspects and can be divided into two parts: a cryptographic toolbox and methods for pairing and exchanging keys. The toolbox for cryptography includes operations for computing hash values and creating keys. Table 2.2 three different security modes can be seen. Each mode is subdivided into levels. The third mode is available since Bluetooth 5.2. [CPST22]

Mode	Level	Pairing security	Message security
1	1	No pairing	No security
	2	Unauthenticated legacy	Encryption + MAC
	3	Authenticated legacy	Encryption + MAC
	4	Authenticated LE-SC	Encryption + MAC
2	1	unauthenticated	GATTlayer data signing
	2	Authenticated	GATTlayer data signing
3	1	No pairing	No security
	2	Unauthenticated	Encryption + MAC
	3	Authenticated	Encryption + MAC

Table 2.2: Overview of Bluetooth LE security modes [CPST22]

Attribute Protocol (ATT) and Generic Attribute Profile (GATT)

The ATT in Bluetooth LE enables a device to expose its attributes and associated values to another device through a client-server model. Attributes consist of an attribute type (specified by a UUID), an attribute handle (a 16-bit number), and access permissions (like read and write permissions). [CPST22]
The GATT builds upon ATT to define a data abstraction model for attributes. Single attributes are called characteristics. A service is a set of characteristics. A profile consists of multiple services. [CPST22]
Both central and peripheral devices in a Bluetooth LE connection can act as both ATT servers and clients simultaneously. Clients can read, write, and discover attributes, while the server can indicate and notify attributes and their values to the clients. [CPST22]

Generic Access Profile (GAP)

GAP is a profile that explains how Bluetooth LE devices can find, connect, and secure each other. It uses features from other parts of the stack to make sure devices from different makers can work together. GAP has four roles for Bluetooth LE devices: broadcaster, observer, peripheral, and central. The GAP tells which features of other parts of the stack are required and which of them are optional. [CPST22]

2.3.4 Known problems of Bluetooth

There are problems with the Bluetooth technology which will be further discussed in the following subchapters.

Interference and range

Bluetooth technology allows devices to connect in the close vicinity, but the actual range depends on various factors such as environment, interference, and transmission power. [Intel22]
Class 3 devices can reach up to 10m without interference. This class can send through a maximum of one wall only. Class 1 devices can reach devices through multiple walls and even reach distances up to 100m with their higher transmission power. However, the maximum possible range depends on the device with the lower class. [Sau22]
Bluetooth technology operates in the same frequency band as other wireless technologies such as Wi-Fi and ZigBee. [YPMN19] Devices that operate on that frequency are also "older PCs, tablets, phones, and Wi-Fi networks, as well as microwaves and baby monitors" which can interfere with Bluetooth. [Intel22] However, comprehensive tests have shown no significant interference between Bluetooth LE and Wi-Fi, but Bluetooth Classic and ZigBee occasionally collide with Bluetooth LE. Additionally, studies have shown that the probability of failed Bluetooth LE connections and transmissions increases significantly in complex and dense wireless environments. Using frequency hopping in the physical layer has been proven to be a successful method to prevent interference in most instances. [YPMN19]

Security

Bluetooth is a widely used wireless technology that is known for being reliable and secure. However, with billions of Bluetooth devices in use, they are exposed to various threats. Adapting and extending security measures is a necessary step to address these emerging threats. [Moh22]
Security threats can be categorized into three major types. The first type is a disclosure threat, which involves the unauthorized access of information by an eavesdropper who is not permitted to access it. The second type is an integrity threat, which occurs when the information is intentionally modified to deceive the recipient. The third type is a Denial of Service (DoS) threat, where authorized users are prevented from accessing a service by making it unavailable or severely limiting its availability. [ZSB19]

Bluetooth LE compatibility

In 2016, Google introduced its Nearby feature at the Google I/O conference. This feature allowed users to get notifications even without an installed app. Google used a technology called Eddystone-URL, which could send notifications to users via the Physical Web feature of the Chrome browser or the Nearby Notification feature on Android devices. iOS users needed to download Chrome for iOS and set it to the Today Widget to use this feature. [AH20]
The technology has been utilized for example to inform users about available free Wi-Fi in the vicinity, offer museum guides, and display transit schedules at bus stops. Nearby notifications have

been discontinued since December 6, 2018. This decision was made because a growing number of notifications were deemed locally irrelevant and “spammy”. Sending Bluetooth LE notifications is now only possible with an app. [AnD18]

3 State of the art

In this chapter, some related projects are described that are relevant to this thesis and could help future work to be done in the field. Additionally, information on the newest Bluetooth 5.4 standard and Bluetooth Auracast will be shown.

3.1 Relevant Projects

In the following chapters, three projects which are relevant to big Bluetooth networks are outlined.

3.1.1 First project - “A survey on Bluetooth multi-hop networks”

This paper discusses the history and development of Bluetooth technology. It is especially focused on its potential for ad-hoc networking. The introduction of Bluetooth LE Mesh Profile was a significant development for multi-hop networks. It was not optimal for all use cases that involve multi-hop networks. [TK19]
Bluetooth multi-hop networks are a particular kind of wireless ad-hoc network where the transfer of data between nodes depends on intermediate nodes forwarding packets. Each node in the network needs to decide whether a received packet should be forwarded or is intended for itself. [TK19]
The project reviews the progress made on Bluetooth multi-hop networks in both Classic and LE versions of Bluetooth over the past two decades, focusing on the demands for real-world implementation. Two types of Bluetooth multi-hop networks are identified: connected and connection-less networks. Both have their own applications and technical feasibility. However, there have been only four successful Bluetooth multi-hop network deployments. One of the four successful deployments is integrated into a photovoltaic power plant. This is the only one integrated into a real-world deployment. [TK19]

3.1.2 Second project - “An AI and Equation based Modelling Solution for Multiple Bluetooth Connections with Single Master Device”

The paper discusses the limitations of Bluetooth networks and the inconvenience of multitasking over a single master Bluetooth device when multiple devices with similar functionalities are connected. This case can lead to the possibility of a connection conflict. [DG22]
The authors proposed a mathematical calculation-based modelling solution to group connected devices based on their Bluetooth functionality, which is categorized into five groups: music/calls (both), calls, music, file transfer, and miscellaneous. The data is then transmitted using frequency hopping and time division techniques. The solution involves assigning multiplication factors to each cluster according to their priorities. This enables multiple Bluetooth connections with

higher priority functionalities to be assigned larger weighted timeslots. The solution also provides suggestions to the user for the best device based on data transfer rate, signal clarity and data type. [DG22]

3.1.3 Third project - "A Loop Reduction Method Utilizing Public Device Addresses of Master Terminals in Connection Procedures of Multiple Piconets for Bluetooth MANETs"

The article discusses the current research being done on Bluetooth-based mobile ad hoc networks (MANETs), particularly in the context of disaster information propagation systems. The existing method of establishing Bluetooth MANETs involves the formation of multiple piconets, which can easily form loops in high-density terminal environments. MANETs use both Bluetooth Classic and LE. [ISKK21]
The authors propose a method to reduce the occurrences of loops by regulating the way slave terminals establish connections. They evaluated their suggested method through simulation experiments and found that it decreases the number of loops and transmissions of data packets. It keeps the duration of data packet distribution unchanged compared to the existing methods. [ISKK21]

3.2 New Bluetooth 5.4 Standard with Auracast and bidirectional Bluetooth LE in 2023

The introduction of the latest Bluetooth standard 5.4, along with Auracast, occurred only recently in early 2023. [SIG23d][SIG23c]

3.2.1 Bluetooth standard 5.4

Prior to the introduction of Periodic Advertising with Responses (PAwR), bidirectional communication in a large-scale one-to-many topology was not possible with Bluetooth LE. The new communication method now enables such communication. In contrast, the old specifications of advertising data only allowed for unidirectional communication. The new method is energy-efficient and offers a secure way of broadcasting data in advertising packets. PAwR provides periodic and deterministic advertising. It is possible for observing devices to synchronize with this deterministic schedule. Compared to the previous Bluetooth LE specifications, advertising packets are now eight times larger, with a capacity of 254 bytes instead of the old 37 bytes. Additionally, these new data packets can be chained six times up to a size of 1650 bytes. [SIG23d]
Bluetooth Core Specification version 5.4 also introduces Encrypted Advertising Data, LE GATT Security Levels Characteristic, and Advertising Coding Selection features, which enhance the user experience and provide control over important parameters. [SIG23d]

3.2.2 Auracast

Auracast broadcast audio is a new feature of Bluetooth technology that allows for an unlimited number of nearby audio receivers to pick up audio broadcasts from a transmitter. This capability is expected to become widely available in new and existing devices in 2023, with Android already announcing support for it. Auracast broadcast can be transmitted from either a dedicated Auracast transmitter or other compatible devices such as a smartphone. Multi-language support is possible by broadcasting separate streams for each language. The range of an Auracast broadcast depends on the implementation (transmit power and antenna design) and can reach up to 30,000 square feet ($2800m^2$). While currently focused on audio, Auracast broadcast may have the potential for transmitting video in the future. [SIG23c]

3.2.3 Future development of the Bluetooth standard

The Bluetooth SIG is developing new specifications for Bluetooth technology that includes higher frequency bands and data rates. The new specification will expand the frequency band to 6GHz, which could increase the performance of Bluetooth. [SIG23a]

4 Project architecture

The aim of the thesis is to evaluate the effectiveness of sending data to multiple devices using the latest Bluetooth technology. However, prior to conducting the research, an appropriate solution for the study must be devised.
The project architecture is illustrated in the Figure 4.1. The design involves a single computer serving as a database that supports multiple senders (antennas). These antennas can communicate with multiple phones and transmit data from a centralized database. The use of multiple antennas in a project can greatly enhance the number of devices that can be reached, based on the findings from the previous project in 2009 [Rem09]. The centralized database enables efficient data management and retrieval, making it easy to deploy.

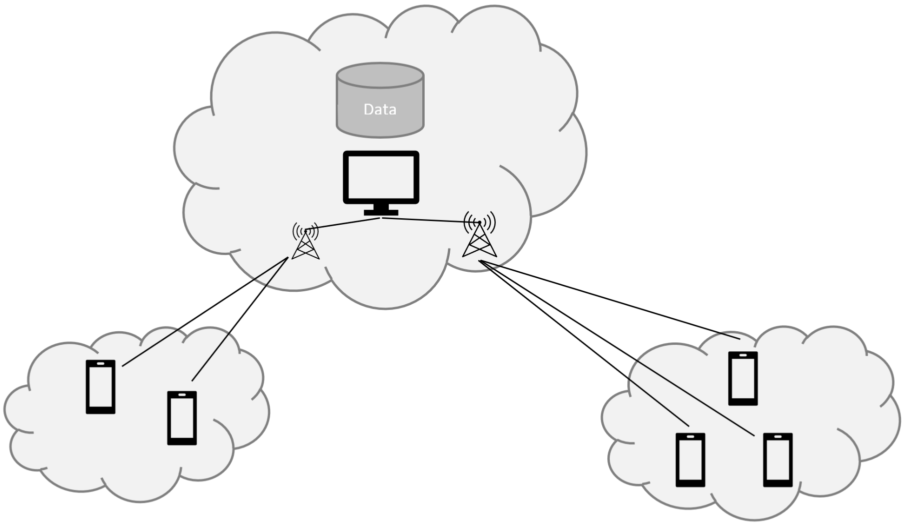


Figure 4.1: Architecture of the project

To extend the reach of the project and increase its scalability, horizontal scaling can be implemented by incorporating multiple computers with their respective antennas, all linked to a centralized data storage system. By doing so, a larger number of devices can potentially be reached, particularly when the devices are located out of their transmitting range. Figure 4.2 illustrates this concept, although it is important to note that the extended architecture will not be tested in the current research.

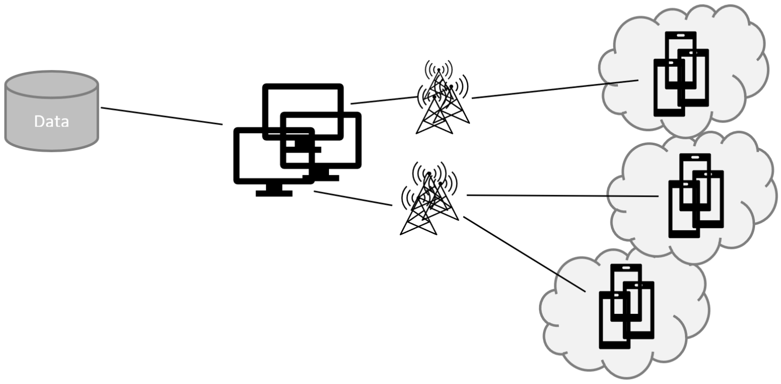


Figure 4.2: Extended architecture of the project

5 Project implementation

The following chapter provides an overview of the decisions that were made regarding the hardware and software selection to achieve the goal of broadcasting messages to multiple Bluetooth enabled devices. Furthermore, the software project design will be explained, and an overview of the components and classes of the software are provided.

5.1 Hardware and software decisions

In order to successfully use Bluetooth to send messages to multiple Bluetooth enabled devices, various decisions need to be made in terms of hardware, software and other related components.

5.1.1 Operating System

The previous experiments from 2009 [Rem09] showed that there were limitations on the number of Bluetooth dongles that could be used, which led to further research. It was discovered that Windows and Mac still only support one dongle. Linux is capable of supporting multiple, which is needed if more than seven devices are to be connected per Bluetooth Classic, as stated in Section 2.3.1. For the experiment, the possibility of using Android was considered, which is based on Linux [AnD23a] and has mobility advantages. But it seems not to support any dongles due to the inability to change the Bluetooth manager [AnD23b], making it impossible to run the tests on a mobile phone. Therefore, Ubuntu (version 22.10) was selected, as it is currently the most popular Linux distribution with a 31.9% market share in website hosting [TL23]. The distribution is also better documented, making it easier to troubleshoot.

5.1.2 Hardware

To test the system in multiple locations, mobile hardware was necessary. As Android did not support multiple dongles, a phone was not feasible. The options were between a Razer Blade 15 2018 laptop and a Raspberry Pi 3B+. Both Systems are compatible with Ubuntu. After testing, it was found that the Raspberry Pi often freezes up and has problems when using more than two dongles on one system. Therefore, the more stable and faster laptop was chosen, which also has more USB ports to support multiple dongles.
Two 4-Port USB hub 3.0 hubs from Beikell were also acquired for the experiment. The hubs enable connecting multiple Bluetooth dongles to the laptop. This is essential since at least as many dongles as the last experiment from 2009 were required, which needed a minimum of six. Before selecting the Bluetooth dongles, it was important to check their compatibility with Linux. According to the analysis, most Bluetooth dongles are marketed only to support Windows, which is not surprising

given Windows' 73% market share compared to Linux's 2% [TL23].
After considering multiple options, the final choices are shown in Table 5.1 below and were tested in the experiment. These included two long-range dongles which are generally not common in the market and two of the most popular dongles that are also supported by Linux and comply with the 5.0 standard.
During testing, it was discovered that all four adapters contained the same Bluetooth chip (Realtek RTL8761B chipset). However, only the TP-Links and Asus dongles supported SimplePairing mode, which allows Bluetooth devices to connect without a passcode. After considering the results, we chose to use the Jameila dongles for their superior range in detecting devices, and the TP-Link dongles for sending due to their support for SimplePairing mode and better distance than Asus. The Jameila dongles were immediately active upon connection to the laptop, while TP-Link dongles required manual activation using hciconfig. If multiple TP-Link dongles were used on one system, only one would be active, and the rest would need to be manually activated.

Picture	Brand	Name	Bluetooth version	Price
	TP-link	UB500 Nano	5.0	11.99€
	Asus	USB-BT500	5.0	14.99€
	Jaimela	Bluetooth Adapter Long Distance	5.1	6.99€
	ZEXMTE	Bluetooth Adapter Long Distance	5.1	18.99€

Table 5.1: Overview of the tested Bluetooth dongles. The photos are from our own archive.

5.1.3 Programming language and libraries

During the analysis, a decision had to be made between Java and Python as the programming languages for Bluetooth, as a majority of Bluetooth libraries are written in these languages. The following libraries were considered: BlueCove [BC08], native Python sockets [PSF17], Pybluez [AlH19], Bluetooth-manager [VM18] and tinyb [TT18]. Upon further observation most of the available Bluetooth libraries had not been updated in years.

Library	Language
BlueCove	Java
native Python sockets	Python
Pybluez	Python
bluetooth-manager	Java
tinyb	Java, C++

Table 5.2: List of Bluetooth libraries

After careful consideration, BlueCove was chosen since it is a Java library with good performance, is suitable for using multiple adapters and has good documentation and code examples [BC08]. Although using native sockets e.g. from python [PSF17] was possible, it would have required more time to fully adapt the code.
For the development environment experiment it was decided to use IntelliJ IDE as it allowed to save logs directly to a text file without the need for an external database. To ensure easy processing of the logs, they were formated accordingly. This made it possible to save and use the data for future evaluation. This approach also allowed quick changes to the code, without having to make significant modifications to an existing database if something changed during the experiments.
Figure 5.1 shows the Bluetooth Classic stack in combination with the BlueCove Library [BC08] that is used in the project.

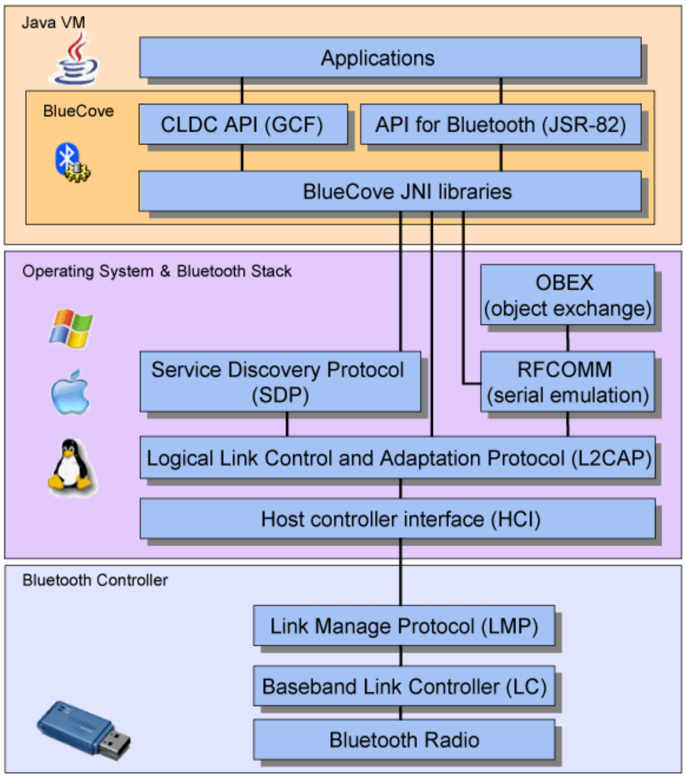


Figure 5.1: Bluetooth Classic stack with BlueCove Library [BC08]

5.2 Components of the Software Project

The software architecture is depicted in Figure 5.2, illustrating how the various components interact with each other. The system logs important steps, which are valuable for evaluation purposes. The sequence order of each step is presented in Appendix A. In the following subchapters, detailed explanations of each component will be provided.

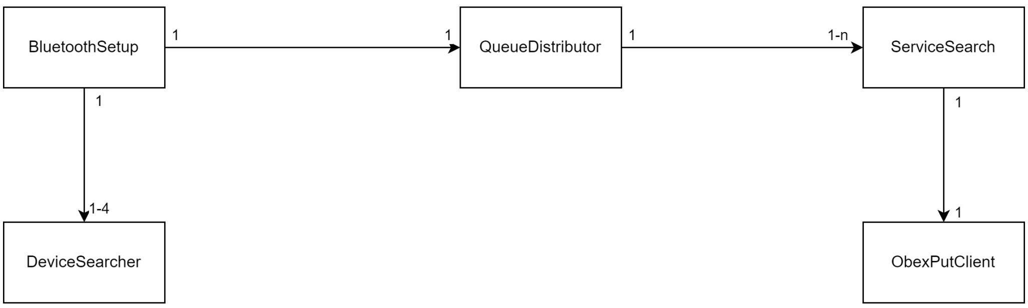


Figure 5.2: Component diagram of the software

5.2.1 BluetoothSetup.java

The main file of the project is BluetoothSetup.java, which is responsible for setting up the project according to the individual requirements of each experiment. The details of how the experiments are set up will be explained in Chapter 6. It is important to note that each experiment uses a different number of device searchers and phone service searchers, which send data if they find any device.
When the experiment round is selected, the software initializes the right number of Bluetooth dongles and assigns them to QueueDistributor.java. Next, the required number of DeviceSearcher.java threads are initialized and started on the designated Bluetooth dongles. If multiple DeviceSearcher.java threads are needed, they will be started at different times as far as possible from each other, to ensure they don't find the same devices at the same time in a discovery cycle. Each discovery cycle lasts for 11 seconds.
For example, if two DeviceSearcher.java threads are required, the first one is activated initially, followed by the second one 5.5 seconds later. The discovery cycle time is divided by the number of DeviceSearcher.java threads (i.e., 11/2s = 5.5s).

5.2.2 DeviceSearcher.java

The DeviceSearcher.java thread continuously searches for Bluetooth activated devices. If a device is found, it compares the Bluetooth device's Bluetooth address with a HashMap that includes all previously found devices with the number of tries. The maximum number of tries for discovering Bluetooth devices in the DeviceSearcher.java thread is set to three. This is because there may be some devices that are always discoverable but do not have the desired service, such as televisions. Attempting to constantly connect to them could consume excessive resources, and potentially overload the Bluetooth dongle.
If a device is not found in the HashMap, the new device is added to the map with number of tries set to one. Additionally, the status is added indicating whether the device is queued, in progress or has already been sent a message. Finally, it adds the device to a queue for further processing.
If the device found is already known, the thread checks whether it has already been marked as sent, queued or in progress and is saved in the variable sent_Queued_InProgress.
If it has not been marked as the above mentioned statuses and the number of tries to connect is less than three, the device is added to a queue and marked as queued. The number of tries is also incremented by one.
However, if the device has already been marked as sent, queued or in progress, or the number of tries is greater than or equal to three, the found device is ignored.

5.2.3 QueueDistributor.java

The QueueDistributor.java is a separate thread that continuously polls the queue from the DeviceSearcher.java threads for any new devices. It distributes the devices fairly and evenly among the assigned Bluetooth dongles from the BluetoothSetup.java in a round-robin style. A ServiceSearch.java thread is initiated as soon as a device is assigned to a dongle.

5.2.4 ServiceSearch.java

The ServiceSearch Thread is an essential component of the system that is designated to one Bluetooth enabled device. It performs the search for the OPP service on that device which allows for the transfer of data (see Section 2.3.2). If no service is found, it is marked as false in the sent_Queued_InProgress Map from the DeviceSearcher.java class, allowing it to be searched and processed again. Additionally, if there are any issues during the search, it resets the map entry to false and the number of tries to the previous value. However, if the service search is successful, the message is set in binary and the ObexPutClient.java is initiated.

5.2.5 ObexPutClient.java

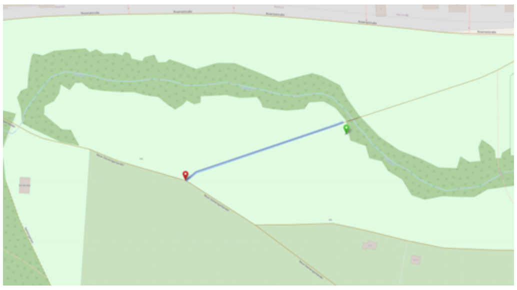
The ObexPutClient.java is responsible for establishing a connection to the phone and sending the message to the right service. It prepares the message, including all necessary headers, and then attempts to send the data. If the transmission is successful, the connection is closed. However, if it fails, the sent_Queued_InProgress Map entry is reset to false, indicating that the entire process needs to be retried.

6 Experimental setup

The experiments conducted in this thesis aim to compare the differences and changes between the current project and a previous project from Nes and de Jong conducted in 2009 [Rem09]. To achieve this, the experiments followed the setup described in the previous project. However, tests using Bluetooth LE were added, a newer technology that was unavailable in 2009 (see Section 2.1). The Bluetooth LE experiment is less detailed, because as already described in Section 2.3.4, Bluetooth LE requires an app to be used on phones. A description of each experimentation setup is provided in the following chapters.
All tests, except for the Bluetooth LE test, were conducted using the software project presented in Chapter 5. For the Bluetooth LE test, a built-in function of the Linux operating system was used, specifically the bluetoothctl function.

6.1 Range test

The aim of the first experiment was to determine how far Bluetooth-enabled devices can detect and communicate with each other. There are many factors that can affect the effectiveness of Bluetooth, such as obstacles like walls, windows, trees or other signals (see Section 2.3.4). To minimize the influence of external factors on the experiments, tests were conducted outside on an open field. This environment allowed the reduction of potential sources of interference that could impact the results.



(a) [OSM23]

(b) The photo is from our own archive.

Figure 6.1: Map view and a photo of the test location

6.1.1 Different brands of dongles

There are different Bluetooth dongles with different antennas and quality assurances. According to the product specifications, the long-range dongles can communicate up to 100m, while the other two dongles in the test were limited to a range of 30-40 m. The reach of communication is tested to see if these specifications hold up in practice.
To validate the range, tests were conducted to determine the maximum distance at which messages could still be successfully received and detected between two devices. For this purpose, a personal Samsung Galaxy S21 Ultra smartphone was used as testing device with a specific Bluetooth address. The software was set up to send messages as often as it found the specified device, eliminating the need to reset the software each time to measure distance. For this experiment the limit of three tries was canceled.
To conduct the experiments on the Bluetooth dongles, each test was divided into two parts: discovery and message reception. For the discovery test, the dongle in question was used as a DeviceSearcher and a TP-Link dongle as the ServiceSearcher and sender. In the message reception test, a Jameila long range dongle was used as the DeviceSearcher, while the dongle to be tested served as the ServiceSearcher and sender.

6.1.2 Bluetooth Classic vs LE

In addition to testing the range of Bluetooth dongles, the performance between Bluetooth Classic and LE was investigated. As only the broadcasting feature was in scope for the experiment, each dongle was tested individually. This only requires sending data and therefore does not need a separate device searching dongle.

6.2 Reception test

Since the initial project in 2009 [Rem09], mobile phones have undergone significant technological advancements. Therefore, it is essential to investigate how these advancements impact display of messages or data on phones.

6.3 Test with small group

In order to further investigate the performance of Bluetooth dongles, a small group test was conducted at the University of Stuttgart. The test involved 28 students, each with a Bluetooth enabled phone, who were required to stay within a range of 20m and remain stationary during the test to allow for maximum connection testing. The test consisted of four rounds, each lasting approximately one minute. To ensure better comparison with the project conducted in 2009 [Rem09], the same number of dongles was used in all rounds. The first round is done with one dongle for searching devices (DeviceSearcher.java) and one for service searching and sending messages (ServiceSearch.java and ObexPutClient.java). The second round involved an additional DeviceSearcher, while the third had three DeviceSearcher. Lastly, in round four was conducted with an equal amount of DeviceSearchers and ServiceSeachers, that is two searchers each. In Table 6.1, the number of dongles used in each round is summarized.

Round	Nr. of DeviceSeracher	Nr. of ServiceSearch
1	1	1
2	2	1
3	3	1
4	2	2

Table 6.1: Number of dongles for each round (short test)

In this study, TP-Links dongles were used for all tests, as it was discovered in a small-scale test that the long-range dongles did not support simple pairing mode, which is necessary for connecting to a device without entering a PIN (see Section 5.1.2). This ensures faster establishment of connections and data transfer. Furthermore, in the small-scale test, it was unnecessary to have long-range dongles since the students were within 20m of reach to the dongles.
To understand how the messages are handled on different phones, the students were asked to complete a questionnaire, providing details about their phone brand and model, the time of receiving a message, and the method of reception. See Appendix B for more details on the questionnaire.
The information gathered from this small test is used in the following tests.

6.3.1 Responsiveness test from questionnaire

This test aims to evaluate the speed of multiple Bluetooth-enabled devices in range and identify any limitations that may arise. This is crucial for broadcasting purposes, as the goal is to reach as many devices as possible within a short amount of time. It is also important to consider that most Bluetooth-enabled devices are mobile and may move out of range during the discovery process, leading to less devices being sent messages. Therefore, this test will focus on the number of devices discovered, the number of attempted message transmissions and the actual number of successful message receptions within a brief period. The impact of the number of dongles assigned for searching and sending messages will be investigated as well.

6.3.2 Responsiveness test from log

For the test, the differences between the answers provided in the questionnaire and the log of the program need to be determined. It is important to investigate these potential differences to ensure the accuracy of the collected data and to identify any discrepancies that may exist. By comparing the questionnaire responses with the log data, a better understanding of the responsiveness of the system can be gained and it can be determined whether there are any issues that need to be addressed.

6.4 Endurance test

In order to simulate a real-world scenario and determine the number of devices that could be found, a longer test was conducted for multiple hours. The test was performed three times. This test was done in 2009 [Rem09] as well, so it was interesting to see the difference in a real-life situation. The test location was the Milaneo shopping center, a popular spot in Stuttgart. The test was done in the food court, which is located in the center of the mall. As evident from Figure 6.2, the Bluetooth coverage from the top of the food court extends to all floors, making it accessible to a potentially large number of people.

Figure 6.2: View of the Milaneo food court from the top floor. The photo is from our own archive.

In all rounds of testing, the Jameila Long range dongles were used to discover devices, as simple pairing mode is not required for this task. However, for sending messages, TP-Links was the preferred dongle, as simple pairing mode was necessary and random people could not be instructed to enter a specific PIN.
The first test was conducted on Tuesday, February 28th, 2023, from 11:30h to 14:30h, during the time when most people were present in the shopping center. Two discovery dongles and three delivery dongles were used. The second test was conducted on a Saturday, March 18th, 2023, from 11:30h to 15:30h, which is the day of the week with the highest foot traffic. This time, two discovery dongles and four delivery dongles were used. The final test was also conducted on a Saturday, March 18th, 2023, but from 10:00h to 13:00h, and four discovery dongles and four delivery dongles were used. The summary of the number of dongles used for each round is shown in Table 6.2.

Round	Nr. of DeviceSeracher	Nr. of ServiceSearch
1	2	3
2	2	4
3	4	4

Table 6.2: Number of dongles for each round (long test)

6.5 Simultaneous connections test

The theoretical limitation of simultaneous connection per dongle are seven active connections. Since Linux was used for the experiment, it is possible to increase the number of dongles. The maximum should increase by seven additional device connections for each dongle added.
To test the simultaneous connections, the data from both the short test and the endurance long test was analyzed. The question of whether a maximum number of simultaneous connections is achieved, depends on the number of devices present at the same time.

7 Results and comparison

The presentation of results is an essential aspect of this study, and in order to provide a clear and structured overview of the findings, the results will be presented in the order of the experiments outlined in Chapter 6. Additionally, the results are compared with the data from the 2009 study [Rem09].

7.1 Range Test

7.1.1 Different brands of dongles

The range test was conducted to evaluate the performance of the dongles and to determine if any differences exist between the different brands. The results are listed in Table 7.1.

Brand	**Discovery**	**Message received**
TP-link	~120m	~110m
Asus	~60m	~55m
Jameila	~130m	~120m
ZEXMTE	~120m	~120m

Table 7.1: Results of the range test with different brands

Although the specifications listed for each device/dongle did not suggest significant differences in their range, the range test revealed that they were able to reach further distances. The discovery range and range from receiving a message were almost the same, with only a few meters difference. It is worth noting that obstacles such as people or trees in direct line of sight can significantly reduce or completely cut off the signal.
Comparing the results from Table C.1 shows that the Bluetooth dongles can reach farther distances than expected. However, only the Linksys Dongle from 2009 [Rem09] was able to match the range of the dongles in this study, but only regarding the discovery distance of around 110m. Although most of the dongles in this study were able to send messages successfully up to around 120m, the results for the 2009 experiment [Rem09] were significantly shorter, with a maximum distance of only 46m and a minimum distance of 10m.
Based on the results in the range test, TP-Link, Jameila, and Zextme dongles performed the best. As such, TP-Link and Jameila dongles were selected for the rest of the experiments, as mentioned in Section 5.1.2. The TP-Link dongles were used for simple pairing mode, while the Jameila dongles were chosen for their high range and affordable price.

7.1.2 Bluetooth Classic vs LE

In the second part of the experiment, the focus of testing was on the range of the two selected dongles, TP-Link and Jameila, in combination with Bluetooth LE. As Bluetooth LE behaves differently than Bluetooth Classic, this test aimed to observe how the dongles perform with this technology. Table 7.2 presents the results, and interestingly, the TP-Link exhibited the same performance as in the Bluetooth Classic mode, which is not typical as Bluetooth LE should allow for longer distances. On the other hand, the Jameila outperformed the TP-Link, with a maximum range of up to 210m, almost double the distance. This improvement in range is expected from the specifications of Bluetooth.

Brand	**Distance**
TP-link	~110m
Jameila	~210m

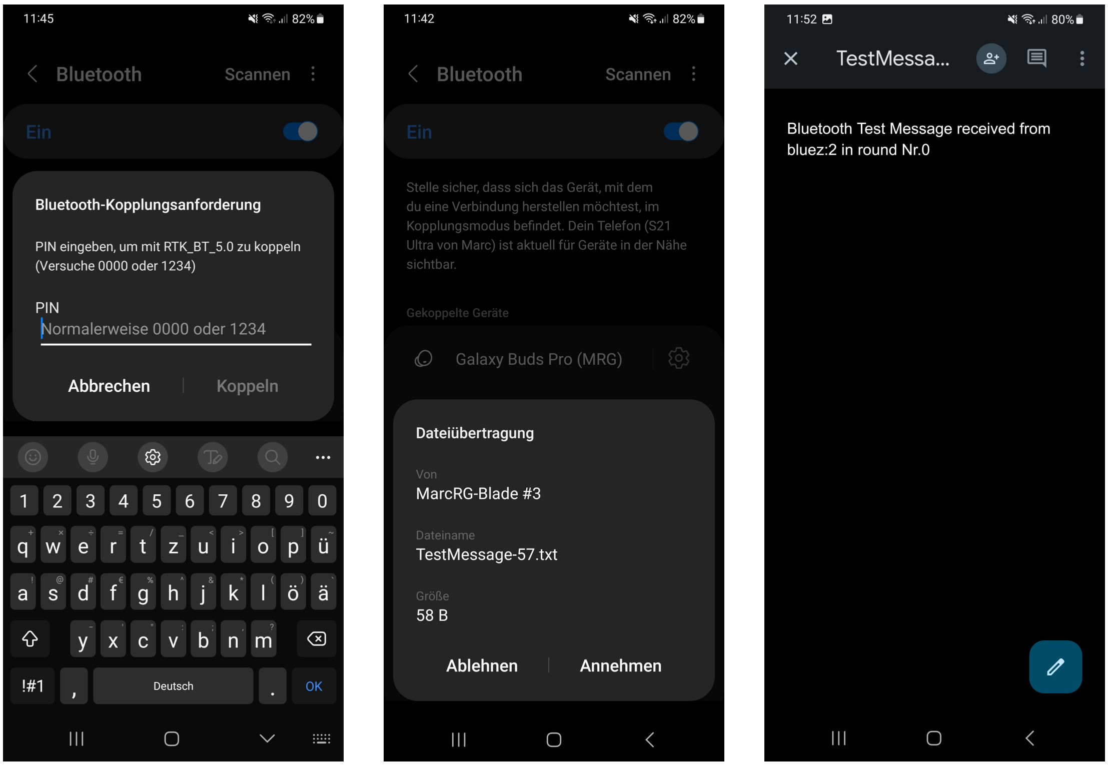
Table 7.2: Results of the range test with Bluetooth LE

7.2 Reception test

In this section, a reception test was conducted and screenshots of the process on a Samsung S21 Ultra running Android 13 were captured. This is depicted in Figure 7.1.
With the increasing size of device screens in recent years, it has become easier to read all the information displayed during the test. Two versions were presented depending on the way they were received. The first version displayed a PIN request, which led to the second version where the user could either accept or decline the message. This first version occurred if the device did not support Secure Simple Pairing, while the second version was used if Simple Pairing mode was enabled. The received message could be viewed in various apps depending on the data type being transmitted. Figure C.1 shows photos from older devices that received message in the study of 2009 [Rem09].

7.3 Results with small group

In accordance with the experiment setup, a test was conducted with a small group of 28 students at the University of Stuttgart. Each student possessed a Bluetooth-enabled phone and remained stationary at a distance no greater than 20m throughout the duration of the test. They were instructed to stay in the Bluetooth settings.
The default setting for Android devices is to not allow other Bluetooth-enabled devices to discover them. However, users can choose to enable discoverability manually for a brief period of time by accessing their system settings. [AnD21]



(a) Pin request **(b)** Message accept or decline **(c)** Display message

Figure 7.1: Screenshots of Samsung phone in the different prompts and display message. The photos are from our own archive.

7.3.1 Responsiveness test from questionnaire

The following data is based on the responses collected from the questionnaire. Figure 7.2 presents the percentage of messages received per round within each time slot, providing a visual representation of the data.

The absolute numbers of the test are presented in Table 7.3, which is divided into each round and the total of all rounds. It is also sorted by time slots, ranging from under 10s to under 30s or under 60s. The table indicates the number of devices that received a message and how many did not. The number in parentheses is one device, that had received the message, but could not find the data when they accepted it.

In general, it can be observed that only a few devices were able to successfully send a message under 10 or 30s. Most messages were sent in the last 30s to the end of each round. The number of devices that did not receive any message was approximately twice the number of devices that received a message.

The second and third rounds were the most successful, as they had more device searchers and could theoretically find all devices faster. However, the last round had the highest number of people who did not receive a message, which could be attributed to several reasons.

Upon analyzing Figure 7.3 which displays the number of messages received per brand in each round, it was found during the test that Apple devices did not receive any messages. This is due to the fact that OPP is not listed in the supported Bluetooth profiles on their support page [Apple19].

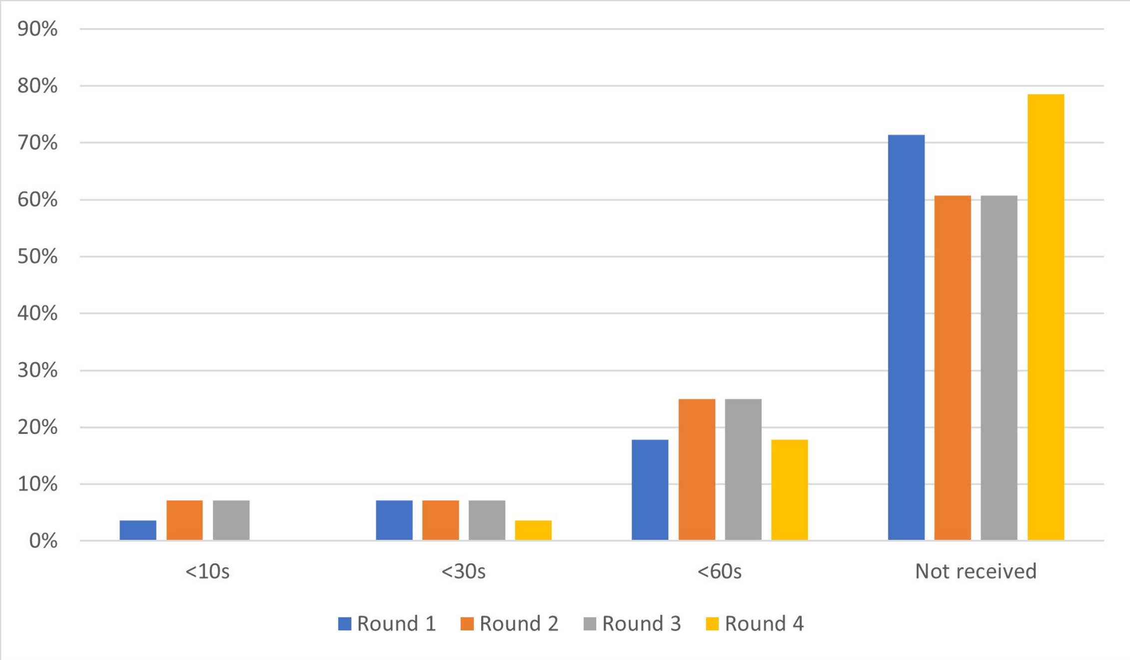


Figure 7.2: Percentage of messages received per round

Round	< 10s	< 30s	< 60s	Received	Not received
1	1	2	5	8	20
2	2	2	6 (1)	11	17
3	2	2	7	11	17
4	0	1	5	6	22
total	5	7	23 (1)	36	76

Table 7.3: Results from questionnaire data

As a result, Apple devices were unable to receive any messages. Instead, Apple uses Airdrop for their ecosystem, which utilizes Bluetooth LE and Wi-Fi to transmit data to devices that are in the near vicinity. Officially, this technology is only compatible with Apple products and is not designed to work with devices from other manufacturers. [Apple21]
All other devices seen in the Figure are running Android operating system, which does support the Bluetooth profile OPP. In theory all Android devices can be found. Yet, the questionnaire revealed that two Android devices, namely the Redmi Note 5 and OnePlus Nord CE, did not receive any messages during the experiment (see Table 7.3).
In total there are nine different phone brands participating in the experiment, but only two different operating systems are present, namely iOS and Android. In 2009 there were only six phone brands involved in the study, as shown in the Figure C.3.

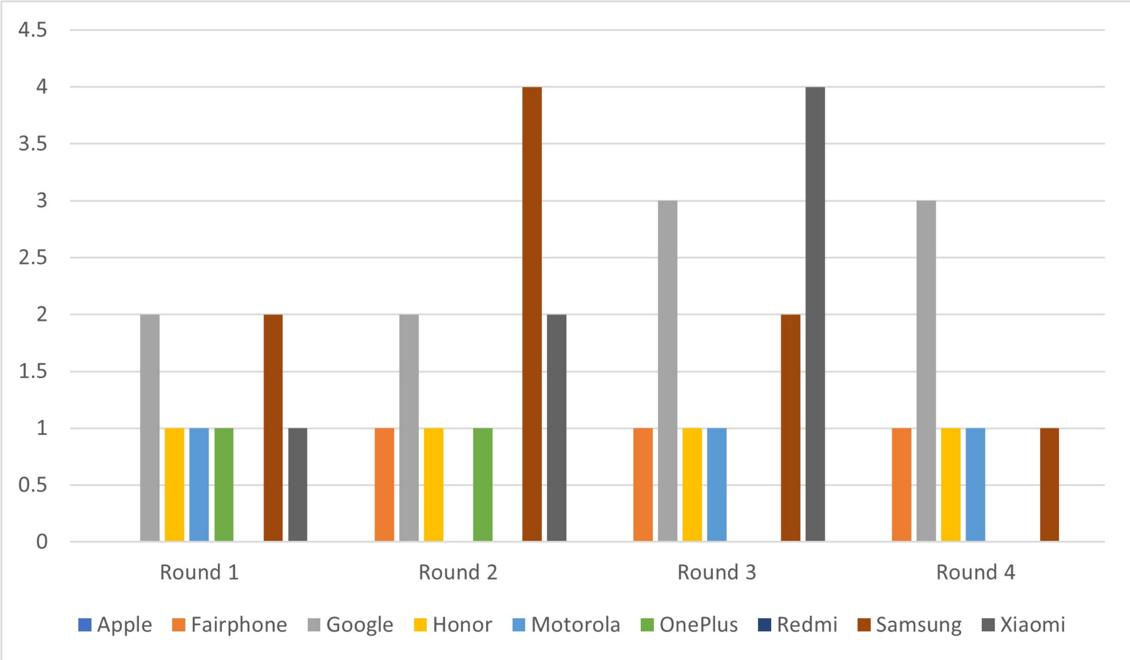


Figure 7.3: Number of messages received per brand

Looking at Figure 7.4, where the difference between how many devices got a message was compared, the rate of not receiving a message seems to be lower with 24.32% in 2023 compared to 32.43% in 2009. In 2009, only nine of the 37 devices did not receive a single message. Out of the nine, seven were phones that were present in the questionnaire did reach a message, so they were capable of receiving a message. Only two were Apple devices. In 2023 we have a way higher percentage of Apple phones. Ten out of 28 were Apple iPhones. TThis is around 35.71% of tested phones, which is almost the same as the market share in Germany, with 37.93and 61.16% Android phones[SC23b]. Worldwide, iOS has 27.11% and Android 72.26% of the market share, so Apple devices are still very present in the market [SC23c]. This suggests that the numbers in the current experiment are reasonable. Only two other phones did not receive a message.

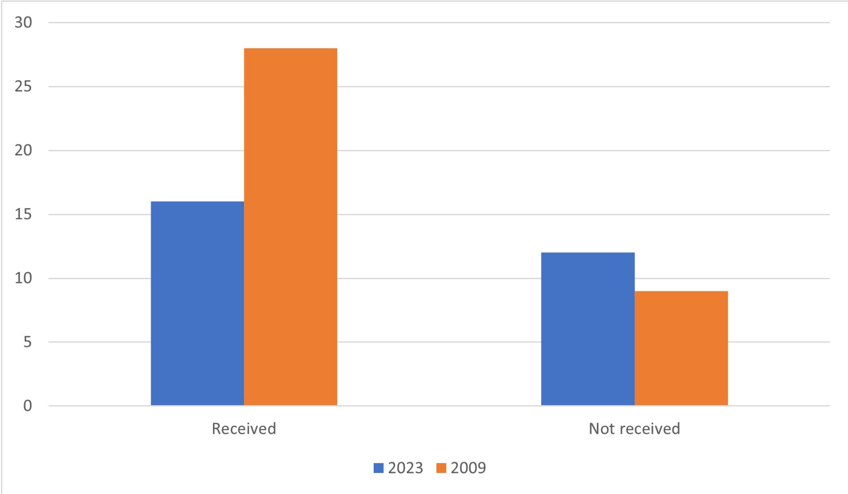


Figure 7.4: Devices that received at least once a message in a testround. Data from [Rem09] for the year 2009.

Finally, the number of times a device received a message can be compared between the two experiments. These numbers are depicted in Figure 7.5. In 2009, most devices were able to receive at least one message, and there were no devices that received more than four messages. However, in 2023, there were two devices that received two messages. Nevertheless, in both experiments, the majority of devices that could receive a message only received one.

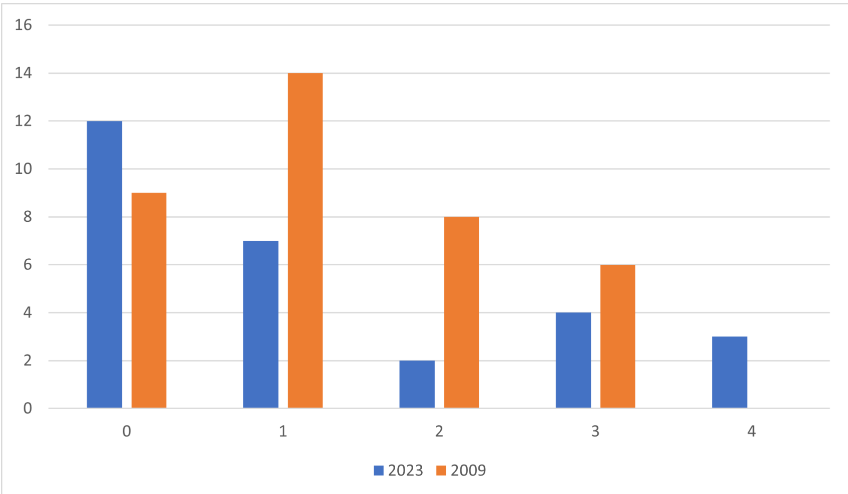


Figure 7.5: Total messages received by all devices after all test rounds. Data from [Rem09] for the year 2009.

7.3.2 Responsiveness test from log

The log of the system provides a more detailed look at what happened during the four rounds of the experiment. The data from this experiment log and the total results of the old study's log are presented in Table 7.4. Interestingly, the number of successful attempts is less than the number of people in the questionnaire who received the message. However, if we look at the number of delivery attempts, it shows almost the same number as people who received the messages according to the questionnaire. This may be because the students marked it as receiving a message as soon as they got a message on their screen to accept the message. Only when the message is accepted or declined it is logged as successful. The message is technically already on the phone at that point. This behavior is also described in the GOEP and OPP (see Section 2.3.2). It can also be seen that the software found devices like iPhones but did not find any services, sometimes even multiple times. Therefore, the software was more occupied with those devices instead of working on other devices that could have received a message. This explains why not all devices were reached in every round. This could potentially be optimized by skipping service search if it knows it is an iPhone to reach more devices.
Comparing the log results to the results of 2009, they had more delivery attempts per phone and more connection terminations. However, they also found more devices in a shorter period of time. The total count enclosed in brackets represents the unique number of devices identified in each round, summed up without any duplicates. The bracketed total indicates that the software has found every device.

Round	Unique devices	Delivery attempts		Successful attempts	Connection terminated		No service found	
		Total	Unique		Total	Unique	Total	Unique
1	16	9	9	5	0	0	11	9
2	20	11	11	7	0	0	7	6
3	18	11	11	9	0	0	11	9
4	19	6	6	5	0	0	15	12
total	73 (27)	37	37	26	0	0	44	36
total 2009	144(40)	176	100	48	10	7	118	66

Table 7.4: Results from log data. Data from [Rem09] for the year 2009.

7.4 Endurance test

The long-term test was done three times. Table 7.5 shows the absolute numbers of the experiment for every round and the total of all three rounds.
Even though two rounds were conducted on a busy Saturday during peak hours, it can be observed that the last round had the lowest number of devices found and successful message transmission attempts, while the second round was the most successful. The first test found more devices and was more successful in sending messages compared to the third test, which is surprising, since there was less foot traffic to be seen during the test.
Comparing the results to 2009 Table C.4, shows that they have tested longer periods of time. They had a total of 1934 devices over a period of 44.5 hours, resulting in an average of 43.5 devices per hour. In 2023, 237 devices were discovered within 10 hours, averaging at 23.7 devices per hour. This is approximately half of the results obtained in 2009.

In 2023, the improved performance is evident in the success rate of message delivery attempts, with 53 successful attempts (equivalent to 5.3 per hour), compared to only 47 attempts (equivalent to 1.03 per hour) in the previous results. In the current findings, if a service was detected, the success rate was 66.25%, whereas in 2009, only 4.53% of the attempts were successful. In 2009, there were no limitations on the number of service search attempts per device, unlike the current project, which had a limit of three attempts due to the resources of the dongle being occupied by discoverable but non-compatible devices such as TVs. The devices from the 2009 study were searched for the service approximately eight times.

Round	Time	Unique devices	Delivery attempts		Successful attempts	Connection terminated		No service found	
			Total	Unique		Total	Unique	Total	Unique
1	3h	95	20	17	12	8	6	176	84
2	4h	96	47	45	33	14	13	157	73
3	3h	46	13	13	8	4	4	92	39
total	10h	237 (215)	80	75	53	26	23	425	196
total 2009	44.5h	1934	1037	564	47	990	526	15167	1890

Table 7.5: Results from the endurance test

The names of the devices were saved in the log. Upon examination of the log, it contained various names that indicated what kind of devices were found. There were phones, smartwatches, TVs, headphones, a cloud printer and other miscellaneous names that could not be categorized. The logs revealed 33 OPPO, 30 OnePlus, and 22 iPhone devices. Looking at the market share of brands, OnePlus (0.85% market share Germany) and Oppo (1.09% market share Germany) have a high presence statistically compared the market share [SC23d].
This can be explained by the following: Nowadays most Android devices can only be found if the user was in the Bluetooth settings. Otherwise, the device is not discoverable/visible [AnD23b]. People do not often go into their Bluetooth settings. Older devices are able to turn on visibility the entire time [TR21]. There is a high chance that OPPO and Oneplus phones were running on an older version of Android, since 52.98% of these phones [SC23a] run older versions than 12. However, it is intriguing that no older versions of Android from other phone brands were found. OPPO and OnePlus devices probably were set to be discoverable the constantly as a standard setting. Alternatily, they could have an adapted interface on top of Android with other settings added.

7.5 Simultaneous connections test

The last test was the maximum simultaneous connections reached in a round with the number of dongles present. The results are depicted in Table 7.6. Since every dongle can theoretically service seven devices over Bluetooth Classic at once, the max number is multiplied by the number of dongles. Depending on the definition of initializing, there are two distinct versions of the maximum simultaneous connections achieved. The first number is only if it prepares and sends a message (only ObexPutClient.java). In brackets is the number which includes service searching (ObexPutClient.java and ServiceSearch.java).
In the small group test with the four test rounds, only in the first round a maximum was reached with one dongle. If more delivery dongles were used, the number even decreased with the number of delivery dongles. Counting the numbers in brackets, a better result with almost constant 10

devices is achieved. In 2009, they were also only able to reach a maximum of 14 connections, despite having three dongles that could theoretically support up to 21 connections. This could be attributed to limitations in the software's processing speed and other contributing factors.

Round	Nr. of delivery dongles	Max simultaneous connections possible	Max simultaneous connections achieved 2023	2009
1	1	7	7(10)	7
2	2	14	5(10)	13
3	3	21	4(7)	14
4	2	14	3(10)	14

Table 7.6: Max simultaneous connections possible and achieved (short test)

Also, looking at the numbers of the long-term testing in Table 7.7, only two simultaneous sending devices were reached. When taking into account the service search, this number increased to as high as 18 simultaneous connections. However, in 2009, they were able to achieve higher numbers in each round. This suggests that the capability to reach multiple devices simultaneously may be reduced in 2023 compared to 2009.

Round	Nr. of delivery dongles	Max simultaneous connections possible	Max simultaneous connections achieved 2023	2009
1	3	21	2(18)	21
2	4	28	2(13)	19
3	4	28	2(8)	22

Table 7.7: Max simultaneous connections possible and achieved (long test)

8 Conclusion and outlook

8.1 Conclusion

In conclusion, the research findings demonstrate that the success of Bluetooth Classic largely depends on the specific use case. When attempting to send data, the technology performed significantly better due to its faster data rates, greater distances, and overall reliability. We also found that the readability of data was better on newer phones.
However, the technology proved less effective in enabling simultaneous connections and reach less devices. The limitation of Android devices only being visible during Bluetooth settings made it difficult to discover devices. Additionally, devices that are not compatible were still found. The inclusion of devices such as iPhones, TVs, smartwatches, and headphones has made the Bluetooth network more complex and congested, making it difficult in identifying compatible devices. Especially the Bluetooth TVs were constantly discoverable and used a lot of bandwidth of the dongles. In the time the dongle had to check if it is a new device and if it is compatible, another compatible device could have already walked out of the radius of reaching it. All this results in fewer compatible devices being discovered and, consequently, less simultaneous connections.
In my opinion, the Bluetooth technology is better documented and developed on the Android platform, for example. There have been limited developments of new Bluetooth libraries compatible with Windows, Linux, or Mac platforms.

8.1.1 Bluetooth LE

Bluetooth LE emerged as a promising technology for this project, given its achievable range and official compatibility to broadcasting data. However, the long-term testing was not performed due to the limitation of not having an app. The complexity of developing an app for multiple platforms, including Android, iOS, and even operating systems of PCs such as Windows, Mac and Linux, makes it difficult. We were also limited in terms of the number of people available to test the app. It is worth noting that Bluetooth LE allows the use of a phone as the host, but it cannot use a Bluetooth dongle. It is limited to the built-in Bluetooth chip. It could be beneficial to further test the differences between a phone and a PC using a long-range dongle. It is important to note that the Nearby Standard, which was better suited for big data transfers over a website and free Wi-Fi, was discontinued in 2018.

8.2 Future work

For future work, it is important to explore ways to motivate people to leave their Bluetooth enabled inside an app, particularly for the purpose of device discovery.
One approach could be to develop an app that can run in the background, similar to the Corona WarnApp. The Corona WarnApp is an application used in Germany to track the virus causing COVID-19. The app works by collecting nearby identifiers. On mobile devices, the Exposure Notification System created by Apple and Google broadcasts a unique identifier that changes after a short period of time. These are called the Rolling Proximity Identifiers. The system also scans for other phones' identifiers using Bluetooth LE technology. [CWA23]
Instead of tracing COVID-19, this new app could be used for other beneficial purposes for the user. For instance, it could be integrated into a reward program or a game that people could use for entertainment.
For example, Universal Studios sells Power-Up Bands™ at SUPER NINTENDO WORLD that allow users to engage in more interactive play [SNW23]. The Bluetooth function on a user's phone could be used instead of a band to keep score, and complete challenges. Or similar to the project's initial concept in 2009 [Rem09], it could be utilized to send information based on the location, such as updates about exams, lectures, and news within the university.
However, it is important to consider the security implications of using such an app and ensuring that it uses encryption and authorization.
Bluetooth LE with an app could be a more reliable option than Bluetooth Classic. It would allow for connections with Apple users and other brands that may have had issues with compatibility and technically has the possibility to broadcast to an unlimited number of devices.
Additionally, it may be worth exploring the use of Auracast. We could encode data as audio as a potential means of broadcasting data. SIG is also looking into expanding the Auracast feature and allowing for sending video signal with it [SIG23c].

Bibliography

[AG21] Andy G. *Different Bluetooth Versions: What You Need to Know*. 2021. URL: https://www.headphonesty.com/2021/01/bluetooth-versions/ (cit. on p. 16).

[AH20] A. Hsu. *Is it possible to send a notification via BLE to mobile devices without installing an app?* 2020. URL: https://medium.com/coder-life/is-it-possible-to-send-a-notification-via-ble-to-the-users-mobile-device-without-install-app-684ff997cce0 (cit. on p. 25).

[AJN09] M. Aiello, R. de Jong, J. de Nes. "Bluetooth broadcasting: How far can we go? An experimental study". In: *2009 Joint Conferences on Pervasive Computing (JCPC)*. 2009, pp. 471–476. DOI: 10.1109/JCPC.2009.5420140 (cit. on p. 13).

[AlH19] PyBluez Team. *Github PyBluez*. 2019. URL: https://github.com/pybluez/pybluez (cit. on p. 35).

[AnD18] Android Developers. *Discontinuing support for Android Nearby Notifications*. 2018. URL: https://android-developers.googleblog.com/2018/10/discontinuing-support-for-android.html (cit. on p. 26).

[AnD21] Android Developers. *Find Bluetooth devices*. 2021. URL: https://developer.android.com/guide/topics/connectivity/bluetooth/find-bluetooth-devices (cit. on p. 46).

[AnD23a] Android Developers. *Platform Architecture*. 2023. URL: https://developer.android.com/guide/platform (cit. on p. 33).

[AnD23b] Android Developers. *Set up Bluetooth*. 2023. URL: https://developer.android.com/guide/topics/connectivity/bluetooth/setup (cit. on pp. 33, 52).

[Apple19] Apple. *Bluetooth profiles that iOS and iPadOS support*. 2019. URL: https://support.apple.com/en-us/HT204387 (cit. on p. 47).

[Apple21] Apple. *AirDrop security*. 2021. URL: https://support.apple.com/guide/security/airdrop-security-sec2261183f4/web (cit. on p. 48).

[BC08] BlueCove Team. *BlueCove*. 2008. URL: http://www.bluecove.org/ (cit. on pp. 35, 36).

[CPST22] M. Cäsar, T. Pawelke, J. Steffan, G. Terhorst. "A survey on Bluetooth Low Energy security and privacy". In: *Computer Networks* 205 (2022), p. 108712. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2021.108712. URL: https://www.sciencedirect.com/science/article/pii/S1389128621005697 (cit. on pp. 18–20, 23, 24).

[CWA23] German government. *Corona-Warn-App Open Source Project*. 2023. URL: https://www.coronawarn.app/en/ (cit. on p. 56).

[DG22] A. Dixit, K. Guleri. "An AI and Equation based Modelling Solution for Multiple Bluetooth Connections with Single Master Device". In: *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*. 2022, pp. 19–23. DOI: 10.23919/INDIACom54597.2022.9763152 (cit. on pp. 27, 28).

[Intel22] Intel Corporation. *What Is Bluetooth Technology?* 2022. URL: https://www.intel.com/content/www/us/en/products/docs/wireless/what-is-bluetooth.html (cit. on p. 25).

[ISKK21] H. Ishizaki, R. Saka, E. Kohno, Y. Kakuda. "A Loop Reduction Method Utilizing Public Device Addresses of Master Terminals in Connection Procedures of Multiple Piconets for Bluetooth MANETs". In: *International Journal of Networking and Computing* 11.2 (2021), pp. 492–515. DOI: 10.15803/ijnc.11.2_492 (cit. on p. 28).

[Joe12] Joel de Nes. "Bluetooth Broadcasting: Scaling in a proximity student information system". bachelor thesis. University of Groningen, 2012. unpublished thesis (cit. on p. 13).

[KHC23] A. Kumar, J. Hussain, A. Chun. "Bluetooth". In: *Connecting the Internet of Things : IoT Connectivity Standards and Solutions*. Berkeley, CA: Apress, 2023, pp. 215–287. ISBN: 978-1-4842-8897-9. DOI: 10.1007/978-1-4842-8897-9_4. URL: https://doi.org/10.1007/978-1-4842-8897-9_4 (cit. on pp. 13, 17–20).

[Moh22] K. S. Mohamed. "An Introduction to Bluetooth". In: *Bluetooth 5.0 Modem Design for IoT Devices*. Cham: Springer International Publishing, 2022, pp. 1–32. ISBN: 978-3-030-88626-4. DOI: 10.1007/978-3-030-88626-4_1. URL: https://doi.org/10.1007/978-3-030-88626-4_1 (cit. on pp. 15, 18, 25).

[OSM23] OpenStreetMap contributors. *openStreetMap location of OpenField*. 2023. URL: https://www.openstreetmap.org/ (cit. on p. 39).

[PSF17] Python Software Foundation. *18.1. socket — Low-level networking interface*. 2017. URL: https://docs.python.org/3.3/library/socket.html (cit. on p. 35).

[Rem09] Remko de Jong. "Bluetooth Broadcasting: Designing a scalable system to distribute context aware information to mobile groups via Bluetooth". master thesis. University of Groningen, 2009. unpublished thesis (cit. on pp. 13, 31, 33, 39, 40, 42, 45, 46, 50, 51, 56, 63, 65–67).

[Sau22] M. Sauter. "Bluetooth". In: *Grundkurs Mobile Kommunikationssysteme: 5G New Radio und Kernnetz, LTE-Advanced Pro, GSM, Wireless LAN und Bluetooth*. Wiesbaden: Springer Fachmedien Wiesbaden, 2022, pp. 339–389. ISBN: 978-3-658-36963-7. DOI: 10.1007/978-3-658-36963-7_5. URL: https://doi.org/10.1007/978-3-658-36963-7_5 (cit. on pp. 15, 19–23, 25).

[SC23a] StatCounter. *Android Version Market Share Germany 26.03.2023*. 2023. URL: https://gs.statcounter.com/android-version-market-share/all/germany (cit. on p. 52).

[SC23b] StatCounter. *Mobile Operating System Market Share Germany*. 2023. URL: https://gs.statcounter.com/os-market-share/mobile/germany (cit. on p. 49).

[SC23c] StatCounter. *Mobile Operating System Market Share Worldwide*. 2023. URL: https://gs.statcounter.com/os-market-share/mobile/worldwide (cit. on p. 49).

[SC23d] StatCounter. *Mobile Vendor Market Share Germany*. 2023. URL: https://gs.statcounter.com/vendor-market-share/mobile/germany (cit. on p. 52).

[SIG23a] Bluetooth Special Interest Group. *2023 Bluetooth Market Update*. 2023. URL: https://www.bluetooth.com/2023-market-update/ (cit. on pp. 13, 14, 16, 17, 29).

[SIG23b] Bluetooth Special Interest Group. *About Us*. 2023. URL: https://www.bluetooth.com/about-us/ (cit. on p. 15).

[SIG23c] Bluetooth Special Interest Group. *Answers to Commonly Asked Questions About Auracast Broadcast Audio*. 2023. URL: https://www.bluetooth.com/blog/answers-to-commonly-asked-questions-about-auracast-broadcast-audio/ (cit. on pp. 28, 29, 56).

[SIG23d] Bluetooth Special Interest Group. *Bluetooth Core Specification Version 5.4 - Technical Overview*. 2023. URL: https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-version-5-4-technical-overview/ (cit. on pp. 16, 28).

[SNW23] Universal Studios. *SUPER NINTENDO WORLD™ - A New Way to Play!* 2023. URL: https://www.universalstudioshollywood.com/web/en/us/things-to-do/lands/super-nintendo-world (cit. on p. 56).

[TK19] N. Todtenberg, R. Kraemer. "A survey on Bluetooth multi-hop networks". In: *Ad Hoc Networks* 93 (2019), p. 101922. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2019.101922. URL: https://www.sciencedirect.com/science/article/pii/S157087051930126X (cit. on p. 27).

[TL23] Truelist. *Linux Statistics – 2023*. 2023. URL: https://truelist.co/blog/linux-statistics/ (cit. on pp. 33, 34).

[TR21] T. Rutter. *Bluetooth visibility (My device)*. 2021. URL: https://tamingthedroid.com/my-device-visible-not-visible (cit. on p. 52).

[TT18] tinyb Team. *Github Tiny Bluetooth LE Library*. 2018. URL: https://github.com/intel-iot-devkit/tinyb (cit. on p. 35).

[VM18] V. Kolotov, M. Rathgeb. *GitHub bluetooth-manager*. 2018. URL: https://github.com/sputnikdev/bluetooth-manager (cit. on p. 35).

[YPMN19] J. Yang, C. Poellabauer, P. Mitra, C. Neubecker. *Beyond Beaconing: Emerging Applications and Challenges of BLE*. 2019. DOI: https://doi.org/10.48550/arXiv.1909.11737. arXiv: 1909.11737 [cs.NI] (cit. on pp. 18, 25).

[ZSB19] S. Zeadally, F. Siddiqui, Z. Baig. "25 Years of Bluetooth Technology". In: *Future Internet* 11.9 (2019). ISSN: 1999-5903. DOI: 10.3390/fi11090194. URL: https://www.mdpi.com/1999-5903/11/9/194 (cit. on p. 25).

All links were last followed on April 12, 2023.

A Sequence diagram

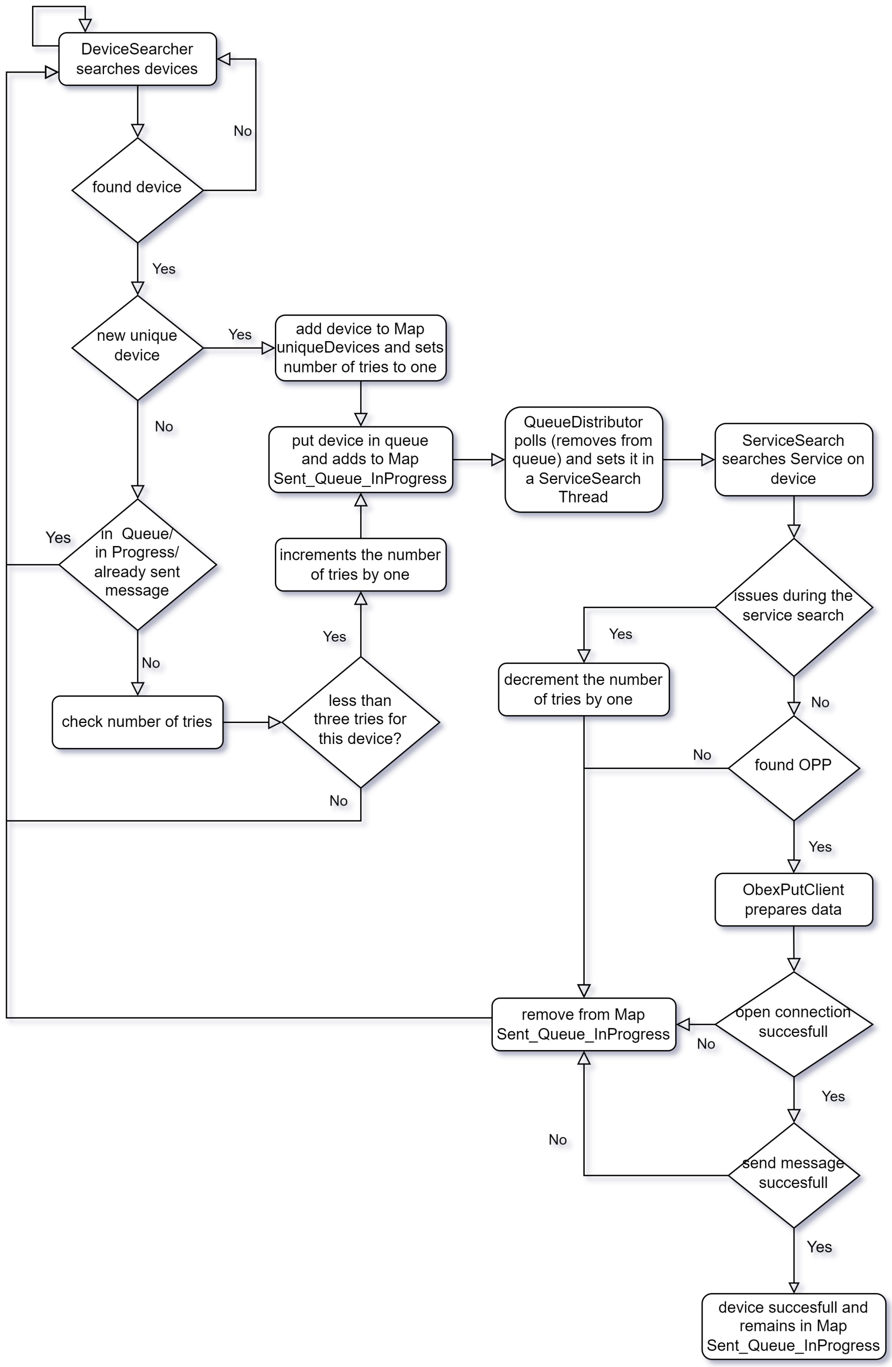
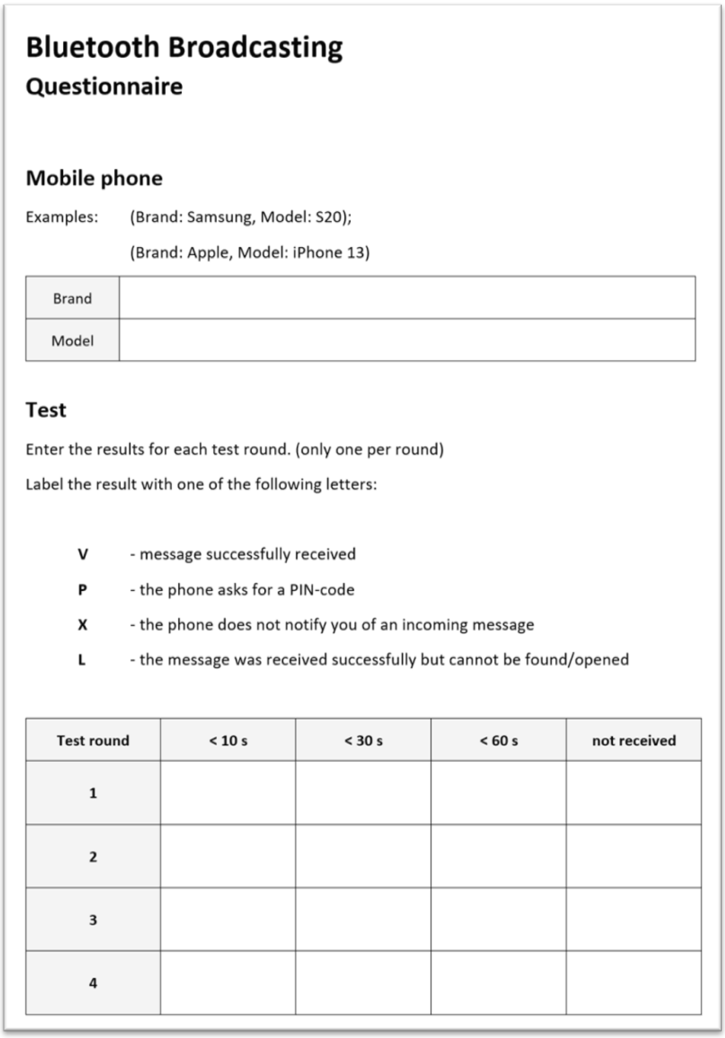


Figure A.1: Sequence diagramm of the software project

B Questionnaire

B.1 Design of the questionnaire

The questionnaire in Figure B.1 was designed similar to the questionnaire in 2009 [Rem09]. The language here is English, because the students, which were partaking in the experiment were in an English hold module at the University Stuttgart.



Bluetooth Broadcasting

Questionnaire

Mobile phone

Examples: (Brand: Samsung, Model: S20);

(Brand: Apple, Model: iPhone 13)

Brand	
Model	

Test

Enter the results for each test round. (only one per round)

Label the result with one of the following letters:

V - message successfully received

P - the phone asks for a PIN-code

X - the phone does not notify you of an incoming message

L - the message was received successfully but cannot be found/opened

Test round	< 10 s	< 30 s	< 60 s	not received
1				
2				
3				
4				

Figure B.1: Design of the questionnaire

B.2 Results of the questionnaire

The results from the questionnaire condensed in Figure B.2. For the description of the indicators look at the questionnaire in Appendix B.1. (n.r. = not received)

		round1				round2				round3				round4			
Brand	Model	<10s	<30s	<60s	n.r.	<10s	<30s	<60s	n.r.	<10s	<30s	<60s	n.r.	<10s	<30s	<60s	n.r.
Apple	Iphone 11				X				X				X				X
Apple	Iphone 12				X				X				X				X
Apple	Iphone 12				X				X				X				X
Apple	Iphone 12 Pro				X				X				X				X
Apple	Iphone 12 Pro Max				X				X				X				X
Apple	Iphone 13 mini				X				X				X				X
Apple	Iphone 13 mini				X				X				X				X
Apple	Iphone 13 Pro				X				X				X				X
Apple	Iphone 13 Pro Max				X				X				X				X
Apple	Iphone 14 Pro Max				X				X				X				X
Fairphone	4				X	V							X				X
Google	Pixel 6		V				V					V			V		
Google	Pixel 6 pro				X				X			V				V	
Google	Pixel 6a	V						V		V						V	
Honor	8x			V				V			V					V	
Motorola	G60			V					X			V				V	
OnePlus	6			V				V					X				X
OnePlus	Nord CE				X				X				X				X
Redmi	Note 5				X				X				X				X
Samsung	A52s				X			L					X				X
Samsung	S10			V					X	V						V	
Samsung	S10+			V			V					V					X
Samsung	S20e				X			P					X				X
Samsung	S20FE				X			V					X				X
Xiaomi	11T		V			V						V					X
Xiaomi	Mi10				X				X		V						X
Xiaomi	Mi10				X				X			V					X
Xiaomi	X3 Poco Pro				X			V				V					X

Figure B.2: Results of the questionnaire

C Results from 2009

C.1 Range test

Brand	Discovery	Message received
Linksys	111m	29m
Sweex	36m	36m
Sitecom	46m	32m
Conceptroni	45m	34m

Table C.1: Different brands of dongles (max distance* adapted). Adapted from [Rem09]

C.2 Reception test

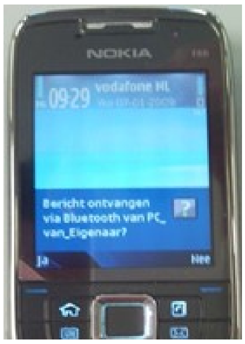
(a) Nokia – View of message

(b) Nokia E71 – Message received in text message inbox

(c) Nokia E71 – Message received in text message inbox

(d) Sharp 920SH – Pairing request

(e) Sony Ericsson w710i – Accept incoming message



(f) Nokia E66 – Accept incoming message

Figure C.1: Reception of message by different brands of mobile phones. Adapted from[Rem09]

C.3 Responsiveness test

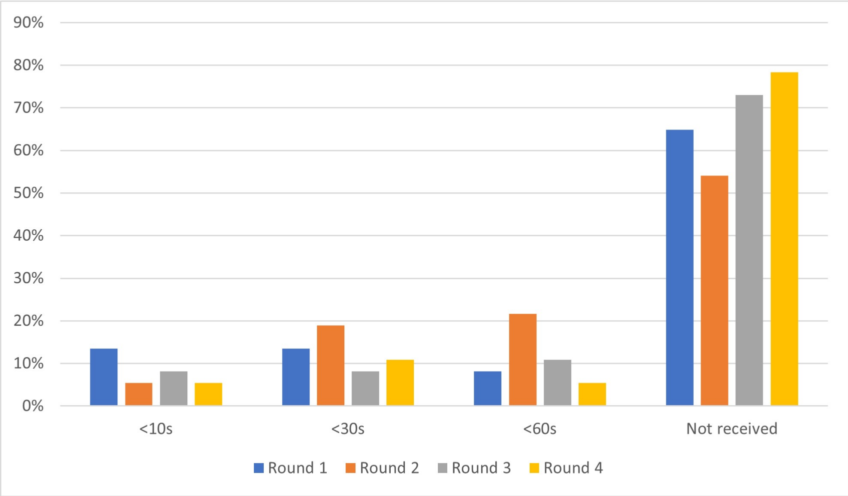


Figure C.2: Percentage of messages received per round. Adapted from [Rem09]

Round	< 10s	< 30s	< 60s	Received	Not received
1	5	5	3 (1)	13 (1)	24
2	3	3	4 (2)	10 (2)	27
3	2 (1)	7 (1)	8 (2)	17 (4)	20
4	2 (2)	4 (1)	2 (1)	8 (4)	29
total	12 (3)	19 (2)	17 (6)	48 (13)	100

Table C.2: Data from the questionnaire. Adapted from [Rem09]

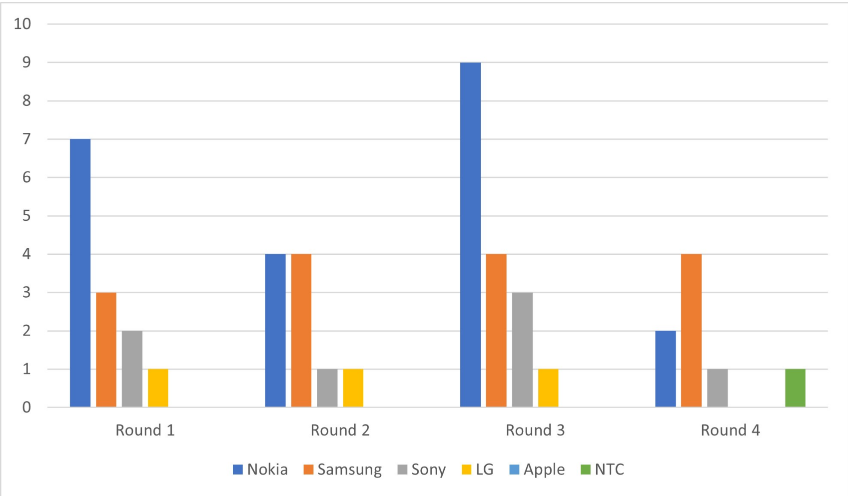


Figure C.3: Number of messages received per brand. Adapted from [Rem09]

Round	Unique devices	Delivery attempts		Successful attempts	Connection terminated		No service found	
		Total	Unique		Total	Unique	Total	Unique
1	36	28	20	14	2	1	12	10
2	36	42	24	9	2	1	31	20
3	34	43	30	17	2	2	24	16
4	38	63	26	8	4	3	51	20
total	144(40)	176	100	48	10	7	118	66

Table C.3: Data from the log. Adapted from [Rem09]

C.4 Endurance test

Round	Time	Unique devices	Delivery attempts		Successful attempts	Connection terminated		No service found	
			Total	Unique		Total	Unique	Total	Unique
1	8h	439	241	136	8	233	129	1567	412
2	19h	612	228	140	13	215	127	5314	606
3	17.5h	883	568	288	26	542	270	8286	872
total	44.5h	1934	1037	564	47	990	526	15167	1890

Table C.4: Endurance test results. Adapted from [Rem09]

D Kurzfassung - Abstract in German

Die Welt der Technologie verändert sich rapide und entwickelt sich weiter, auch Bluetooth. Im Jahr 2009 wurde an der Universität Groningen eine experimentelle Studie durchgeführt, um die Bluetooth-Technologie hinsichtlich der Übertragung von Daten an eine Vielzahl von Geräten zu bewerten. Die Autoren untersuchten insbesondere die Leistungsfähigkeit und Skalierbarkeit. Sie kamen zu dem Schluss, dass Bluetooth für die Datenübertragung an viele Geräte geeignet ist, obwohl es auch Einschränkungen in der Technologie gibt, wie zum Beispiel die Begrenzung von sieben gleichzeitig verbundenen Geräten an eine einzige Antenne (Bluetooth-Dongle).
Im Jahr 2023 wurde eine Überprüfung durchgeführt, um zu untersuchen, wie diese Aussage sich bewährt hat und wie Bluetooth mit seinen neuen Entwicklungsversionen zur heutigen Übertragung beitragen kann. Daher werden in dieser Arbeit die Veränderungen in Bluetooth und dessen Funktionsweise erforscht, die Hardwarekompatibilität und Veränderungen im Marktanteil von heute. Darüber hinaus wurden experimentelle Studien nach dem Vorbild von der Studie von 2009 durchgeführt, um die Leistungsfähigkeit und Skalierbarkeit mit realen Daten zu vergleichen.
Im Jahr 2009 verwendeten die Autoren Bluetooth-Version 2.1 für ihr Experiment, während im Jahr 2023 Version 5.4 verwendet wurde. Im Laufe der Jahre wurden mit jeder Version von Bluetooth viele Verbesserungen hinzugefügt. Insbesondere mit den besseren Datenraten und der besseren Zuverlässigkeit von Bluetooth hat die Technologie großes Potenzial. Eine vielversprechende Änderung ist die Hinzufügung von Bluetooth Low Energy (LE), was für die Datenübertragung vorteilhaft sein kann. Obwohl Bluetooth LE die Fähigkeit hat, eine unbegrenzte Anzahl von Geräten mit größerer Entfernung mit nur einem Dongle zu erreichen, ist es wichtig zu beachten, dass es auch Einschränkungen gibt. Insbesondere ist es nur mit Apps kompatibel und funktioniert nicht nativ auf einem Telefon.
In unseren experimentellen Studien wurde festgestellt, dass die Bluetooth-Technologie bessere Leistung in Bezug auf Datenübertragungsraten und Reichweite zeigt, wenn sie mit kompatiblen Geräten verbunden ist. Trotz der steigenden Anzahl von Bluetooth-fähigen Geräten im Jahr 2023 im Vergleich zu 2009 war Bluetooth in der Lage, weniger Geräte zu lokalisieren. Dies liegt hauptsächlich daran, dass Geräte in ihren Bluetooth-Einstellungen aktiviert sein müssen, um erkannt zu werden, und es gibt viele andere Nicht-Telefon-Geräte wie Fernseher, Smartwatches, Kopfhörer und mehr, die ebenfalls die Bluetooth-Technologie nutzen. Darüber hinaus ist Bluetooth-Technologie hauptsächlich mit Android-Geräten kompatibel, die das offene Standardprofil Object Push Profile (OPP) verwenden. Dies stellt ein Problem für viele iPhones auf dem Markt dar, da sie nicht kompatibel sind und nur mit einer App oder ihrem eigenen Airdrop-Standard arbeiten können.
Zussamenfassend kann man sagen, dass Bluetooth-Datenübertragung an vielen Geräten möglich ist, jedoch ist Bluetooth Classic aufgrund seiner aktuellen Einschränkungen nicht optimal. Die Empfehlung für die Übertragung von Daten für eine Viezahl an Geräte ist, dass Benutzer eine App haben müssen, in der sie Daten empfangen möchten, und Bluetooth LE verwenden sollten, da es eine bessere Reichweite hat und keine Einschränkungen hinsichtlich der Anzahl von Geräten hat.
Schlüsselwörter: Bluetooth, Broadcasting, Experimentelle Studie, Bluetooth Low Energy, Marktanteil, Bluetooth-Geschichte, Kompatibilität, App, Skalierbarkeit.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature