Praktikumsversuch
Bluetooth

Dr.-Ing. Maik Debes
B.Sc. Max Helbig

13. August 2019

Inhaltsverzeichnis

1 Versuchsziel

In diesem Versuch soll grundlegendes Wissen zum Thema Bluetooth vermittelt und gefestigt werden. Mit Hilfe von Bluetooth-Geräten werden Versuche durchgeführt, die die Eigenschaften und Besonderheiten von Bluetooth verdeutlichen. Dazu zählen vor allem die Koexistenz von Bluetooth- und **W**ireless **L**ocal **A**rea **N**etwork (WLAN)-Geräten sowie das Bluetooth-Protokoll, das mit dem BPA500 Bluetooth-Analysator ausgewertet wird. Um die während des Versuchs gestellten Aufgaben durchführen, sowie die Ergebnisse besser einordnen zu können, sollte der theoretische Teil der Anleitung durchgearbeitet werden (Die Vorbereitungsaufgaben sind verpflichtend).

2 Allgemeines

Bluetooth ist eine Technik zur drahtlosen Verbindung von Geräten. Durch fertige Ein-Chip-Lösungen sind sehr preisgünstige Implementationen möglich. Bluetooth ist kein proprietäres Verfahren, sondern ein Industriestandard, der Verbindungen zwischen Geräten wie Druckern, PCs, Smartphones, Headsets usw. ermöglicht. Auch das Einbinden von Geräten in Netze oder der Aufbau von Ad-Hoc-Netzen wird von Bluetooth unterstützt. Um die Entwicklung von Bluetooth kümmert sich die 1998 gegründete Bluetooth **S**pecial **I**nterest **G**roup (SIG).

Namensgeber: Harald Blåtand (dänisch für Blauzahn) (geb. um 910, gest. 1.11.986) vereinte 983 Dänemark und Norwegen.

3 Der Bluetooth Protocol Stack

Der Bluetooth-Protokollstapel stellt das Regelwerk dar, nach dem sich alle Bluetooth-Geräte richten müssen. Er enthält alle standardisierten Protokolle und Schnittstellen (siehe Abbildung 1).

Die Identifikation eines Bluetooth-Gerätes erfolgt über seine Adresse. Diese ist, ähnlich wie die **M**edia **A**ccess **C**ontrol (MAC)-Adresse bei Ethernet, sechs Byte lang und hat folgende hexadezimale Darstellungsform: 01:23:45:67:89:AB.

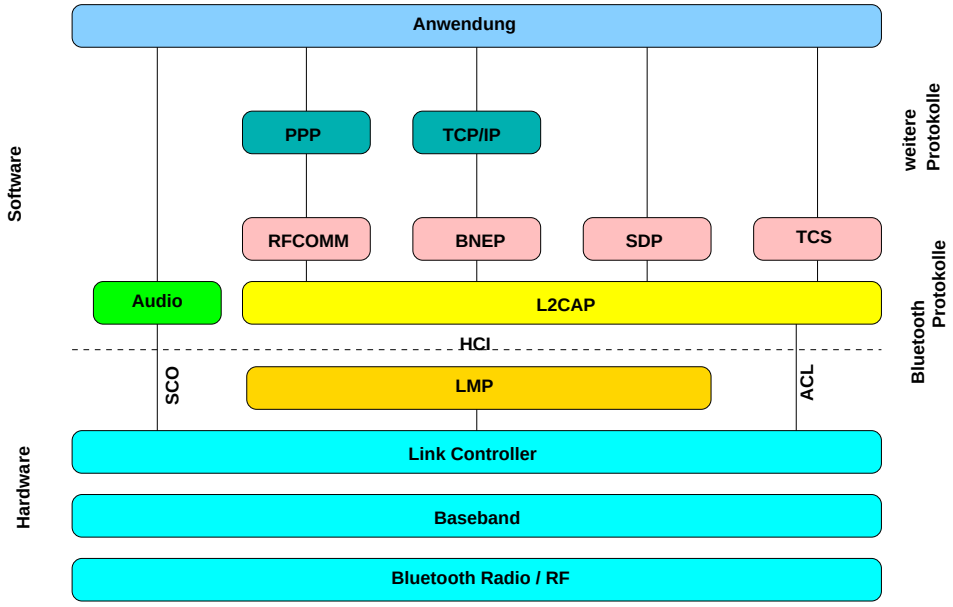

Abbildung 1: Hierarchie des Bluetooth-Protokollstapels [4]

Die folgende Grafik zeigt, wie der Bluetooth Protokoll Stack auf das ISO/OSI-Referenzmodell gemappt werden kann.

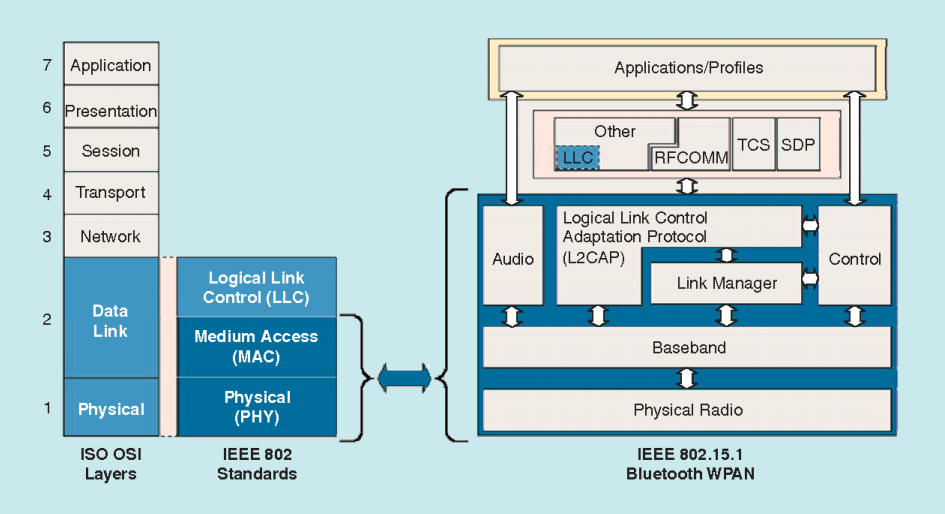

Abbildung 2: Mapping auf ISO/OSI-Referenzmodell [6]

3.1 Bluetooth Hardware

Dieser Absatz beschreibt die Funktionen der einzelnen Teile innerhalb der Hardware von Bluetooth-Geräten.

3.1.1 Frequenzen und Kanäle

Bluetooth funkt bei 2,4 GHz im lizenzfreien **I**ndustrial **S**cientific **M**edical (ISM)-Band. Die in Deutschland zur Verfügung stehende Bandbreite von 83.5 MHz (in anderen Ländern teilweise eingeschränkt) wird auf 79 **R**adio **F**requency (RF) Kanäle im Abstand von 1 MHz aufgeteilt. Die Datenübertragung erfolgt mit **T**ime **D**ivison **D**uplex (TDD). Es sind bei Bluetooth 1.1 und 1.2 Übertragungsraten bis zu 1 Mb/s, bei Bluetooth 2.0 sogar 3 Mb/s, bei einer Reichweite von 10 m bis 100 m erreichbar. Die Version 3.0+HS kann im Highspeed-Modus sogar bis zu 24 Mb/s erreichen (im Bluetooth-Modus hingegen weiterhin 3 Mb/s).

Bluetooth (bis Version 1.1) nutzt **F**requency **H**opping **S**pread **S**pectrum (FHSS). Das heißt, nach jedem Paket wird die Frequenz, durch Pseudozufallszahlen gesteuert, gewechselt (1600 mal pro Sekunde). Dies ermöglicht eine geringe Störempfindlichkeit durch schmalbandige Störer (z.B. WLAN) und erschwert das Abhören der Kommunikation.

Seit Bluetooth Version 1.2 wird statt FHSS das so genannte **A**daptive **F**requency **H**opping (AFH) verwendet. Zusätzlich zur Funktionalität von FHSS ermöglicht AFH den Ausschluss von bestimmten Kanälen aus der Sprungsequenz, falls diese von anderen Geräten (z.B. Funkmikrofonen) [5] belegt bzw. gestört werden.

3.1.2 Basisband

Das Basisband ist die physikalische Schicht des Stacks. Allerdings kann dies nicht exakt auf das ISO/OSI-Referenzmodell abgebildet werden. Das Basisband, der Link Controller und **L**ogical **L**ink **C**ontrol and **A**daption **P**rotocol (L2CAP) bilden gemeinsam die ersten beiden Schichten des ISO/OSI-Referenzmodells ab.

3.1.3 Der Link Controller

Der Link Controller steuert die Funkkanäle, die Frequenzwechsel und die Funkverbindungen. Zu seinen Aufgaben gehören auch die Fehlerkorrektur **F**orward **E**rror **C**orrection (FEC) und das Data-Whitening (Mischung der Bits eines Pakets, um kurze 0 bzw. 1 Sequenzen zu erzeugen).

3.1.4 Das Link Manager Protocol

Der Linkmanager nutzt dieses Protokoll für Verbindungssetup und -aufbau, Sicherheit, Identifikation und die Verbindungskontrolle. Es regelt unter anderem die Aushandlung von Paketgröße, RF-Sendeleistung und die Verwaltung des Piconets.

3.2 Das Host Controller Interface

Das **H**ost **C**ontroller **I**nterface (HCI) ist das Command-Interface für den Link-Manager und den Basisband-Controller. Es stellt einheitliche Zugriffsmethoden auf Basisbandfunktionen zur Verfügung, indem es für die elektrische/physikalische Ansteuerung der

Hardwareschnittstelle sorgt, mit der das Endgerät das Bluetooth-Gerät ansteuert. Bei einer Bluetooth-PC-Karte ist dies zum Beispiel der Cardbus-Steckplatz bzw. Cardbus-Controller im Notebook.

3.3 Software

3.4 Das Logical Link Control and Adaption Protocol

Neben der Unterscheidung von Protokollen höherer Ebenen (z.B. RFCOMM, BNEP, SDP) die auf dem L2CAP aufsetzen, übernimmt es auch die Übermittlung von Quality-of-Service-Informationen zwischen kommunizierenden Geräten.

Da die maximale Nutzdatengröße pro Paket variiert, müssen größere Datenmengen in entsprechende Teile zerlegt bzw. wieder zusammengefügt werden (Segmentierung, Reassemblierung). Dies ist ebenfalls eine Aufgabe des L2CAP. Die maximale Nutzdatengröße hängt von den Fähigkeiten der beteiligten Bluetooth-Geräte sowie der Übertragungsqualität ab. In Tabelle 1 sind die von der Bluetooth-SIG spezifizierten Pakettypen dargestellt.

Typ	Max. Nutzdaten in Byte
DM1	17
DH1	27
DM3	121
DH3	183
DM5	224
DH5	339
AUX1	29
2-DH1	54
2-DH3	367
2-DH5	679
3-DH1	83
3-DH2	552
3-DH3	1021

Tabelle 1: Spezifizierte Pakete (Tabelle 6.9 in [1])

3.5 Service Discovery Protocol (SDP)

Das SDP ermöglicht es Anwendungen, zur Verfügung stehende Dienste und deren Charakteristika herauszufinden.

Das SDP ist notwendig, da Bluetooth in dynamischer Umgebung arbeitet. Das bedeutet, die angebotenen Dienste können sich unterscheiden und geändert werden. Des Weiteren können die Dienstanbieter außer Reichweite geraten.

3.6 Radio Frequency Communication Port emulation (RFCOMM)

RFCOMM [3] emuliert serielle Schnittstellen über Bluetooth und stellt so ein einfaches Transportprotokoll zur Verfügung. Es werden bis zu 60 simultane Verbindungen zwischen 2 Bluetooth-Geräten unterstützt. Viele Protokolle höher liegender Schichten bauen auf dem RFCOMM-Protokoll auf.

3.7 Bluetooth Network Encapsulation Protocol (BNEP)

BNEP steuert den einheitlichen Transport von Netzwerkpaketen über eine Bluetooth-Verbindung. Dabei hat es bezüglich des **I**nternet **P**rotocol (IP) die gleiche Funktion, die in Ethernet-Netzwerken von der MAC-Schicht übernommen wird. Somit entspricht BNEP der Schicht 2 im ISO/OSI-Referenzmodell.

Auf BNEP wird IP aufgesetzt, welches weitere typische Protokolle wie **T**ransmission **C**ontrol **P**rotocol (TCP), **U**ser **D**atagram **P**rotocol (UDP), **I**nternet **C**ontrol **M**essage **P**rotocol (ICMP) uvm. über eine Bluetoothverbindung ermöglicht.

3.8 Weitere Protokolle

Im Bluetooth-Standard sind noch andere Protokolle, wie z.B. **Ob**ject **Ex**change Protocol (OBEX) und **T**elephony **C**ontrol **S**pecification (TCS) definiert. Für zusätzliche Informationen siehe [1][2][4][8].

3.9 Netztopologie

Kommunizierende Bluetooth-Geräte bilden ein so genanntes Piconetz. Dabei fungiert genau ein Gerät als Master, alle anderen Teilnehmer werden als Slave bezeichnet. Die Slaves können nicht direkt miteinander in Verbindung treten. Alle Daten müssen über den Master, der die gesamte Datenübertragung eines Piconetzes koordiniert, gesendet werden.

Es können sich bis zu 255 Slaves in einem Piconetz anmelden, wobei nur sieben Slaves gleichzeitig aktiv Daten senden können. Da der Master immer aktiv sein muss, können maximal acht Geräte gleichzeitig kommunizieren.

Ein Bluetooth-Gerät kann Mitglied in mehreren Piconetzen sein, jedoch Master nur in einem. Piconetze können durch die Verknüpfung über ein Bluetooth-Gerät zu so genannten Scatternetzen zusammengeschlossen werden (siehe Abbildung 3).

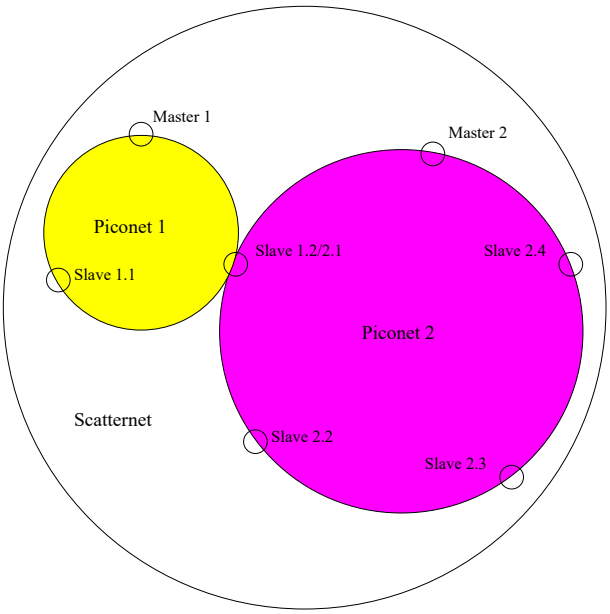

Abbildung 3: Piconet und Scatternet

Mit der Einführung von Bluetooth-Low-Energy in der Bluetooth-Version 4.0 wird es möglich, beliebig viele Slaves mit einem Master zu verbinden (siehe Abbildung 4). Dabei ist zu beachten, dass jeder Slave mit einer eigenen Hoppingsequenz mit dem Master kommuniziert. Zudem werden die Verbindungen nicht lange aufrechterhalten, um in den Slaves Energie zu sparen.

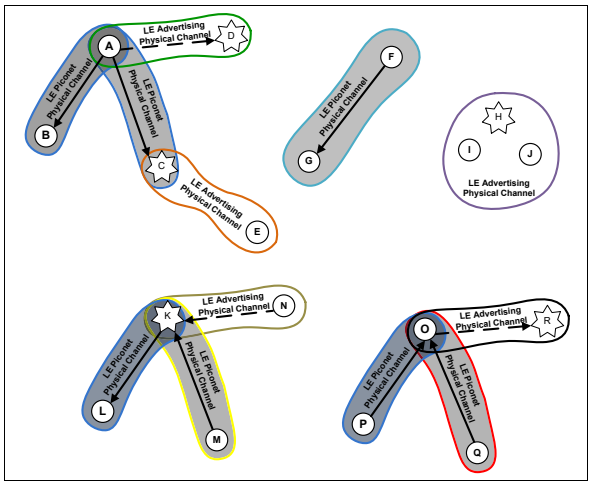

Abbildung 4: Bluetooth-LE-Topologie (Abbildung 4.2 in [1]) Durchgezogene Pfeile zeigen von einem Master zu einem Slave, gestrichelte Pfeile zeigen einen Verbingunsaufbau vom Initiator aus. Sternförmige Geräte sind Paarungsbereit.

Mit diesem System lässt sich eine Star-Bus-Topologie realisieren um beispielsweise viele Sensoren anzubinden. Diese Topologie kann nur bei Bluetooth-Low-Energy genutzt werden. Klassische Bluetooth-Geräte nutzen weiterhin die weiter oben beschriebenen Piconetze.

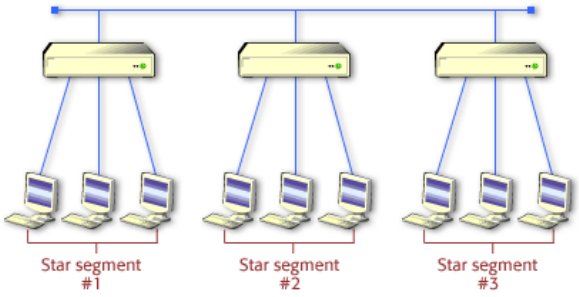

Abbildung 5: Star-Bus-topologie

3.10 Verbindungsarten

Bluetooth unterstützt sowohl verbindungsorientierte als auch verbindungslose Datendienste. Im Basisband stehen dazu zwei Übertragungsmodi zur Verfügung.

3.10.1 Synchronous Connection-Oriented (SCO)

Bei diesem Übertragungsmodus erfolgt eine feste Reservierung von Zeitschlitzen, und somit eines Kanals wodurch eine synchrone Verbindung zwischen Master und Slave realisiert wird. Es sind maximal 3 synchrone Kanäle mit je 64 kb/s zulässig, wobei die Datenrate pro Kanal fest reserviert und sowohl in Sende- als auch in Empfangsrichtung gleich groß ist.

SCO ist sehr gut für die zeitkritische Übertragung isochroner Datenströme geeignet und kommt beispielsweise bei der Sprachübertragung zum Einsatz.

3.10.2 Asynchron Connection-Less (ACL)

Im Gegensatz zu SCO bietet dieser asynchrone verbindungslose Übertragungsmodus eine paketorientierte Punkt-zu-Mehrpunktverbindung. Für den ACL-Kanal verbleibt die restliche Datenrate nach Abzug der maximal drei reservierbaren SCO-Kanäle. Die ACL-Daten werden in den nicht für SCO reservierten Zeitschlitzen übertragen. Sind keine Zeitschlitze für SCO reserviert, so steht dem ACL-Kanal die gesamte Datenrate zur Verfügung. Über den ACL-Kanal werden unter anderem Nutzer- oder Steuerdaten übertragen.

4 Bluetooth-Profile (Auswahl)

Die Unterstützung eines Bluetooth-Profils befähigt ein Bluetooth-Gerät zur Bewältigung einer bestimmten Aufgabe bzw. eines bestimmten Anwendungsfalles. Zwei kommunizierende Geräte müssen immer beide das gleiche Profil unterstützen, um die gestellte Aufgabe zusammen bewältigen zu können. Neben der Unterstützung des gleichen Profils muss auch die Rollenverteilung beider Geräte zueinander passen. Das bedeutet, ein Gerät stellt eine Aufgabe und der Kommunikationspartner kann diese lösen. Da Bluetooth-Geräte ihre Unterstützung für verschiedene Profile anzeigen können, ist ein gezieltes Suchen nach passenden Partnern möglich.

Die Profile und ihre Rollenzuordnung sind im Bluetooth-Standard festgelegt. Es existieren bereits über 30 Profile, wobei die Liste kontinuierlich erweitert wird. Es wird bereits eine Vielzahl von Anwendungen unterstützt. Dazu zählen die Bildung von privaten Netzwerken, Internetzugang, Datenaustausch zwischen Smartphones und anderen mobilen Geräten uvm.

4.1 Generic Access Profile (GAP)

Das GAP muss auf jedem Bluetooth-Gerät implementiert sein. Es beschreibt den Verbindungsaufbau und -status, das Verhalten im Standby-Modus, regelt die Verarbeitung weiterer spezifischer Profile und die Sicherheitsprozeduren (z.B.: PIN-Austausch).

4.2 Personal Area Network (PAN)

Das Netzwerk-Profil PAN ermöglicht den Aufbau von kleinen Netzwerken. In einem Ad-Hoc-Netzwerk können bis zu acht Teilnehmer aktiv miteinander kommunizieren.

Die Nutzung gemeinsamer Ressourcen wie Druckern, Festplatten oder eines Internetzugangs ist ähnlich wie in einem **L**ocal **A**rea **N**etwork (LAN) möglich.

4.3 Serial Port Profile (SPP)

Das SPP unterstützt den Aufbau serieller Verbindungen zwischen zwei Bluetooth-Geräten. Die Datenübertragung erfolgt dabei über RFCOMM.

Das SPP eignet sich zum Beispiel für die Kommunikation zwischen Bluetooth-Modems oder Mobiltelefonen.

5 Pairing

Das Pairing (dt. Paarung) ist die Verbindungsaufbauprozedur zwischen Bluetooth-Geräten. Es dient zur Überprüfung der Zutrittsberechtigung zu einem Piconetz. Während der Paarung ist die Eingabe der so genannten Bluetooth-PIN nötig. Bei beiden Geräten muss die gleiche PIN verwendet werden, da diese unter anderem zur Erzeugung des 128 Bit langen Verbindungsschlüssels genutzt wird. Dieser Schlüssel wird in beiden Geräten für zukünftige Verbindungen gespeichert. Das heißt, eine Paarung ist nur beim ersten Verbindungsaufbau zweier Geräte notwendig. Bei jedem weiteren Verbindungsaufbau wird statt der PIN nur noch der Verbindungsschlüssel als Zugangsberechtigung gebraucht.

Neben dem Verbindungsschlüssel, der ausschließlich als Zugangsberechtigung dient, wird noch ein weiterer Schlüssel mit einer Länge zwischen 8 und 128 Bit erzeugt. Er wird für die Sicherung der zu übertragenden Daten genutzt und muss für jede Sitzung neu ausgehandelt werden.

Diese Methode des Pairing wurde jedoch mit der Version 2.1 überarbeitet: Die Bluetooth SIG wollte die Komplexität (aus Sicht des Nutzers) senken und gleichzeitig die Sicherheit erhöhen. Dies geschieht mit dem SPP, die Paarung ist nun einfacher, da keine PIN mehr eingegeben werden muss. Dennoch ist die Verbindung gleichzeitig abhörsicherer, denn bisher genügte es einem Abhörer, die (meist nur 4-stellige) PIN zu kennen, um den Verbindungsschlüssel zu berechnen. Beim **S**ecure **S**imple **P**airing (SSP) wird stattdessen der Elliptic-Curve-DiffieHellmann-Algorithmus zur Schlüsselerzeugung genutzt, welcher zur Berechnung sogenannter öffentlicher Schlüssel und privater Schlüssel verwendet wird. Die privaten Schlüssel werden niemals über die Luftschnittstelle übertragen und ohne diese ist es für einen Abhörer unmöglich den Verbindungsschlüssel zu berechnen. Die gepaarten Geräte können jeweils anhand der eigenen privaten Schlüssel und der übertragenen öffentlichen Schlüssel den gleichen Verbindungsschlüssel berechnen.

In Version 4.1 wurde ein weiterer Mechanismus eingeführt, um gestiegenen Sicherheitsanforderungen gerecht zu werden. Die älteren Versionen des Pairings werden weiterhin unterstützt. Abbildung 6 zeigt eine Übersicht der verfügbaren Pairing-Versionen und der von ihnen eingesetzten Algorithmen. Es ist zu beachten, dass sich die erzeugten Schlüssel in ihrer Länge nicht geändert haben.

Security Mechanism	Legacy	Secure Simple Pairing	Secure Connections
Encryption	E0	E0	AES-CCM
Authentication	SAFER+	SAFER+	HMAC-SHA256
Key Generation	SAFER+	P-192 ECDH HMAC-SHA-256	P-256 ECDH HMAC-SHA-256

Abbildung 6: Bluetooth Pairing Versionen (Tabelle 1.1 in [1])

Während des Pairings wird ebenfalls bestimmt, welche Rolle ein Gerät in einem Netz übernimmt. Das Gerät, das die Verbindung initiiert, ist immer der Master. Das Gerät, das dann beispielsweise einen Network-Access-Point bereitstellt, ist stets der Slave. Diese implizit zugewiesene Rolle kann jedoch später noch verändert werden.

6 Vergleich der verschiedenen Bluetooth-Versionen

Bluetooth 1.0a und 1.0b waren die ersten von der Bluetooth-SIG freigegebenen Versionen. Da diese noch viele schwerwiegende Probleme aufwiesen, wurde ein Jahr später die Version 1.1 veröffentlicht. Diese hatte in etwa den gleichen Funktionsumfang wie die Vorgängerversionen, war aber nicht abwärtskompatibel. Es wurden allerdings viele Probleme beseitigt.

Bluetooth 1.2 als Weiterentwicklung von Version 1.1 bringt neben der Beseitigung von kleineren Problemen auch neue Funktionen mit sich. Die wichtigsten Neuerungen sind eQoS (enhanced Quality of Service), eSCO (enhanced Synchron Connection Oriented) und AFH (Adaptive Frequency Hopping). Durch diese Zusätze sollen eine bessere Fehlererkennung und bessere Übertragungseigenschaften erreicht werden. Mit AFH soll außerdem die gleichzeitige Nutzung verschiedener Geräte im ISM-Band verbessert werden. Bluetooth 1.2 ist abwärtskompatibel zu Version 1.1.

Bluetooth 2.0 wurde am 08. November 2004 offiziell von der Bluetooth-SIG freigegeben. Als wichtigste Neuerungen sind die erhöhte Datenrate **E**nhanced **D**ata **R**ate (EDR)von bis zu 3 Mb/s, der verminderte Energieverbrauch und eine weiter verbesserte Fehlerkorrektur zu nennen. Auch Bluetooth 2.0 ist abwärtskompatibel zu den Versionen 1.2 und 1.1.

Mit Bluetooth 2.1 wurden August 2007 unter anderem neue Funktionen zum vereinfachten und sichereren Pairing (Secure Simple Pairing) spezifiziert. Weiterhin wurden Funktionen des Nahfunkstandard NFC (Near Field Communication) integriert. Hiermit kann zum Beispiel das bargeldlose Bezahlen von Fahrscheinen realisiert werden. Auch diese Revision des Funkstandards ist vollständig abwärtskompatibel zu den Versionen 2.0, 1.2 und 1.1.

Im April 2009 wurde die Bluetooth Spezifikation 3.0 veröffentlicht. Die wichtigsten Neuerungen sind hierbei das Enhanced Power Control, welches für ein effizienteres Energiemanagement sorgt und damit die Anzahl der energieverbrauchenden Disconnects reduziert. Mit dem Feature Unicast Connectionless Data können Daten ohne Berücksichtigung der L2CAP-Schicht übertragen werden - allerdings nur dann, wenn wenig Daten übertragen werden und eine geringe Latenzzeit wichtig ist. Eine besondere Neuerung stellt jedoch die Integration eines Highspeed-Kanals dar. Neben der

Bluetooth-Übertragungstechnik wurde der 802.11g-Standard integriert (ein Bluetooth 3.0+HS Dongle enthält daher sowohl einen Bluetooth- als auch einen WLAN-Chip). Der Bluetooth-Standard schließt nicht aus, dass ein externer WLAN-Chip genutzt werden kann, wenn dafür die nötige Treiberunterstützung vorhanden ist. Die Verbindung zwischen Protokollstapel und Funktechnik wurde aufgehoben, sodass der neu eingeführte Alternate MAC/PHY Manager zwischen den beiden Controllern wählen kann: Bei großen Datenmengen wird automatisch die WLAN-Technologie gemäß 802.11 verwendet und ansonsten immer die stromsparendere Bluetoothfunktechnik. Auch Bluetooth 3.0 ist abwärtskompatibel.

Die Version Bluetooth 4.0 wurde im Juni 2010 veröffentlicht und bietet einen deutlich verringerten Energieverbrauch sowie eine 128 Bit AES-Verschlüsselung. Da sich der Host die meiste Zeit im „Sleep“-Modus befindet, genügt eine Knopfzelle, um den Chip über Jahre hinweg mit Energie versorgen zu können. Allerdings gibt es in dieser Version nun zwei Gerätetypen: „Bluetooth Smart Ready“ ist abwärtskompatibel und findet sich in allen Geräten mit einer „Nutzeroberfläche“ (z.B. Handys, Notebooks) wieder. „Bluetooth Smart“ hingegen kann nur mit Geräten der Version 4.0 kommunizieren und besitzt keine direkte Nutzeroberfläche. Hauptsächlich Sensoren (vor allem im Sport- und medizinischen Bereich) werden damit ausgestattet. Durch die neue Star-Bus-Topologie kann zudem eine Vielzahl von Sensoren mit einem Master gleichzeitig kommunizieren.

Bluetooth 5.0 Standard wurde im Dezember 2016 verabschiedet. Geräte die diesem Standard entsprechen sind zu älteren Bluetooth-Versionen kompatibel. Alle Neuerungen dieser Version beziehen sich auf Bluetooth Low Energy. Um von Bluetooth 5.0 profitieren zu können, müssen alle Kommunikationsteilnehmer Bluetooth 5.0 beherrschen. Entsprechend der Evolution von Bluetooth wurden mit der neuen Bluetooth-Version in erster Linie Energieverbrauch, Durchsatz und Reichweite optimiert. Die Reichweite wurde gegenüber Bluetooth 4.0 von 50m auf bis zu 200m (Line of Sight) erhöht. Der Datendurchsatz wurde von 1Mb/s auf 2Mb/s erhöht.

7 Versuchsaufbau

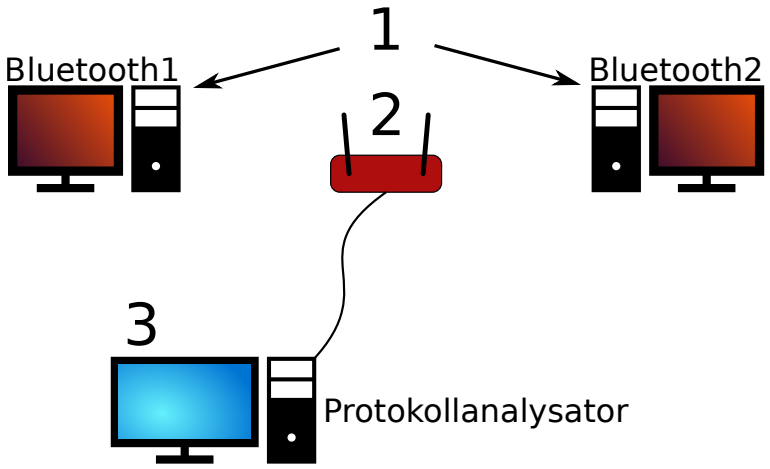

Abbildung 7: Schematische Darstellung des Versuchsaufbaus

Die beiden Rechner „bluetooth1“ und „blutooth2“ (1) können mit Bluetooth- und

WLAN-Geräten ausgestattet werden. Zwischen ihnen werden die Datenverbindungen aufgebaut. Beide Geräte verwenden ein Linux-Betriebssystem und sind über Kommandozeile konfigurierbar. Nach dem Start der Rechner stehen vier Fenster zur Kommandoeingabe und zum Starten verschiedener Programme bereit.

Ein manuelles Paaren der beiden Rechner beim Aufbau von Bluetooth-Verbindungen sowie eine PIN-Eingabe sind nicht nötig, da beide Geräte bereits entsprechend vorkonfiguriert sind.

7.1 Der Bluetooth Protokollanalysator

Der Bluetooth-Protokollanalysator BPA500 von Frontline Test Equipment ist zwischen den beiden Bluetooth-Rechnern (siehe Abbildung 7 (2)), positioniert. Die Bluetooth-Pakete werden an der Luftschnittstelle von der so genannten Comprobe mitgelesen und von der Software ausgewertet.

Zur Auswertung einer Bluetooth-Übertragung muss zunächst die Analysator-Software durch Doppelklick auf das „Frontline ComProbe Protocol Analysis System“-Icon auf dem Windows-Desktop (3) gestartet werden.
Als Synchronisationsmodus wird „Classic Only Single Connection“ ausgewählt.
Über den „Discover Devices“-Button kann nach in der Nähe befindlichen Geräten gesucht werden. Hierzu ist zunächst die Sichtbarkeit beider Geräte mit dem Kommando *hciconfig hci0 piscan* zu aktivieren, bevor nach Geräten gesucht wird. *bluetooth1* wird stets als „Master“ ausgewählt. Hierbei ist darauf zu achten, dass die richtigen Adressen in den Eingabefeldern stehen. Durch die Verwendung mehrerer Bluetooth-Dongles muss ein Eintrag, der den Namen *bluetooth1* trägt, nicht zwangsläufig zu dem aktuell eingesteckten Dongle gehören. Die letzten durch den Analysator gefundenen Geräte finden sie im Dropdown-Menu unter der Kategorie „last discovery“.

Eine Synchronisation kann lediglich während des Verbindungsaufbaus erfolgen. Hierzu ist als **erstes** der „Start Sniffing“-Button (im „datasource“-Fenster“) am Analyserechner zu betätigen. Erst **danach** ist die jeweilige Bluetooth-Verbindung zwischen den Bluetooth-Rechnern aufzubauen.
Hinweis: Eine bereits aufgebaute Bluetoothverbindung muss erst getrennt und anschließend neu initiiert werden, um eine Synchronisation mit dem Protokollanalysator zu gewährleisten. Bei einer erfolgreichen Synchronisation färbt sich das Icon im „datasource“-Fenster blau (siehe Abbildung 8).

Im Fenster „Frame Display“(Abbildung 9 Button im Hauptfenster) wird eine Liste aller empfangener Frames oben rechts dargestellt (Rot). Im unteren Bereich sehen sie den Inhalt des markierten Frames in den Darstellungsformen Binär, Hexadezimal und Character (Blau). Auf der linken Seite sind die Daten der einzelnen Protokolle dargestellt die dieser Frame enthält (Grün).

Der „Packet Error Rate Statistics“-Button im Hauptfenster öffnet das Fenster „Packet Error Rate by Channel Number“, welches eine Übersicht über die Anzahl empfangener Pakete und Datenmengen gibt. Dabei wird unter anderem nach verschiedenen Framegrößen, Durchsatzraten und fehlerhafter bzw. erneut gesendeter Frames aufgeschlüsselt.
Das Fenster enthält Statistiken zu jedem der 79 RF-Kanäle. Durch einen Mausklick

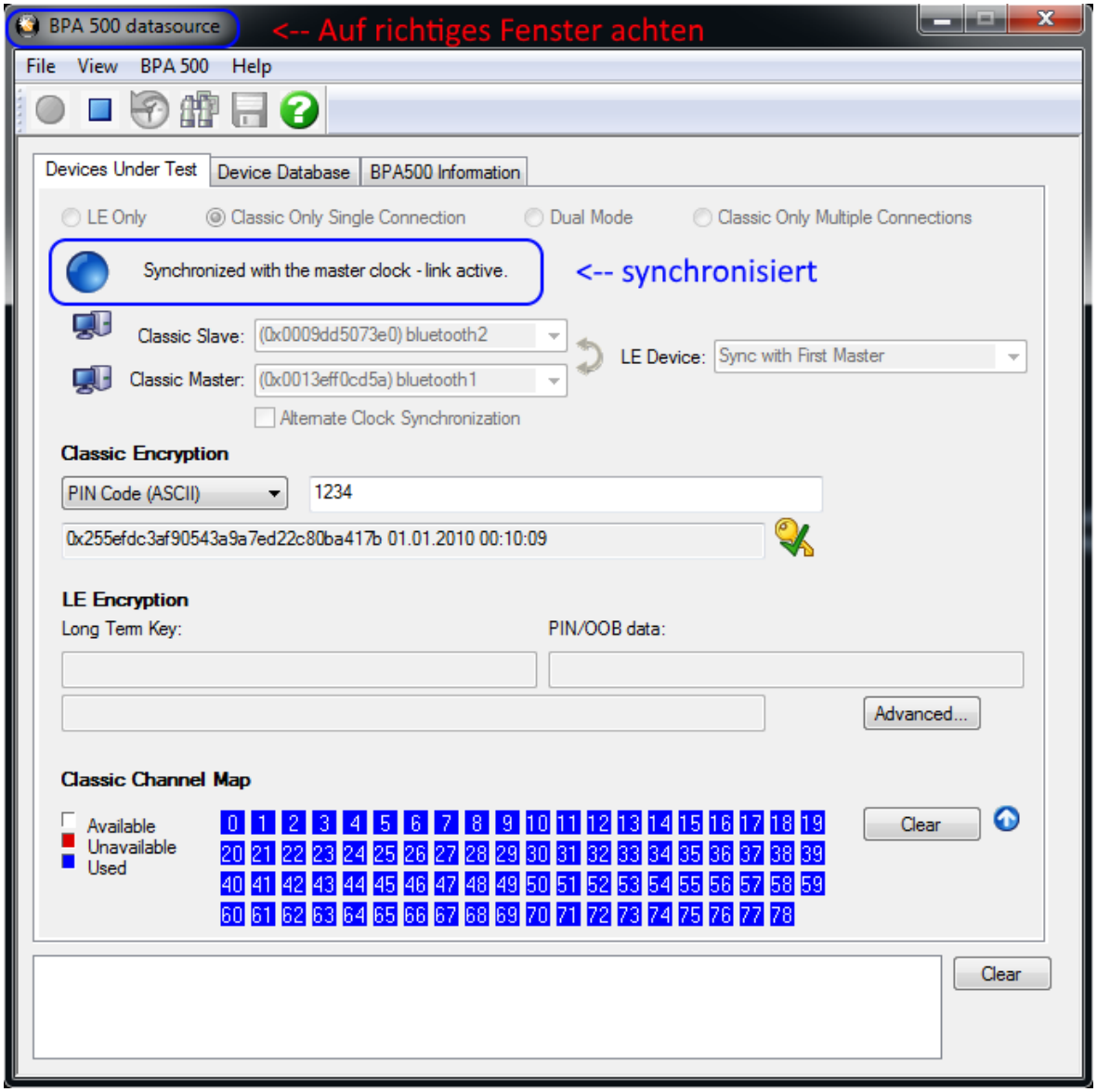

Abbildung 8: Datasource Fenster nach Synchronisation

auf eines der Balkendiagramme oder des „Total“-Tortendiagramms, werden detaillierte Informationen zum jeweiligen RF-Kanal angezeigt.

Die **M**essage **S**equence **C**hart (MSC)-Ansicht im Hauptfenster zeigt den Ablauf einer Verbindung grafisch aufbereitet an und ist somit gut geeignet, um das Verhalten bei einem Verbindungsaufbau oder „feature-request“ zu analysieren. Für eine bessere Übersichtlichkeit kann ausgewählt werden, welche Protokollebenen angezeigt werden sollen.

Hinweis: Die Fenster MSC, „Statistics Display“ und „Frame Display“ werden von Windows in der Taskleiste nicht als mehrere Fenster dargestellt, wenn sie Schwierigkeiten haben ein gewünschtes Fenster zu finden minimieren sie ihr aktuelles Fenster.

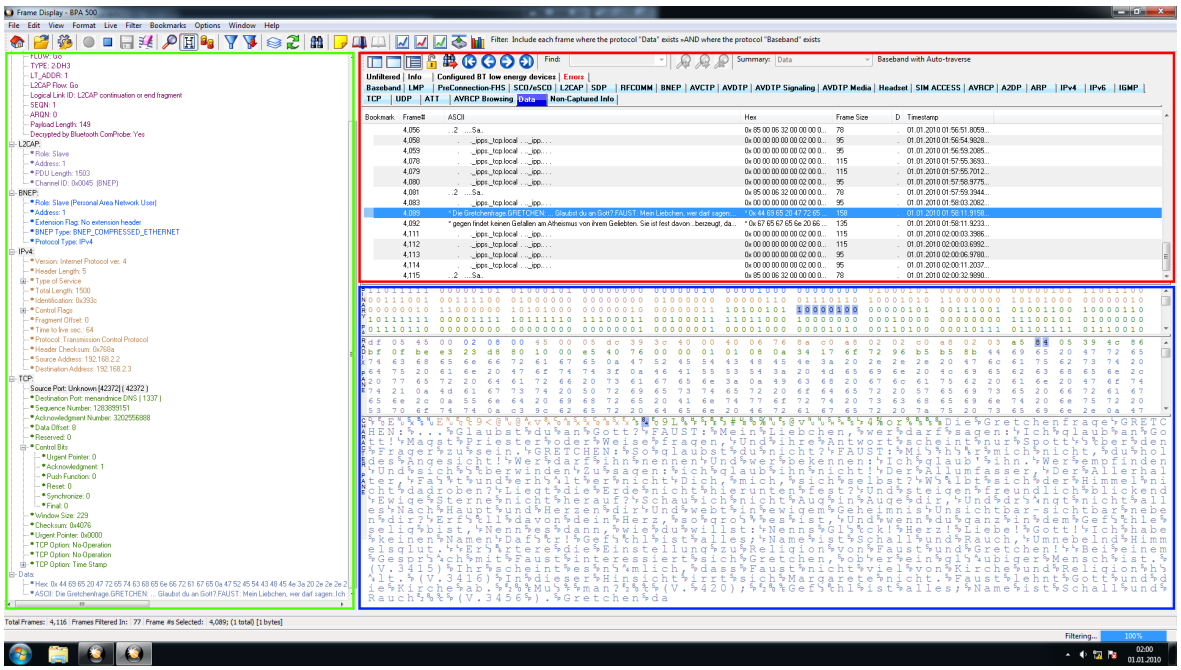

Abbildung 9: Unterteilung des Frame Displays

7.2 Erklärung zu den verwendeten Kommandos

- **sdptool**
  Über sdptool können mit der Option „add“ Dienste zur Anzeige durch das SDP hinzugefügt werden. Die Option „browse“ ermöglicht das Abfragen der von einem Gerät angebotenen Dienste.

- **rfcomm**
  Mit diesem Programm lassen sich serielle Bluetooth-Verbindungen herstellen. Dabei kann der Dienst mit „listen“ angeboten bzw. mit „connect“ in Anspruch genommen werden.

- **hcitool**
  Die Konfiguration bzw. die Anzeige von Informationen zu Bluetooth-Geräten und -Verbindungen wird durch „hcitool“ ermöglicht. Die Option „con“ zeigt aktuelle Verbindungen mit der Adresse des Partners, der Rolle und des Übertragungsmodus an. Mit „scan“ können in Reichweite befindliche Geräte gesucht werden.

- **cat**
  Dieses kleine Werkzeug gibt den Inhalt von Dateien aus. Durch Umleitungsoperatoren (<, >, |) kann der so erzeugte Datenstrom weiterverarbeitet oder in andere Programme eingespeist werden.

- **iperf**
  „iperf“ ist ein Hilfsmittel zur Messung der maximalen Datenraten von Netzwerkverbindungen. Dazu werden zehn Sekunden lang Daten gesendet und anschließend aus der Datenmenge und der Zeit die Datenrate berechnet. Die Datenübertragung erfolgt über TCP. Die gemessene Datenrate basiert auf TCP-Nutzdaten, also ohne Header.

- **nc**
  Mit „nc“ ist der Aufbau von Netzwerkverbindungen möglich. Mit der Option „-l 1337“ wird auf TCP-Port 1337 auf eingehende Verbindungen gewartet. Über „nc <hostname> <port>“ wird eine Verbindung zu einem wartenden Rechner hergestellt. Daten, die übertragen werden sollen, können mit „cat“ eingespeist werden.

- **./NAPServer.sh**
  Dieses Skript erzeugt einen NAP und nimmt eine Verbindungsanfrage an. Ebenso wird die Sichtbarkeit des Bloetooth-Gerätes aktiviert und die Verschlüsselung abgeschaltet.

- **./NAPClient.sh**
  Dieses Skript erzeugt eine Verbindung zu einem **N**etwork **A**ccess **p**oint (NAP) auf einem anderen Rechner.

7.3 Hinweise zur PC-Bedienung

Nachdem Sie sich auf den Linux-Rechnern eingeloggt haben, finden Sie vier Terminals vor. Alle Terminals können gleichermaßen zur Eingabe von Kommandos genutzt werden.

Kommandos, die bereits einmal in einem Fenster eingegeben wurden, können mit der UP-Cursortaste wieder zurückgeholt werden.

Falls nicht anders angegeben, können laufende Programme mit der Tastenkombination STRG+C beendet werden.

Bei den Aufgaben mit Durchsatzmessung (Verwendung des iperf-Kommandos) ist es wichtig, dass der iperf-Server (welcher die Verbindung anbietet, also die Option **-s** beinhaltet) auch die **eigene** IP-Adresse verwendet:
bluetooth1bt und *bluetooth1wlan* sind hierbei Hostnamen für die IP-Adressen des Rechners „bluetooth1“. *bluetooth2bt* und *bluetooth2wlan* sind Hostnamen für die IP-Adressen des Rechners „bluetooth2“. Auf Client-Seite wird anschließend auf die zur Verfügung gestellte IP-Adresse zugegriffen.

Damit die Bluetooth- und die WLAN-Dongles an den vorderen beiden USB-Buchsen nebeneinander Platz finden, sollten Sie für die WLAN-Dongles die beiliegenden USB-Verlängerungskabel verwenden.

Falls die Bluetooth-Geräte nicht vom Protokollanalysator gefunden werden, so kann man alternativ auch die Bluetooth-Adressen in die entsprechende Zeile der Analysesoftware eintippen (Geräteadressen über „hcitool dev“ ermitteln).

Die zum kopieren und einfügen bekannten Standard-Hotkeys STRG+C und STRG+V funktionieren in den Terminals nicht, hier muss stattdessen STRG+SHIFT+C und STRG+SHIFT+V verwendet werden. Es ist auch möglich, einen Text in einem Terminal zu markieren und dann in einem anderen Terminal die mittlere Maustaste zu

drücken, um den markierten Text an dieser Stelle einzufügen.

7.4 Übersicht der Bluetooth-Dongles sowie WLAN-Dongles

- Bluetooth 2.0 EDR USB-Dongles

- Bluetooth 3.0 +HS USB-Dongles

- WLAN-Dongles

8 Vorbereitungsaufgaben

1. Welche der folgenden Bluetooth-Anwendungen würden Sie im ACL- bzw. SCO-Übertragungsmodus betreiben?
   - Netzwerkverbindung zwischen vier Notebooks
   - IP-Telefonie über Headset und PC
   - Datentransfer zwischen Tablet und Smartphone

2. Die maximale Reichweite ihnen zur Verfügung stehender Bluetooth-Geräte ist auf ca. 100 m beschränkt. Erläutern Sie Möglichkeiten,wie der Abstand zweier Kommunikationsteilnehmer bis Bluetooth-Version 3.0 erhöht werden könnte.

3. Die gleichzeitige Nutzung von WLAN (IEEE 802.11b/g/n) und Bluetooth ist gerade in Notebooks üblich. Da beide Standards das ISM-Band zur drahtlosen Datenübertragung nutzen, ist eine gegenseitige Beeinflussung kaum zu vermeiden. Im Gegensatz zu Bluetooth (mit FHSS bzw. AFH) belegt WLAN einen Kanal mit einer Bandbreite von 5 MHz dauerhaft.

   Welches der beiden Systeme wird Ihrer Meinung nach bei gleichzeitiger Nutzung stärker gestört? (mit Bluetooth Version 1.1)

4. Welchen Vorteil bringt die Standardisierung von Bluetooth-Profilen für bestimmte Anwendungsfälle (z.B. normale Funkmäuse) gegenüber anderen Systemen für Hersteller sowie Kunden?

5. Warum ist es möglich, die meisten Notebooks, welche mit Bluetooth v2.1 ausgestattet sind, mittels Softwareupdate auf v3.0+HS aufzurüsten (also ohne neue Hardware verbauen zu müssen)?

6. Warum wird der neue Bluetooth 4.0 Standard vor allem für Sensoren (z.B. zur Herzratenmessung) verwendet und mit welchen anderen Bluetooth-Versionen kann dieser kommunizieren?

9 Häufig verwendete Kommandos

Diese Übersicht soll als Nachschlagewerk für häufig verwendete Kommandos und Vorgehensweisen während der Durchführung der Praktikumsaufgaben dienen.

- **eigene Geräteadresse ermitteln:**
  bluetooth1/2: `hcitool dev`

- **RFCOMM-Verbindung aufbauen:**
  bluetooth2: `sdptool add SP`
  bluetooth2: `rfcomm listen 0 1`
  bluetooth1: `rfcomm -r connect 0 <BD-ADDR> 1`

- **BNEP-Verbindung aufbauen:**
  bluetooth2: `./NAPServer.sh`
  bluetooth1: `./NAPClient.sh`

- **Durchsatzmessung Bluetooth (funktioniert nur bei aktiver BNEP-Verbindung!)**
  bluetooth2: `iperf -s -B bluetooth2bt`
  bluetooth1: `iperf -c bluetooth2bt`

- **Durchsatzmessung WLAN**
  bluetooth1: `iperf -s -B bluetooth1wlan`
  bluetooth2: `iperf -c bluetooth1wlan`

- **Analysesoftware synchronisieren (Mitsniffen der Bluetooth-Verbindung)**
  bluetooth1/2: `hciconfig hci0 piscan`
  In der Analysesoftware über „Discover Devices“ nach den Geräten suchen.
  „bluetooth1“ als Master und „bluetooth2“ als Slave auswählen.
  „Start Sniffing“-Button aktivieren.
  Bluetooth-Verbindung (RFCOMM oder BNEP) herstellen.
  Darauf achten, dass das Icon in der Analysesoftware blau erscheint.

10 Praktikumsaufgaben

Bitte lesen Sie sich jede Aufgabe erst ganz durch, **bevor** Sie mit der Durchführung beginnen. Sollten während der Versuchsdurchführung Probleme auftreten, so ist der Betreuer jederzeit telefonisch zu erreichen. Fragen, die sie während der Versuchsdurchführung stellen, fließen nicht in die Bewertung mit ein, stellen sie diese also am besten vor dem Testtatgespräch. (Hinweis: Im Folgenden stellt der Ausdruck **<BD-ADDR>** einen Platzhalter für die entfernte Bluetooth-Geräteadresse dar.)

1. **Eigenschaften einer RFCOMM-Verbindung**
   Starten Sie die beiden Linux-Rechner „bluetooth1“ und „bluetooth2“ sowie den Windows-Rechner „bluetooth3“. Die Logindaten erhalten Sie vom Betreuer.

   Stecken Sie die Bluetooth-2.0-Dongles in die USB-Buchsen der Rechner „bluetooth1“ und „bluetooth2“. Ermitteln Sie nun die lokale Bluetooth-Adresse, aktivieren Sie die Sichtbarkeit des lokalen Gerätes (Page & Inquiry Scan aktivieren) und suchen Sie nach entfernten Bluetoothgeräten mittels folgender Kommandos:

   bluetooth1/2: hcitool dev
   bluetooth1/2: hciconfig hci0 piscan
   bluetooth1/2: hcitool scan
   (sollte kein Gerät gefunden werden, wiederholen sie diesen Befehl)

   Optionale Aufgabe:
   Wenn Sie ein Bluetooth-fähiges Gerät (Handy, Smartphone o.ä.) bei sich haben, dann lassen Sie sich die Dienste, die ihr Bluetooth-Gerät anbietet, anzeigen. Schalten Sie Bluetooth auf Ihrem Gerät ein und aktivieren Sie die Sichtbarkeit. Als nächstes wird die Bluetooth-Geräteadresse benötigt: Falls Sie diese nicht finden können, so führen Sie am Linux-Rechner erneut einen Scan aus:

   bluetooth1/2: hcitool scan

   Der Scan sollte die Adresse Ihres eigenen Geräts liefern. Geben Sie nun folgenden Befehl in die Konsole ein:

   bluetooth1/2: sdptool browse <BD-ADDR-Ihres-Gerätes>

   1.1 (optional) Welche Profile werden angezeigt und wozu dienen Sie?
   (Hinweis: Es ist nicht nötig alle Profile exakt zu identifizieren. In den meisten Fällen gibt aber bereits der Name eines Profils Aufschluss über dessen Funktion.)

   Starten Sie das Programm „Frontline ComProbe Protocol Analysis System“ auf dem Analyserechner. Wählen Sie „Combined Connection“ aus.
   Über den „Discover Devices“-Button kann nach in der Nähe befindlichen Geräten gesucht werden. Im „Datasource“ Fenster können die gerade gefundenen Geräte nun ausgewählt werden („bluetooth1“ ist Master, auch auf die Adresse

achten nicht nur auf den Namen). Sind die richtigen Geräte ausgewählt betätigen sie den „Start Sniffing“-Button  , um die Aufzeichnung zu starten.

Stellen Sie nun eine serielle RFCOMM-Verbindung zwischen den Rechnern „bluetooth1“ und „bluetooth2“ her. Geben Sie dazu die nachfolgenden Kommandos in die Eingabeaufforderungen der Bluetooth-Rechner ein.

bluetooth2: `sdptool add SP`

Durch das letzte Kommando wurde der Serial-Port-Service durch das SDP hinzugefügt. Fahren Sie nun mit dem Herstellen der RFCOMM-Verbindung fort:

bluetooth2: `rfcomm listen 0 1`
(Wartet auf eingehende RFCOMM-Verbindungen auf serieller Schnittstelle 0 und Kanal 1)
bluetooth1: `rfcomm -r connect 0 <BD-ADDR> 1`
(Verbindet mit dem Bluetooth-Gerät <BD-ADDR>. <BD-ADDR> ist dabei die sechs Byte lange Geräteadresse.)

Nun sollte sich das Icon oben im „datasource“ Fenster blau gefärbt haben. Falls die Synchronisation nicht erfolgreich war (Icon bleibt grün oder rot), beenden Sie auf beiden bluetooth-PCs die RFCOMM-Verbindung und betätigen den „Stop Sniffing“-Button des Protokollanalysators. Anschließend wiederholen Sie die oben beschriebene Prozedur solange, bis die Synchronisation erfolgreich war (Icon wird blau).

Wird bei „Connections“ keine aktive Verbindung angezeigt, so können sie allgemein (auch bei BNEP-Verbindungen) die Dongles entfernen, fünf Sekunden warten und diese wieder einstecken. Danach können Sie einen erneuten Versuch zum Verbindungsaufbau durchführen.

1.2 Nun ist eine serielle Verbindung hergestellt, durch welche Terminalausgabe ist dies zu erkennen?

Die Terminals, in denen Sie gerade die RFCOMM-Verbindung hergestellt haben, können erst wieder zur Befehlseingabe genutzt werden, nachdem dort die RFCOMM-Verbindung getrennt wurde. Da die folgenden Aufgaben jedoch eine bestehende RFCOMM-Verbindung voraussetzen, verwenden Sie nun bitte ein anderes Terminal:

1.3 Welcher Übertagungsmodus wird für RFCOMM genutzt?

bluetooth1/2: `hcitool con`
(Zeigt alle aktiven Bluetooth-Verbindungen und deren Übertragungsmodus an)

Nutzen Sie die serielle Schnittstelle um den Inhalt einer Textdatei zu übertragen (Die Datei „faust“ befindet sich bereits im /home-Verzeichnis):

bluetooth2: `cat /dev/rfcomm0`
(nach der Übertragung mit STRG+C beenden)
bluetooth1: `cat faust > /dev/rfcomm0`
(nach der Übertragung mit STRG+C beenden)

(Hinweis: Wenn die Analysator-Software fragt, welches Protokoll von RFCOMM getragen wird, antworten sie mit „RFCOMM is carrying – Raw Data – . . . “. Diese Frage tritt immer dann auf, wenn der Analysator den Inhalt von Paketen nicht eindeutig zuordnen kann.)

(Hinweis: Stoppen sie nach dem Senden die Aufzeichnung in der Analysatorsoftware um unnötige Aufzeichnungen zu vermeiden)

1.4 Ermitteln Sie die Protokollhierarchie für die Übertragung eines RFCOMM-Datenpakets.
Nutzen sie hierzu Abbildung 1 aus der Vorbereitung. Im Fenster „Framedisplay“ des Protokollanalysators werden auf der linken Seite die einzelnen Protokolldaten dargestellt (siehe Abbildung 9). Hierzu müssen sie in der Liste der empfangenen Frames (oben rechts im Framedisplay) ein beliebiges RFCOMM-Frame auswählen.

1.5 Welche weiteren Protokolle, die nicht zur RFCOMM-Verbindung gehören, werden angezeigt? Wozu dienen sie?
Beschränken sie sich hierbei auf Protokolle, die auch in Abbildung 1 enthalten sind.
(Wählen sie dazu den Karteireiter „unfiltered“)

1.6 Wie viele Pakete sind nötig um den Text aus der Datei „faust.txt“ komplett zu übertragen?
(Hinweis: Nutzen Sie hierfür das „Frame Display“ mit der Karteikarte „RFCOMM“. Dort werden die Nutzdaten in der Spalte „Payload“ angezeigt: Genauer gesagt wird hier der *Inhalt* jedes Datenpakets dargestellt, d.h. Reintext wird also auch als solcher angezeigt. Hierbei ist auf die Darstellung von Sonderzeichen zu achten). Jede Zeile entspricht einem Datenpaket. Um eine komplette Darstellung des Payloads zu erhalten, klicken sie den gewünschten Frame in der Liste an, suchen sie anschließend den Payload auf der linken Seite und klicken sie ihn an. Speichern sie ihre Aufzeichnung um diese später dem Betreuer zu zeigen.

2. Eigenschaften einer BNEP-Verbindung

Nutzen Sie wieder den Protokollanalysator für die Durchführung der folgenden Aufgaben. Für eine bessere Übersichtlichkeit löschen Sie den Puffer im Protokollanalysator durch Klick auf den Button „Clear“  . Lassen Sie nun wieder den Protokollanalysator mitlesen.

Beenden Sie die RFCOMM-Verbindung und stellen Sie eine BNEP-Netzwerkverbindung zwischen den Rechnern „bluetooth1“ und „bluetooth2“ her. Geben Sie dazu die nachfolgenden Kommandos in die Eingabeaufforderung ein. „bluetooth2“ soll als Server und „bluetooth1“ als Client fungieren.

Hierzu muss am Analyserechner erneut das „Sniffing“ gestartet werden, da nur während des Verbindungsaufbaus die Synchronisation erfolgen kann.

bluetooth2: ./NAPServer.sh
(Erzeugt einen NAP und richtet die BNEP-Verbindung ein)

bluetooth1: ./NAPClient.sh
(Verbindet sich zum NAP auf Bluetooth2)

Beim Verbindungsaufbau durch den Client werden ihnen die notwendigen Einstellungen für die Analysesoftware noch einmal angezeigt. Verifizieren sie ihre Einstellungen und drücken sie „Enter“ um fortzufahren, wenn der Protokollanalysator aktiv ist. Warten sie bis auf beiden Rechnern wieder der Eingabeprompt („#“) erscheint. Überprüfen Sie, ob die BNEP-Verbindung erfolgreich hergestellt wurde durch folgendes Kommando:

bluetooth1/2: hcitool con

(2.1) Welcher Übertragungsmodus wird für BNEP genutzt?

Mit folgenden Kommandos können Dateiinhalte über die Netzwerkverbindung übertragen werden:

bluetooth2: nc -l 1337
bluetooth1: cat faust | nc bluetooth2bt 1337
(Das Zeichen | ist auf der Tastatur mittels“Alt Gr“+“<“(links neben Y) zu erreichen.)

(Hinweis: Beenden sie auch hier wieder die Aufzeichnung am Analyserechner, sobald sie die Datei übertragen haben.)

2.2 Ermitteln Sie die Protokollhierarchie für die Übertragung eines IP-Datenpakets und ordnen Sie die einzelnen Protokolle in das TCP/IP-Referenzmodell ein.
2.3 Wie viele TCP-Pakete sind nötig, um den Text aus der Datei „faust“ komplett zu übertragen? Speichern sie ihre Aufzeichnung! Sie müssen ihrem Betreuer die von ihnen gefundenen Datenpakete zeigen. Das vorgehen wird nachfolgend beschrieben.
Finden der Pakete: Hierzu können sie auf der linken Seite im „Frame Display“ unter „Data“ den Inhalt eines Frames einsehen, um somit die für die Übertragung der Datei benötigten Frames ausfindig zu machen. Oben rechts wird das anzuzeigende Frame ausgewählt, sie sollten in der liste empfangener frames den den Reiter „TCP“ auswählen um die richtigen frames zu finden

Speichern der Aufzeichnung: Um die Aufzeichnung speichern zu können, muss diese zuerst beendet werden. Klicken sie hierfür auf den Stopp Button im Framedisplay oben links. Nun können sie unter *File* → *Save* Ihre Aufzeichnung speichern.

2.4 Durch welches Protokoll werden die Hardware-Adressen auf IP-Adressen gemappt?

3. **Durchsatz Bluetooth 2.0**

Rufen sie mit klick auf  das „Packet Error Rate Statistics"-Fenster auf und klicken sie unten rechts auf „Reset". Somit werden ihre folgenden Messungen nicht durch die bisherige Aufzeichnung verfälscht.

3.1Lassen Sie während der folgenden Messungen den Bluetooth-Analysator mitlaufen und notieren Sie aus dem „Packet Error Rate by Channel Number"-Fenster den prozentualen Anteil wiederholt gesendeter Frames (retransmitted Frames).

Starten Sie auf „bluetooth2" den iperf-Server zur Durchsatzmessung durch folgendes Kommando:

bluetooth2: `iperf -s -B bluetooth2bt`

(bei mehreren Messungen muss dieser Befehl nur einmal eingegeben werden da der Server nach einer Messung nicht automatisch abgeschaltet wird)

Die Durchsatzmessung (bzw. Erzeugung von Datenverkehr) kann durch den iperf-Client auf „bluetooth1" gestartet werden:

bluetooth1: `iperf -c bluetooth2bt`

Nach etwa 10 Sekunden wird unter anderem die gemessene Datenrate ausgegeben.

Hinweis: Bei den Messungen mit iperf sind nur die Werte des Clients ausschlaggebend. Der Client ist in diesem Fall bluetooth1.

3.2 Wiederholen Sie die Messung fünf mal und notieren Sie sich die Datenraten auf der Client-Seite.

3.3 Wie hoch ist die ermittelte Datenrate und wie erklären Sie sich die Diskrepanz zwischen dem in der Vorbereitung angegebenen Maximalwert und Ihrem Messwert?

Stecken Sie nun die WLAN-Dongles in die USB-Buchsen neben den Bluetooth-Dongles.(Nutzen sie hierfür bitte die Verlängerungskabel, um Beschädigungen an den Dongles zu vermeiden.) Starten Sie auf „bluetooth1" den „iperf"-Server, der auf eine WLAN-Verbindung wartet und führen Sie eine Durchsatzmessung über

WLAN durch. Verwenden Sie nun folgende Eingaben:

bluetooth1: iperf -s -B bluetooth1wlan
bluetooth2: iperf -c bluetooth1wlan

3.4 Wiederholen Sie auch diese Messung fünf mal und notieren Sie sich die Durchsatzraten des Clients.

Führen Sie nun die Duschsatzmessungen der Bluetooth- und WLAN-Verbindung **gleichzeitig** durch. Dies ist der Fall mit maximaler Störung, da bei „bluetooth2“ über WLAN empfangen und gleichzeitig über Bluetooth gesendet wird. Löschen Sie vorher wieder sämtlichen Puffer in der Analysesoftware und achten Sie darauf, dass die Analysesoftware die Verbindung mitsnifft. Die beiden Messungen sollten nun möglichst synchron gestartet werden!

3.5 Notieren Sie sich wieder fünf Werte und ermitteln Sie im Anschluss die prozentuale Rate der erneut gesendeten Frames.

4. **Durchsatz unter Bluetooth 3.0**
Entfernen Sie die WLAN-Dongles sowie Bluetooth 2.0 Dongles und stecken Sie die Bluetooth 3.0+HS Dongles ein.
Führen Sie die Synchronisierung mit der Analysesoftware durch (vorher wieder Puffer löschen!).
4.1 Stellen Sie eine BNEP-Verbindung her und führen Sie ein weiteres Mal die Durchsatzmessung mittels iperf (Nur Bluetooth) wie in Aufgabe 4 durch.

4.2 Wie wirken sich die zwei unterschiedlichen Bluetooth-Versionen auf die Durchsatzrate aus? Was hätten sie erwartet?

5. **Role-Switch und Feature Request**
Beenden sie die BNEP-Verbindung indem sie **beide** Bluetooth-Dongles entfernen und sie erneut einstecken. Löschen sie den Puffer der Analysesoftware und bauen sie erneut eine BNEP-Verbindung zwischen den Rechnern auf, diesmal werden jedoch ihre Rollen vertauscht. (Bluetooth1 stellt den NAP zur Verfügung und Bluetooth2 verbindet sich). Dazu müssen sie natürlich die Einträge in der Analysesoftware anpassen, damit sich der Analysator synchronisieren kann.

bluetooth1: `./NAPServer.sh`
bluetooth2: `./NAPClient.sh`

Nach dem Verbindungsaufbau können sie das Sniffing beenden. Hierzu klicken sie im Datasource-Fenster auf den blauen „Stop Sniffing“-Button. Zur Auswertung öffnen sie den Message Sequence Chart () und suchen dort nach den Lösungen für Aufgabe 5.1 und 5.2. Lassen sie das Fenster mit dem MSC geöffnet, da dies im Testatgespräch benötigt wird. (wichtige Ereignisse bzw. Zustandsänderungen werden im MSC als Hinweise in Lila eingeblendet.)

5.1 Sind die Master und Slave Rollen nach einem Verbindungsaufbau änderbar?

5.2 Zeigen sie, wie sich die Kommunikationspartner ihre unterstützten Features mitteilen.

5.3 Wozu ist dieses Verhalten nützlich? Nennen sie ein Beispiel.

Bitte lassen Sie die Rechner nach Durchführung der Aufgaben in ihrem aktuellen Zustand. Das Herunterfahren wird vom Betreuer übernommen.

ACL **A**synchron **C**onnection-**L**ess

AFH **A**daptive **F**requency **H**opping

BNEP **B**luetooth **N**etwork **E**ncapsulation **P**rotocol

DSL **D**igital **S**ubscriber **L**ine

EDR **E**nhanced **D**ata **R**ate

eQoS **e**nhanced **Q**ality **o**f **S**ervice

eSCO **e**nhanced **S**ynchronous **C**onnection **O**riented

FEC **F**orward **E**rror **C**orrection

FHSS **F**requency **H**opping **S**pread **S**pectrum

GAP **G**eneric **A**ccess **P**rofile

HCI **H**ost **C**ontroller **I**nterface

HS **H**igh **S**peed

I/O **I**nput / **O**utput

ICMP **I**nternet **C**ontrol **M**essage **P**rotocol

IP **I**nternet **P**rotocol

ISDN **I**ntegrated **S**ervices **D**igital **N**etwork

ISM **I**ndustrial **S**cientific **M**edical

ISO **I**nternational **S**tandard **O**rganisation

L2CAP **L**ogical **L**ink **C**ontrol and **A**daption **P**rotocol

LAN **L**ocal **A**rea **N**etwork

LMP **L**ink **M**anager **P**rotocol

MAC **M**edia **A**ccess **C**ontrol

MSC **M**essage **S**equence **C**hart

NAP **N**etwork **A**ccess **p**oint

OBEX **Ob**ject **Ex**change Protocol

OSI **O**pen **S**ystem **I**nterconnection

PAN **P**ersonal **A**rea **N**etwork

PC **P**ersonal **C**omputer

PCMCIA **P**ersonal **C**omputer **M**emory **C**ard **I**nternational **A**ssociation

PDA **P**ersonal **D**igital **A**ssistant

PIN **P**ersonal **I**dentification **N**umber

RF **R**adio **F**requency

RFCOMM **R**adio **F**requency **Com**munication Port e**m**ulation

SCO **S**ynchronous **C**onnection-**O**riented

SDP **S**ervice **D**iscovery **P**rotocol

SIG **S**pecial **I**nterest **G**roup

SPP **S**erial **P**ort **P**rofile

SSP **S**ecure **S**imple **P**airing

TCP **T**ransmission **C**ontrol **P**rotocol

TCS **T**elephony **C**ontrol **S**pecification

TDD **T**ime **D**ivison **D**uplex

UDP **U**ser **D**atagram **P**rotocol

USB **U**niversal **S**erial **B**us

WLAN **W**ireless **L**ocal **A**rea **N**etwork

Literatur

[1] Bluetooth SIG: *Core Specification 5.0, 2016*
https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=421043

[2] Bluetooth SIG: *archived Specifications, 2019*
https://www.bluetooth.com/specifications/archived-specifications

[3] Bluetooth SIG: *RFCOMM Specification 1.2, 2012*
https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=263754

[4] Holger Hildebrandt und Kay Pein: *Studienarbeit Bluetooth-Anwendungen, 2004*
http://midas1.e-technik.tu-ilmenau.de/~webkn/Abschlussarbeiten/Studienarbeiten/sta_hildebrandt+Pein.pdf

[5] Bundesnetzagentur: *Funkanwendungen auf den ISM-Bändern, 2015*
https://emf3.bundesnetzagentur.de/pdf/ISM-BNetzA.pdf

[6] Roger Marks and Ian C. Gifford and Bob O'Hara: *Standards in IEEE 802 Unleash the Wireless Internet | NIST, 2001*
https://www.semanticscholar.org/paper/Standards-in-IEEE-802-Unleash-the-Wireless
03e40d3087b9164cb7bd6a10cddfa69f1c35ab80

[7] N. Golmie, N. Chevrollier and O. Rebala: *Bluetooth and WLAN coexistence: challenges and solutions, 2003*
https://ieeexplore.ieee.org/abstract/document/1265849

[8] Lars Grenzendörfer: *Seminararbeit Untersuchungen zu Bluetooth, 2002*
http://midas1.e-technik.tu-ilmenau.de/~webkn/Abschlussarbeiten/Hauptseminararbeiten/hs_grenzendoerfer.pdf

[9] BlueZ Project: *Official Linux Bluetooth protocol stack, 2019*
http://www.bluez.org

[10] ESnet / Lawrence Berkeley National Laboratory: *Iperf, 2019*
https://github.com/esnet/iperf

[11] Giovanni Giacobbi: *Official GNU Netcat Project, 2004*
http://netcat.sourceforge.net/

[12] Mikhail Galeev: *Bluetooth 4.0: An introduction to Bluetooth Low Energy, 2011*
http://www.eetimes.com/design/communications-design/4218319/Bluetooth-4-0--An-introduction-to-Bluetooth-Low-Energy-Part-II

[13] DigitalThink, Inc.: *Star bus topology*
http://thought1.net/nt100/module3/star_bus.html