

STUDIENARBEIT

**Überprüfung der Verwendbarkeit des Openbeacon USB Stick
als Software Radio**

Marcel Berg



Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät II
Institut für Informatik
Lehrstuhl für Systemarchitektur

Gutachter:

.....

.....

Betreuer: Robert Sombrutzki

Berlin, 15. Januar 2013

Zusammenfassung

Durch Software Defined Radio (SDR) können neue Netzwerkprotokolle leicht und schnell umgesetzt und auf ihre Leistungsfähigkeit überprüft werden, da keine teure und aufwändige Umsetzung in Hardware nötig ist. Dies erleichtert deutlich die Erforschung von Übertragungstechnologien in drahtlosen Netzwerken. Das Ziel dieser Arbeit ist die Evaluierung des Openbeacon USB-Stick und die Untersuchung, ob sich dieser als preiswertes SDR verwenden lässt. Zu diesem Zweck wurde er in das Testsystem (Humboldt Wireless Lab) integriert. Im Vergleich zu gängiger WLAN Hardware sind Modifikationen des Media Access Control Layer beim Openbeacon USB Stick leicht möglich. Diese verschaffen allerdings nur wenig Spielraum für Forschung, dennoch lässt er sich als kostengünstiges SDR in begrenztem Maße einsetzen. Im Rahmen der Arbeit wurde ein einfaches MAC-Protokoll implementiert und evaluiert. Der erreichte Durchsatz war mit bis zu 21 Kbit/s gering. Der Openbeacon ist somit vorrangig für Anwendungen mit geringer Datenrate geeignet.

Schlagwörter: Openbeacon, Click, Multiradio, Cognitiv Radio

Inhaltsverzeichnis

1	Einleitung	3
1.1	Aufgabenstellung	5
1.2	Aufbau der Arbeit	5
2	Einführung	6
2.1	Software Defined Radio	8
2.2	Radio Frequency Identification	9
3	Plattform	10
3.1	Openbeacon	11
3.2	Entwicklungsumgebung	13
3.3	Click Router API	13
4	Implementierung	15
4.1	Openbeacon-Firmware	16
4.2	Openbeacon Daemon	18
4.3	Click	19
5	Evaluation	21
5.1	USB Schnittstelle	22
5.2	drahtlose Schnittstelle	24
5.3	Ende-zu-Ende	30
5.4	Indoor	34
6	Fazit	37
7	Anhang	39
	Abbildungsverzeichnis	42
	Tabellenverzeichnis	43
	Literaturverzeichnis	44

Kapitel 1

Einleitung

Die drahtlose Kommunikation ist ein aktuelles und weltweites Forschungsgebiet. Der Aufbau (bzw. Wartung) von drahtlosen Kommunikationsinfrastrukturen ist, im Vergleich zu drahtgebundener Kommunikation, schneller und kostengünstiger. Dabei beschäftigt sich die Forschung mit verschiedenen Problemen, z.B. die Unzuverlässigkeit und die begrenzte Bandbreite in drahtlosen Netzen.

Ein Beispiel dafür ist der Aufbau eines Erdbebenfrühwarnsystems für Istanbul[7].

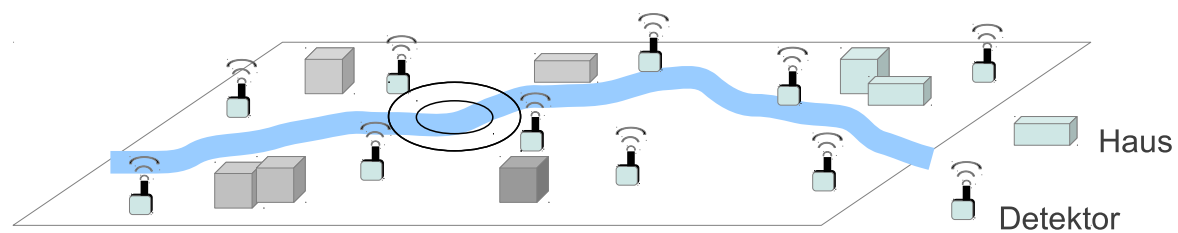


Abbildung 1.1: Erdbeben Anwendung

Die Abbildung 1.1 zeigt eine Region in dem Sensoren Daten über seismische Wellen ermitteln können. Mit Hilfe dieser Daten kann die Ausbreitung von Erdbeben prognostiziert werden und damit darauf reagiert werden. Obwohl eine Vorwarnzeit von wenigen Sekunden vorliegt, sind Maßnahmen zur Verhinderung von Folgeschäden wie z.B. Gasexplosionen oder Brände denkbar. Die Daten über das Erdbeben können dabei auch für die Organisation von Rettungsmaßnahmen zu Hilfe genommen werden. Für die Funktionsfähigkeit eines solchen Sensornetzes ist es wichtig, dass der Ausfall einzelner Netzknoten keine Probleme bereitet. Um diese Ausfallsicherheit zu erreichen, müssen die Netzknoten den Aufbau des Netzes selbst organisieren. Hierzu gehört die Eingliederung in das Netz bzw. das Verteilen der Informationen innerhalb des Netzes. Für die Vernetzung der einzelnen Knoten können verschiedene Funktechnologien eingesetzt werden, z.B. 802.11, GSM/UMTS, ZigBee.

Ein Beispiel für einen weiteren Anwendungsfall ist der Aufbau neuer Kommunikationsstrukturen in teilweise oder noch nicht genug erschlossenen Gebieten, wie z.B. kleinere Dorfgemeinschaften. Hier können drahtlose und drahtgebundene Kommunikationen miteinander kombiniert werden, um wenige Breitbandinternetverbindungen (z.B. DSL) für viele Anwender zur Verfügung zu stellen. Die Abbildung 1.2 zeigt ein teilweise erschlossenes Gebiet, die eine Hälfte der Bewohner haben bereits eine DSL Leitung

und die anderen können über den WLAN-Router den DSL Zugang ihrer Nachbarn mitbenutzen. Eine andere Lösung, wäre das einrichten eines Ad-hoc Netzwerkes bei dem einige zusätzlich Zugang zum DSL haben. Als Ad-hoc bezeichnet man das Verbinden von WLAN Clients ohne die Verwendung eines Router.

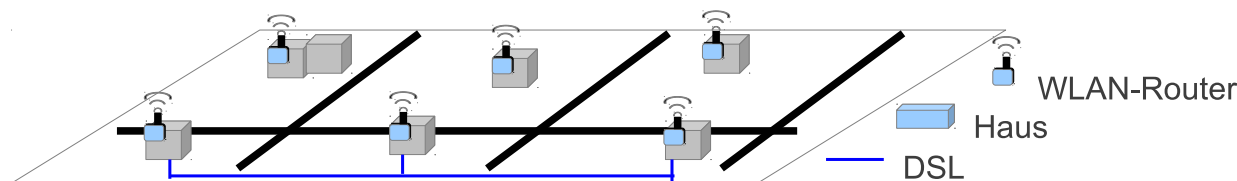


Abbildung 1.2: Wireless erweiterte Kommunikationsanwendung

Je nach Anwendung können dabei verschiedene Technologien und Protokolle verwendet werden. Hierbei müssen Probleme wie der Zugriff auf das Medium mittels Carrier Sense Multiple Access/Collision Detection (CSMA/CA[4]) oder Point Coordination Function[12] und der Vermittlung von Datenpaketen (Routing) gelöst werden, dafür wird eine größere Flexibilität in der Entwicklung von Algorithmen benötigt. Forscher versuchen neue Lösungen für die oben genannten Probleme zu finden und für diese Umsetzung brauchen sie mehr Freiheiten als die meiste Hardware ihnen bietet. Im Fokus der Forschung liegen auch Kommunikationsnetze bei denen den beteiligten Knoten mehrere Kanäle zur Verfügung stehen (multichannel) oder über verschiedene Radios (multiradio) miteinander kommunizieren. Darüber hinaus können auch mehrere Funktechnologien, wie WiFi (802.11), Zigbee (802.15.4), Mobilfunk (GSM/UMTS) oder auch Bluetooth, miteinander kombiniert werden. Durch die Verwendung unterschiedlicher Technologien bzw. Radio Frequenzen können mehrere Kommunikationskanäle gleichzeitig verwendet werden. Die Anzahl der Geräte für eine Funkfrequenz kann gesenkt und damit der Durchsatz bzw. die Zuverlässigkeit der Datenübertragung für diese Geräte erhöht werden. Auch die Sicherheit kann durch verschiedene RF Technologien (getrennte Kommunikationskanäle) erhöht werden. Hierbei können z.B. die Daten für die Authentisierung oder von Aktionen über den ersten Kommunikationskanal gesendet und über einen zweiten bestätigt werden. Diese Trennung ermöglicht das Erkennen bzw. Verhindern von Datenmanipulation während der Übertragung.

Mit drahtloser Kommunikation können Verbindungen zwischen mobilen und stationären Geräten hergestellt werden, z.B. zwischen Laptop und einem Router mit einer DSL Verbindung. In der Forschung werden auch Verbindungen zwischen mobilen bzw. stationären Geräten hergestellt, um z.B. große Sensornetze aufzubauen. Solche Sensornetze können wie in der Abbildung 1.1 dazu verwendet werden, Wetterphänomene, Produktionsstrecken oder Grenzen zu überwachen. Für den Aufbau solcher Funknetze existieren verschiedene Möglichkeiten, wie Ad-hoc Netzwerke oder Infrastruktur Netzwerke. Als Infrastruktur Netzwerk werden Funknetze bezeichnet, in denen ein oder mehrere Geräte (AP) eine besondere Funktion für andere Netzteilnehmer erfüllen. Sie sind für die Koordination der Kommunikation innerhalb des Netzwerkes zuständig. In einem Ad-hoc Netzwerk können alle Geräte, ohne Verwendung eines AP für die Koordinierung, direkt miteinander kommunizieren. Es existieren verschiedene Standards, in denen die Funkfrequenzen, die verwendete Modulation und Protokolle definiert werden. In der Forschung wird versucht bessere Protokolle für die Funktechnologie bzw. den Aufbau von Kommunikationsnetzen zu entwickeln, z.B. Routing bei der Kombination verschiedener Funktechnologien (multi technologie) oder bei Kognitiv Radio[8]. Bei der Nutzung verschiedener Technologien müssen effiziente Routing Protokolle entwickelt werden, um die Datenrate innerhalb des Funknetzes zu erhöhen. Diese Protokolle müssen auf die Eigenschaften der einzelnen Technologien, wie ihre Vor- bzw. Nachteile, eingehen. Bei Kognitiv Radio geht man noch einen Schritt weiter, hier wird die Umgebung auf mögliche Störquellen oder bereits verwendete Funkfrequenzen untersucht. Mit diesen zusätzlichen Informationen und geeigneten Algorithmen

sollen die Geräte eine Strategie für die Nutzung und Einstellung der zur Verfügung stehenden Technologie erlernen. Bei diesen Forschungen kommt SDR (Software Defined Radio) zum Einsatz, um die größtmögliche Flexibilität bei der Entwicklung von Protokollen zu erhalten.

Für die Entwicklung von Netzwerkanwendungen hat sich das ISO/OSI Modell[6] in der Praxis bewährt. Um zu verstehen auf welcher Ebene Eingriffe innerhalb dieses Modells möglich sind, soll ein Blick darauf geworfen werden. Die Anwendungs-, Darstellungs-, Sitzungs-, Transport- und Vermittlungsschicht werden durch Software realisiert und daher sind Anpassungen an die eigenen Bedürfnisse leichter möglich. Den Zugriff auf das jeweilige Medium, die sichere Übertragung sowie die Kodierung der Daten wird in der Sicherungs- und Bitübertragungsschicht abgewickelt. Diese beiden Schichten sind meistens in Hardware implementiert und lassen sich daher schwer bzw. nicht anpassen. Für die drahtlose Kommunikation gibt es einige spezialisierte Hardwarearchitekturen, die die Anpassungen der letzten beiden Schichten durch Software teilweise bzw. vollständig ermöglichen. Der Oberbegriff für diese Hardware ist SDR. Das Ziel von SDR liegt bei der Umsetzung möglichst vieler Teilfunktionen einer Sende-/Empfangsfunktionalität in Software. Da sich Änderungen in der Software schneller und kostengünstiger als mit Hardware realisieren lassen, können die Forscher mit SDR leichter neue Protokolle und Übertragungsverfahren realisieren. Der Preis für solche SDR Hardware beträgt mehr als 500 Euro je Gerät. Durch die Verwendung von Hardware mit genau der notwendigen Funktionalität bzw. Flexibilität könnten preiswertere Lösungen gefunden werden. Als Beispiel dient hier der Openbeacon USB Stick, deren Verwendung als SDR in dieser Arbeit untersucht wird.

1.1 Aufgabenstellung

Das Ziel dieser Arbeit ist es eine solche low-cost SDR Hardware in eine bestehende Plattform zu integrieren und ihr Potenzial festzustellen. Die vorliegende Openbeacon Hardware besteht aus einer RFID Antenne, einem ATMEL Mikrocontroller und ist über USB mit dem PC/Router verbunden. Zur Aufgabe gehörte die grundlegende Anbindung einer neuen Hardware an die bestehende Click-Router Software und Messungen der endgültigen Leistung der Hard- und Software. Die Lösung sollte möglichst flexibel und leicht für spätere Entwicklungen anwendbar sein. Für die meisten Anwendungen wird daher eine Abstraktionsschicht in der Click-Router Software erfolgen. Diese ermöglicht das Versenden und Empfangen von kleinen Ethernetpaketen und dadurch eine leichtere Anwendung. Für die Flexibilität soll die Steuerung des Datenverkehrs möglichst über die Anwendung in der Click-Router Software erfolgen. Aus diesem Grund soll die Firmware eine einfache Sende- bzw. Empfangsfunktionalität ermöglichen.

1.2 Aufbau der Arbeit

Zunächst erfolgt im Kapitel 2 eine Einführung in die Technologien Software Defined Radio und Radio Frequency Identification. Dann wird die verwendete Plattform (Humboldt Wireless Lab) im Kapitel 3 vorgestellt. Hier wird auf die für diese Arbeit wichtigsten, schon vorhandenen Hard- und Softwareelemente eingegangen. Darunter fällt der Openbeacon USB Stick, die für seine Arbeit notwendige Firmware und die Click Router API. Letztere soll die Evaluation des Openbeacons unterstützen. Im Anschluss folgen Erläuterungen über Details der Implementierung im Kapitel 4 und im Kapitel 5 die Evaluation des Openbeacons. Am Ende der Arbeit werden die Ergebnisse zusammengefasst und ein Ausblick gegeben.

Kapitel 2

Einführung

In diesem Kapitel wird ein Überblick über die Grundlagen gegeben, die für das Verständnis dieser Arbeit notwendig sind. Bei der Netzwerkkommunikation handelt es sich um eine umfangreiche und komplexe Technologie, die über drahtgebundene und drahtlose Verbindungen aufgebaut werden kann. Hierbei können in drahtgebundenen Netzwerken einfache Punkt-zu-Punkt Verbindungen, sternförmige Verbindungen über Hubs und komplexere Strukturen über Switches, Bridges und Router hergestellt werden. Das Internet ist das umfangreichste Beispiel für solche Strukturen. Bei drahtlosen Netzen werden ebenfalls einfache Punkt-zu-Punkt Verbindungen per Bluetooth, sternförmige Verbindungen über einen Wireless Access Point (AP), komplexere Strukturen mit Hilfe von mehreren APs, über Wireless Distribution System (WDS) oder über den Ad-hoc Modus aufgebaut. Bei der Verwendung eines AP können die jeweiligen Klienten nicht direkt miteinander kommunizieren, sondern die Kommunikation wird über den AP koordiniert. Dies gilt auch bei der Erweiterung über WDS mit mehreren APs. Hierdurch kann das Netzwerk, ohne Verkabelung der einzelnen APs, erweitert werden. Im Gegensatz dazu kann im Ad-hoc Modus jedes Endgerät mit jedem Anderen, soweit es in Reichweite ist, direkt kommunizieren. Mit mehreren Endgeräten kann dadurch ein sehr großes vermaschtes Netz aufgebaut werden. Für diesen Aufbau wird umfangreiche und komplexe Software benötigt, z.B. für das effiziente Verteilen von Information (Routing) innerhalb solcher so genannten Mesh-Netze. Um die Implementierung für Netzwerkarchitekturen zu vereinfachen, hat sich das Open Systems Interconnection Reference Model (OSI) durchgesetzt. In diesem Modell wird die Komplexität in einzelne Schichten zerlegt, wobei jede Schicht eigene Dienste für die jeweils nächst höhere Schicht erbringt. Die virtuelle Kommunikation zwischen Schichten der gleichen Kommunikationsebene erfolgt hierbei über das gleiche Protokoll. Im ISO/OSI Modell werden 7 Abstraktionsschichten unterschieden, von der obersten Anwendungsschicht bis hinunter zur Bitübertragungsschicht.

Die Daten für die jeweilige Anwendung liegen in der Verantwortung der Anwendungsschicht, also der Endanwendung selbst. Hier sind viele Anwendungen vorstellbar, die folgenden sollen nur ein kleiner Auszug sein.

- Dateien verteilen (ftp)
- Webseiten abrufen (http)
- telefonieren (skype)
- Verteilte Anwendungen (z.B. Cloud)
- ...

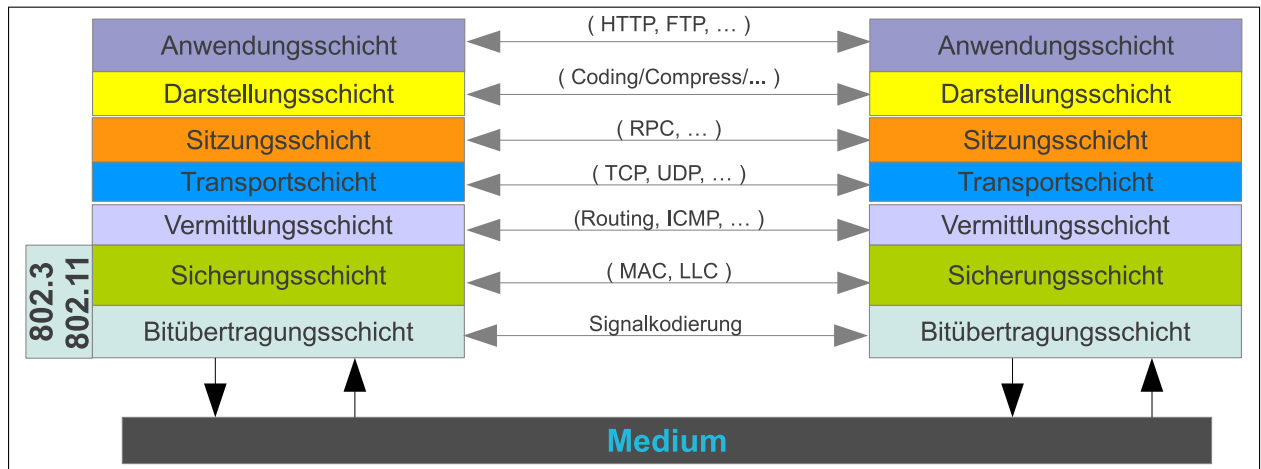


Abbildung 2.1: ISO/OSI Diagramm

Um diese Dienste realisieren zu können, müssen die Daten geeignet kodiert (z.B. den Zeichensatz festlegen ANSI/UTF-8) und eventuell komprimiert werden, aber auch eine etwaige Verschlüsselung der Daten fällt in die Darstellungsschicht. Die Aufgabe der Sitzungsschicht ist die Verbindung aufrecht zu halten und sie gegebenenfalls bei einem Abbruch der Kommunikationsverbindung wieder aufzunehmen. Die Transportschicht ist für den Verbindungsaufbau, dem Versand/Empfang der Daten und Verbindungsabbau zuständig. Die wichtigsten Vertreter sind hier das verbindungsorientierte Transmission Control Protocol (TCP) und das verbindungslose User Datagram Protocol (UDP) Protokoll. In der Vermittlungsschicht wird durch die Netzwerkadressen und zusätzliche Routing Informationen der Weg zum Zielrechner ermittelt. Hier muss die Verbindung zwischen den Endpunkten über mehrere Rechner hergestellt und aufrecht erhalten werden. Jeder Rechner der Teil dieser Verbindungsstrecke ist, muss die Pakete richtig weiterleiten können.

Jede Verbindungsstrecke besteht nun aus einzelnen direkt miteinander verbundene Rechnerpaare. Diese Teilverbindungen werden nun durch die Sicherungsschicht aufgebaut und aufrecht erhalten. Hierzu ist es nötig den Zugriff auf das Medium zu verwalten (MAC-Teilschicht) und die Übertragung fehlerhafter Pakete zu erkennen (LLC-Teilschicht), zum Beispiel durch eine zyklische Redundanzprüfung (CRC). Für die MAC Schicht gibt es je nach Medium verschiedene Verfahren bzw. vorzubeugen. Bei der drahtgebundenen Kommunikation ist es meistens Carrier Sense Multiple Access/Collision Detection (CSMA/CD), um Kollisionen von Paketen zu erkennen. Dieses Verfahren kann bei drahtloser Kommunikation nicht eingesetzt werden, da während einer Übertragung das Medium nicht überwacht werden kann. Hier werden Verfahren wie Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) zur Kollisionsvermeidung eingesetzt.

In der untersten Bitübertragungsschicht müssen nun die einzelnen Bits für das Medium entsprechend kodiert werden. Die Standards 802.3 (drahtgebundene) und 802.11 (drahtlose) legen die Verfahren und die physikalische Hardware fest.

Dies wird zumeist im Treiber, in der Firmware und in der Hardware implementiert. Diese Schichten ist es bei den meisten Geräten, die diese Standards verwenden, nicht möglich bzw. schwierig, tiefer liegende Aspekte zu verändern (z.B. eine Alternative/Verbesserung für CSMA/CA). Lassen sich die Treiber später noch leicht austauschen, ein Beispiel hierfür ist der Opensource MadWifi Treiber. Ist es bei der Firmware schon viel schwieriger Änderungen vorzunehmen. Bei der Funktionalität der Hardware lassen sich dann keine nachträglichen Anpassungen mehr vornehmen. Soweit dies durch den Hersteller ermöglicht wurde, kann die Hardware dann nur noch durch Konfigurationen von Parametern beeinflusst werden. Für

die drahtlose Kommunikation gibt es einige wenige Hardwarearchitekturen, die sich darauf spezialisiert haben, den Zugriff noch weiter runter bis zur Bitübertragungsschicht zu erlauben.

2.1 Software Defined Radio

SDR (Software Defined Radio) ist der Oberbegriff für eine Hardware, mit deren Hilfe möglichst jede Komponente einer Sende-/Empfangsfunktionalität in Software umgesetzt werden kann. Um die größtmögliche Flexibilität bei der Entwicklung von Funktechnologien zu erhalten, wurde daher die in Hardware umgesetzte Funktionalität so klein wie möglich gehalten und die übrige in Software implementiert. Hierbei sollen die empfangenden bzw. gesendeten Signale allein durch Software generiert und verarbeitet werden. Lediglich für das Versenden/Empfangen der Signale wird Hardware benötigt. Neben der Software und dem für die Verarbeitung benötigten Chip (z.B. DSP oder FPGA), werden noch ein Digital/Analog-Konverter (DAC) bzw. Analog/Digital-Konverter (ADC) und die Antenne selbst benötigt. Die Entwickler haben dadurch Einfluss auf die Modulation des Signals, können Medienzugriffsverfahren entwickeln und können einen flexibleren Frequenzbereich verwenden. Dies ermöglicht die Weiterentwicklung alter bestehender Protokolle oder die Entwicklung neuer Protokolle. Im Handel werden einige SDR Geräte angeboten, die sich in ihrem Aufbau und Preis zum Teil erheblich unterscheiden.

Das von der Firma Ettus angebotene USRP[17] kann durch unterschiedliche RF-Module für verschiedene Frequenzen verwendet werden und besteht im wesentlichen aus DAC/ADC und einer FPGA. USRP wurde zusammen mit der Open Source Softwarearchitektur GNU Radio entwickelt, und ermöglicht das schnelle Umsetzen von verschiedenen Protokollen wie z.B. 802.11. Microsoft geht mit SORA[14] einen ähnlichen Weg, versucht aber den Programmierern eine einfachere Entwicklungsumgebung anzubieten. Über ein SDK soll sich der Entwickler nur auf die eigentliche Entwicklung neuer Funktionalitäten konzentrieren. Als Beispiel wurde der 802.11 Standard implementiert. Calradio[1] ist ein weiterer Vertreter, er baut einen kompletten 802.11b Stack in Software auf und erlaubt den Entwickler dort alle Schichten oberhalb der Bitübertragungsschicht für die eigenen Zwecke anzupassen. Alle drei Vertreter haben einen höheren Preis (ab 500 Euro), bieten dafür auch eine Vielzahl an Möglichkeiten für die Modifikation/Entwicklung eigener Protokolle.

Preiswerter und einfacher Ersatz sind Tmote Sky[15], Pololu Wixel[13] und Openbeacon[9], die jeweils weniger als 100 Euro pro Stück kosten. Diese verwenden eine Hardware, die nur Anpassungen der MAC Schicht erlaubt, die Bitübertragungsschicht wird durch Hardware implementiert. Die Anbindung an einen PC erfolgt bei ihnen ebenfalls über eine USB Verbindung.

Tmote Sky implementiert den IEEE 802.15.4[5] Stack (WPAN) und hat Sensoren für Luftfeuchtigkeit, Licht und Temperatur. IEEE 802.15.4 beinhaltet dabei die MAC- und Bitübertragungsschicht. Man unterscheidet bei den Endgeräten zwischen Reduced Function Devices (RFD) und Full Function Devices (FFD). RFD haben einen geringeren Funktionsumfang als FFD und können daher nur mit diesen kommunizieren, während FFD auch untereinander Verbindung aufnehmen können. Dabei werden komplexere Strukturen über einen Koordinator aufgebaut, der die Identifizierung des Netzes gegenüber anderen IEEE 802.15.4 Netzen festlegt. Es können sowohl die ISM-Bänder 868/915 MHz als auch das 2.4 GHz Band verwendet werden, wobei die Datenrate bei 20 bzw. 250 Kbit/s liegt. Als Medienzugriffsverfahren kommt CSMA/CA zum Einsatz bzw. kann der Koordinator die Kommunikation der beteiligten Endgeräte steuern. Für die Funkübertragung kommt des weiteren das Direct Sequence Spread Spectrum (DSSS) Verfahren zum Einsatz, um die Übertragung unempfindlicher gegen Störungen zu machen. Pololu Wixel ist ein frei programmierbarer 2.4 GHz Funkempfänger und wird z.B. bei der Fernsteuerung von kleinen Robotern verwendet. Er verwendet eine Bitrate von 350 Kbit/s und die Reichweite beträgt ca. 15 Meter. Der Openbeacon USB-Stick ist ein preisgünstiges Radio zur Kommunikation im 2.4 GHz Bereich und

wird hauptsächlich für Anwendungen im RFID Umfeld eingesetzt. Es wird hier eine Bitrate von 1 bzw. 2 MBit/s verwendet. Anwendungsbeispiele sind die Standortbestimmung, wie im Jewish Museum in Berlin, oder die Echtzeit Positionserfassung für die Besucher von Museen/Messen um nur zwei zu nennen.

2.2 Radio Frequency Identification

Unter Radio Frequency Identification (RFID) wird im Allgemeinen eine Technik zur berührungsfreien Identifizierung von Objekten verstanden. Ende des zweiten Weltkrieges wurden erste Anwendungen zur Freund-Feind Erkennung eingesetzt. Später fand RFID den Weg in die Produktion und den Vertrieb. Hier sollten die Produkte während der gesamten Herstellung Schritt für Schritt verfolgt werden können. Später war diese Technologie auch bei der Verwaltung großer Warenbestände hilfreich. Im Gegensatz zum Barcode, bei dem die Daten codiert für alle sichtbar sind, müssen bei RFID die Daten abgerufen werden und stehen daher nur ausgewählten Nutzer zur Verfügung. Ein weiterer Vorteil gegenüber dem Barcode ist die Lageunabhängigkeit des Objektes, es muss nicht mehr dicht und möglichst im richtigen Winkel an einem Scanner vorbei geleitet werden. RFID hat eine Reichweite von mehreren Metern. Im RFID Umfeld wird in aktive und passive Transponder unterschieden. Aktive Transponder verfügen über eine eigene Energiequelle und können daher einen größeren Funktionsumfang anbieten (z.B. unaufgefordert Informationen zu senden). Passive Transponder nutzen die Energie der Funkwellen, die von einem Lesegerät ausgesendet werden. Sie müssen daher innerhalb kurzer Zeit antworten. Bei RFID ist sowohl der Transponder als auch das Lesegerät aktiv am Informationsaustausch beteiligt. Für den Austausch von Identifikationsinformationen sind wenige Bytes ausreichend, was eine einfachere Hardware im Vergleich zu anderen Geräten, wie z.B. Routern, benötigt.

Bei der verwendeten Openbeacon Hardware handelt es sich um einen so genannten aktiven Transponder. Er verfügt über einen Mikrocontroller und einer Energieversorgung über USB. Der Prozessor hat eine Leistung von 48 MHz und verfügt über jeweils 128 KByte ROM und RAM. Es wird eine Bitrate von 1 bzw. 2 MBits und 2.4 GHz Funkfrequenz verwendet. Im Hinblick auf spätere Anwendungen für die Openbeacons sollte kein einfaches Identifizierungssystem nachgebildet, sondern komplexere dynamisch erzeugte Nachrichten verteilt werden.

Kapitel 3

Plattform

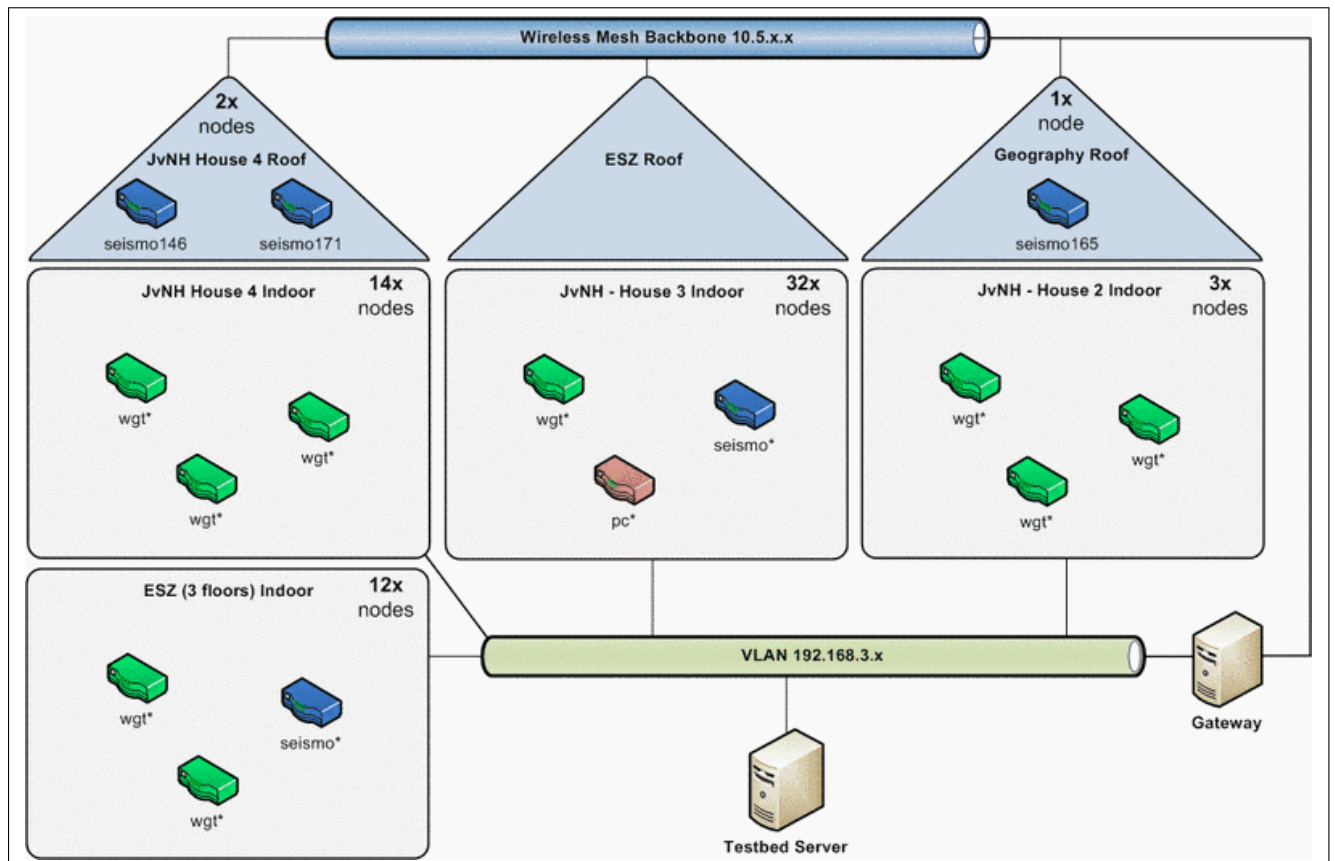


Abbildung 3.1: aktuelle Netzwerk Architektur für das HWL

Um die Eignung des Openbeacon USB Sticks als SDR besser beurteilen zu können, soll dieser in die bestehende Plattform des Humboldt Wireless Lab (HWL) integriert werden. Das HWL ist ein Forschungsnetzwerk, in dem neue Protokolle und Anwendungen in den verschiedenen Ebenen des ISO/OSI Modells

entwickelt und getestet werden können. Das HWL besteht aus einer Reihe von Hard- und Software Komponenten (z.B. auf MIPSSEL basierende Router, Linux als Betriebssystem, Click Router Software, ...). Im HWL liegt das Augenmerk auf dem schnellen und unkomplizierten Einrichten dieser Knoten. Die Knoten innerhalb eines Gebäudes werden über Ethernet mit einem Server verbunden. Alle außerhalb des Gebäudes aufgestellten Knoten werden über eine zusätzliche Wifi Karte/Radio über ein drahtloses Netz mit diesem Server und einem zweiten Gateway verbunden. Alle Knoten beziehen ihre Software zum Arbeiten von und sichern evtl. anfallende Daten auf diesem Server. Zusätzlich sind die Knoten mit verschiedenen Sensoren ausgestattet und diese verteilten Daten können dann zusammengeführt und ausgewertet werden. Die Umsetzung wurde mit Hilfe von Linux(OpenWRT[11]), der Click Modular Router Software[2] und ClickWatch[3] entwickelt. Die Abbildung 3.1 zeigt einen schematischen Aufbau des HWL. Mit Hilfe des HWL können Forscher an Algorithmen und Protokollen für selbstorganisierende Netze arbeiten. Diese Forschung soll dabei helfen, zukünftige Sensornetze bzw. vermaschte Netze schnell, einfach und kostengünstig einrichten zu können.

Im HWL sind drei CalRADIOS und drei Sora Geräte integriert, jedoch sind diese kompliziert und sehr teuer. Für den Aufbau größerer Netze benötigt man eine höhere Anzahl von diesen Geräten, was mit hohen Anschaffungskosten verbunden ist. Aus diesem Grund wäre es von Vorteil preisgünstigere Geräte, wie den Openbeacon USB Stick, für die Forschung nutzbar zu machen.

3.1 Openbeacon

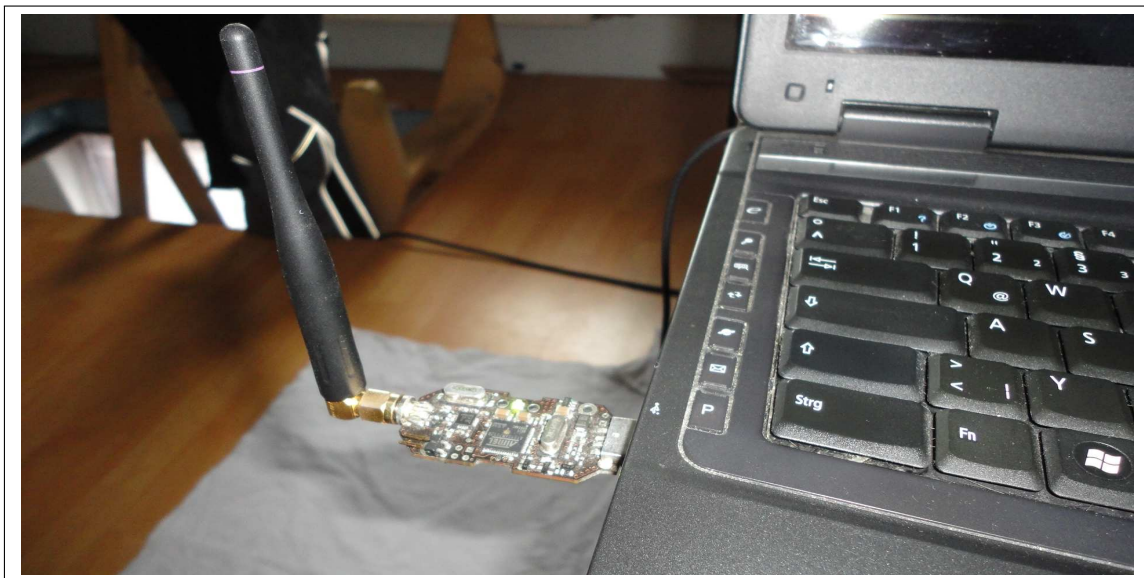


Abbildung 3.2: Anordnungsbeispiel für Openbeacon

Openbeacon ist eine offene Plattform, um aktive RFID Anwendungen zu geringen Kosten entwickeln zu können. Zum Einsatz kommen Open Source Soft- und Hardware. Für die Plattform wurden verschiedene Geräte entwickelt, darunter Karten für Personen zum mitführen, den Openbeacon USB-Stick und andere Geräte. Der im Openbeacon USB Stick eingebaute Atmel Chip erlaubt eigene Anwendungen direkt auf dem Gerät auszuführen, was ihn unabhängig von einem Rechner (PC oder Router) macht. Er benötigt

nur eine geeignete Stromversorgung, um seine Funktionalität ausführen zu können. Verbaut wurde ein nRF24L01 Chip für das Versenden der Daten über die Antenne. Dieser bietet zusätzlich und optional die Funktionalität für MAC und LLC an (Enhanced ShockBurst). Die Daten werden im 2.4GHz ISM Band versendet, dafür lassen sich 126 verschiedene Kanäle nutzen. Darüber hinaus können 4 verschiedene Sendeleistungen und eine Bitrate von 1 oder 2 Mbit/s verwendet werden. Eine Einschränkung der Nutzdaten auf max. 32 Bytes ist durch die interne Registergröße gegeben, größere Datenmengen müssen daher fragmentiert werden. Für die Adressierung der einzelnen Openbeacons werden bis zu 6 Adressen vergeben, die jeweils eine Länge von 3-5 Bytes aufweisen können. Hierbei können 2 Adressen unabhängig von einander gewählt werden. Die 4 anderen Adressen teilen sich den Adressanfang von der 2. Adresse. Dabei werden Pakete nur dann an die Anwendung weitergeleitet, wenn sie als Empfänger eine von den 6 Adressen haben. Für eine einfache Programmierung wird in dieser Lösung eine Adresse (Netzwerk Adresse in dieser Arbeit als NID bezeichnet) verwendet. Alle Knoten bilden daher ein Netzwerk, in dem jeder alle empfangenen Pakete an die Anwendung weiterreicht. Um die Verbindung zwischen zwei Knoten identifizieren zu können, werden die Quell- bzw. Zielknotenadresse innerhalb des Datenpaketes abgelegt. Hierfür werden jeweils 2 Bytes reserviert, somit können innerhalb eines Netzwerkes mehr als 65000 Openbeacons verwendet werden. Für den Transport der Daten über Funk ist der nRF24L01-Chip verantwortlich, bei ihm können Parameter wie der Kanal, die Bitrate oder Sendeenergie für die Funkübertragung der Daten eingestellt werden.

Tabelle 3.1: einstellbare Parameter für die Funkübertragung

Bitrate	1 und 2 Mbit/s
Sendeleistung	-18dBm, -12dBm, -6dBm, 0dBm
Kanal	bis zu 126 Kanäle von 2.400GHz bis 2.525GHz
NID	5 Byte die Identifikation des Netzes, für die Netzunterscheidung

Die Sende- und Empfangsfunktionalität wird in getrennten Modi betrieben, diese müssen vorher eingestellt werden. Der nRF24L01 muss in den Empfangsmodus versetzt werden, um Daten von anderen Geräten empfangen zu können. In diesem Zustand kann er selbst keine Daten versenden, bis er in den Sendemodus versetzt wurde.

Openbeacon Firmware

Auf dem Openbeacon kann eine Software geladen werden, diese muss verschiedene Funktionalitäten bewältigen. Darunter fallen der Transport der Daten über die USB bzw. Funk Schnittstelle. Der Atmel verfügt über ein 128 KByte ROM für das Programm und ein 128 KByte RAM für die Daten. Der ROM enthält eine Basisfirmware, diese kann durch setzen eines Jumpers die laufende Firmware ersetzen. Dadurch ist es möglich jederzeit eigene Firmware auf den Openbeacon zu laden, ohne die Gefahr den Openbeacon dauerhaft unbenutzbar zu machen. Die Openbeacon Programmierschnittstelle (API) unterstützt die Aufgabenverteilung, ähnlich wie beim x86 Prozessor, mit Hilfe der Hardware. Da die Firmware als ein Programm betrachtet werden kann, werden die Teilaufgaben als Threads bezeichnet. Für den Transport per USB wurde ein eigener Thread verwendet. Die Kommunikation zwischen dem USB Thread und der Anwendung, welche in einem eigenen Thread läuft, erfolgt über Queues. Eine Queue enthält die Daten die über USB empfangen wurden und eine andere die Daten, welche über USB versendet werden sollen. Die Firmware beinhaltet auch diverse Hilfsfunktionen, darunter die Implementierung der Queues und des Schedulers, der die Abarbeitungsreihenfolge der einzelnen Thread steuert. Die Arbeit der einzelnen Thread wird über einen Timer gesteuert, der eine Hardware Unterbrechung (IRQ) auslöst und so den Wechsel über den Scheduler ermöglicht. Der Scheduler kann auch einzelne Threads bevorzugen, ihnen also mehr Rechenzeit zur Verfügung stellen.

Hiermit könnte zum Beispiel dem Empfang der Pakete, von USB bzw. über Funk, mehr Rechenzeit zur Verfügung gestellt werden, damit diese nicht verloren gehen. Sollten alle Threads bereits Prozessorzeit beansprucht haben, wird die Prozessorzeit einer zusätzlichen Funktion zugewiesen. Diese Funktion wird vollständig ausgeführt und sollte daher nur kleine Aufgaben erledigen, hier könnten zum Beispiel Daten für Statistiken gesammelt werden.

3.2 Entwicklungsumgebung

Für die Integration der Openbeacon USB Sticks in das HWL, standen mehrere Netgear Router zur Verfügung. Diese verwendeten als Betriebssystem Linux und wurden durch die Click Router Software erweitert, auf die später genauer eingegangen werden soll. Es standen mehrere Openbeacon USB Sticks, eine Toolchain für den ATMEL Mikrocontroller und einige Anwendungsbeispiele zur Verfügung. Die Anwendungsbeispiele wurden der Openbeacon-Quellen[10] entnommen und erlaubten einen einfachen Zugriff auf die Hardware über USB. Die Openbeacon-Quellen enthielten weiterhin auch mehrere Beispielanwendungen, mit denen man Empfangene Pakete anzeigen lassen oder eine Empfangsstatistik erstellen konnte.

3.3 Click Router API

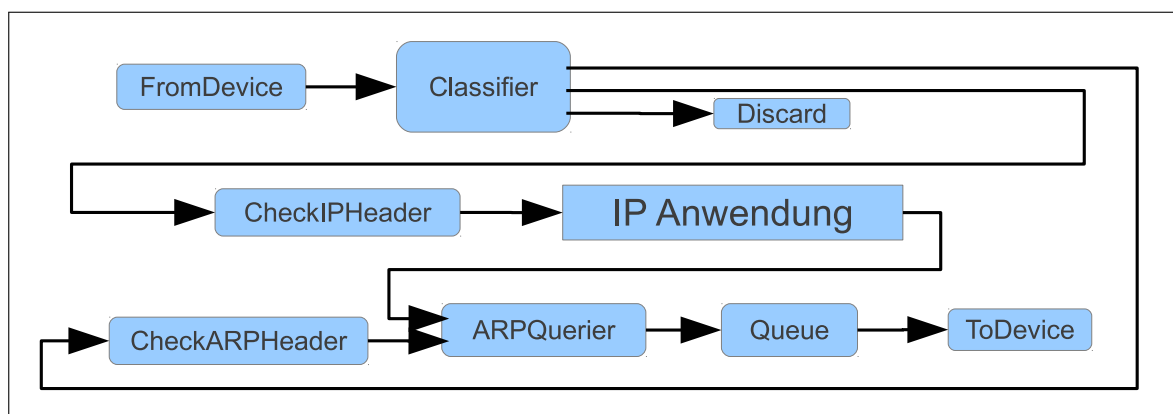


Abbildung 3.3: eine kleine Click Router Anwendung (Auszug)

Für die Entwicklung von neuen Protokollen und Anwendungen wird im HWL die Click Router Software (Click) verwendet. Mit Hilfe von vorgefertigten oder eigenen Komponenten (sogenannte Elemente) lassen sich hier komplexe Anwendungen konfigurieren. Durch diese Elemente lassen sich sehr einfach neue Funktionalitäten entwickeln und der modulare Aufbau vereinfacht die Wiederverwendbarkeit. Für die spätere Verwendung des Openbeacon USB Sticks wurde daher eine Anbindung an Click vorgenommen. Dies ermöglicht eine schnelle Entwicklung neuer bzw. die Verwendung schon vorhandener Protokolle und Anwendungen für den Openbeacon USB Stick.

Die Kommunikation zwischen den jeweiligen Elementen erfolgt hauptsächlich über Netzwerkwerkpakete. In Abbildung 3.3 ist eine Beispielkonfiguration zu sehen. Die empfangenen Pakete werden je nach Typ unterschiedlich weiterverarbeitet. In diesem Beispiel werden nur Address Resolution Protocol (ARP) und

IP Pakete verwendet und alle anderen Pakete werden verworfen. Die IP Pakete gelangen zur Anwendung und die ARP Pakete werden von einem eigenen Element, dem ARPQuerier verarbeitet. Das Element muss auf Anfragen antworten und selbst auch Anfragen aussenden können. Auch der Anwendung wurde ermöglicht Pakete zu versenden, als Anwendungsbeispiele könnten Dynamic Host Configuration Protocol, Hypertext Transfer Protocol oder ähnliche, auf IP aufbauende Anwendungen, verwendet werden. Jedes Element kann dabei über mehrere Eingänge verfügen, über die andere Elemente ihm Pakete zukommen lassen können. Dabei können die eingehenden Pakete gelöscht, manipuliert oder unverändert an eine oder mehrere Ausgänge weitergeleitet werden. Auch kann das Element neue Pakete erzeugen und somit zum Beispiel auch ohne eingehende Pakete Anfragen formulieren. Durch diese Paketverarbeitung können Filter oder auch Antworten für die Pakete erstellt werden (z.B Firewall oder DNS/DHCP Funktionalität). Auf den verschiedenen Ebenen des ISO/OSI Modells gibt es bereits einige komplexere Elemente: In der Vermittlungsschicht wird das Routing durch Elemente wie Dynamic Source Routing (DSR) und Dynamic Address Routing (DART) implementiert. In der Anwendungsebene sind unter anderem verteilte Hash-Tabelle (DHT) und Dynamic Host Configuration Protocol (DHCP) vorhanden.

Click kann als Modul im Kernel (Kernel Modus) oder als User Prozess laufen. Die Netzwerk Pakete können dabei auf verschiedenen Wegen mit der Außenwelt ausgetauscht werden. Dazu stehen verschiedene Elemente zur Verfügung, wie etwa das FromDevice, FromSocket, FromHost, usw. In dieser Umsetzung wird Click als User Prozess laufen und eine Anbindung an andere Software erfolgt über Sockets. Mit Hilfe von Click konnten, für die Bewertung des Openbeacons, verschiedene Messungen durchgeführt werden. Zum Beispiel werden sie auf einem Laptop mit 2 Openbeacon USB Sticks, in einem Raum mit 3 Routern und in einem größeren Rahmen über 2 Gebäude verteilt durchgeführt.

Kapitel 4

Implementierung

Die Verwendung des Openbeacon USB Sticks als SDR erfordert Anpassungen der Firmware für den Transport der Daten über die drahtlose Schnittstelle. Verschiedene Parameter wie Kanal, Bitrate und Sendeleistung können über ein Protokoll zwischen Host und Openbeacon eingestellt werden. Zu diesem Zweck wird ein Programm (Openbeacon Daemon) auf dem Host implementiert, dass diese Eingaben über ein Kommandozeileninterface ermöglicht. Darüber hinaus dient dieses Programm zur Messung der Leistungsfähigkeit. Zum Beispiel soll der Openbeacon USB Stick automatisch Pakete mit einer bestimmten Datenrate erzeugen und versenden bzw. empfangen, um so die Leistungsfähigkeit der drahtlosen Schnittstelle bewerten zu können. Aber auch die USB Schnittstelle soll separat untersucht werden können. Hierfür werden zusätzliche Funktionen benötigt. Für eine leichte Verwendung der Hardware soll es anderen Anwendungen ermöglicht werden, den Openbeacon Daemon (ObD) für den Empfang bzw. den Versand von Daten zu benutzen. Beliebige Anwendungen wie z.B. Click können über Sockets eine Verbindung mit dem ObD herstellen. Da der Openbeacon USB Stick als seriellles Gerät unter Linux eingerichtet ist, wird daher eine Brücke zwischen serieller und Socket basierter Schnittstelle benötigt. Um den Openbeacon USB Stick innerhalb von Click verwenden zu können, werden Click Elemente benötigt, die die Pakete für den Openbeacon aufbereiten.

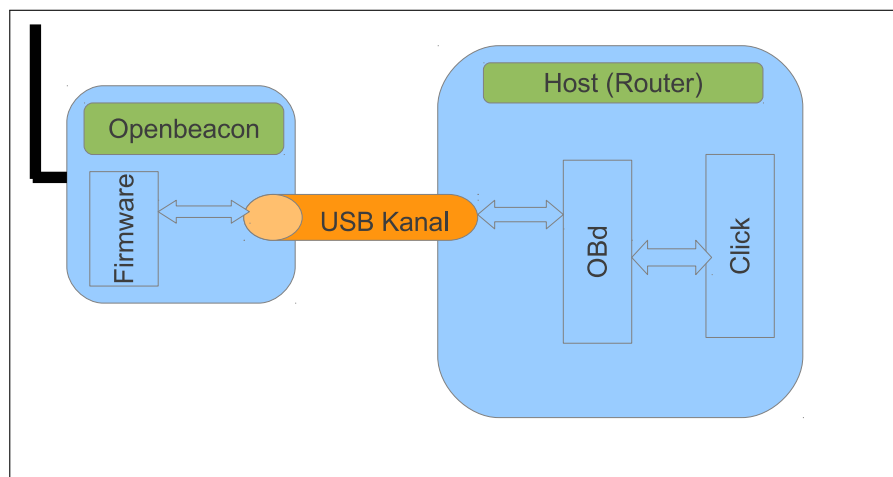


Abbildung 4.1: schematische Darstellung der Anwendung

Bei den zu transportierenden Daten handelt es sich um Ethernet Pakete (802.3), hierauf bauen viele Protokolle in der drahtlosen bzw. drahtgebundenen Kommunikation auf. Auf diese Weise könnten bestehende Protokolle/Anwendungen auch mit dieser Hardware weiter verwendet werden. Allerdings können Ethernet Pakete mehr als 1500 Byte groß sein und Openbeacon begrenzt seinerseits die Größe auf 32 Byte pro Paket. Kleine Ethernet Pakete können über diese Hardware versendet werden, größere Pakete müssten fragmentiert werden. Für die spätere Nutzung dieser Schnittstelle sollte es möglich sein die grundlegenden Parameter der Hardware auch aus dem Click heraus zu konfigurieren.

4.1 Openbeacon-Firmware

Eine Aufgabe der Firmware ist die Übertragung der Datenpakete von der USB Schnittstelle zum RF Modul und Rückrichtung. Für das Einstellen der Parameter wird ein Protokoll verwendet. Abbildung 4.2 stellt die Transformation von Ethernet Paket zu Click2OBD Paket dar. Dort stehen die einzelnen Parameter für die Funkverbindung und können somit vor dem Versand der Daten eingestellt werden. Bei empfangenen Paketen werden der gerade verwendete Kanal und die Bitrate in dieser Struktur vermerkt und über eine Queue einem Thread übergeben, der das Paket über USB versendet. Dieser Thread übernimmt die Kommunikation zwischen Openbeacon und Host, und muss dabei mit Hilfe eines Protokolls(OBD2HW) verschiedene Probleme lösen.

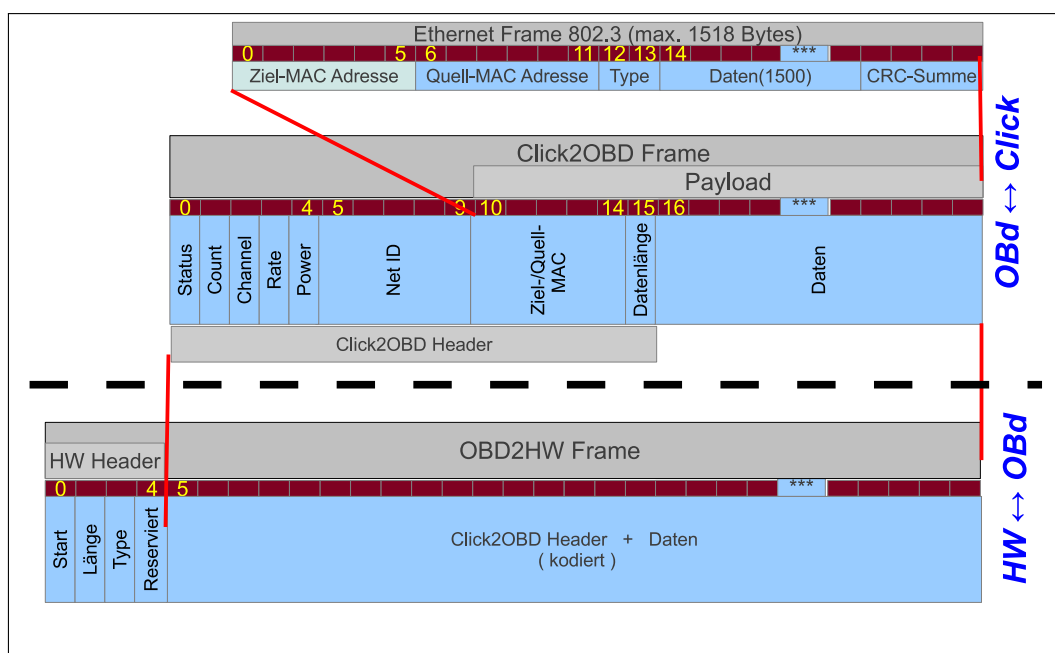


Abbildung 4.2: Protokoll Frames für das Versenden über USB

Beim Versenden der Daten über USB wurden einige spezielle Bytes zwischen 0x00 und 0x1F durch die serielle Schnittstelle ersetzt, in diesem Bereich des ASCII Codes liegen Steuerungszeichen wie Wagenrücklauf (CR=0x0D), Zeilenvorschub (LF=0x0A) und viele andere. Um die Änderung dieser Bytes zu verhindern wurden die betroffenen Bytes durch den ObD bzw. Openbeacon Firmware aus dem Datenstrom gefiltert und transformiert. Für die Transformation werden die Bytes 0x00-0x1F zu der Bytefolge 0x0120-0x013F durch addieren von 0x20. Durch das vorangestellte Byte 0x01 wird die Transformation für die spätere

Bei der Implementierung der Funktionalität wurde deutlich, dass die Auslastung der CPU des Openbeacon USB Sticks Einfluss auf die Übertragungsrate hat. Die Tabelle 5.1 zeigt die bei der Implementierung entstandenen Versionen und deren gemessenen Durchsatz der USB Schnittstelle. Von Version zu Version wurde die Firmware durch neue Funktionen erweitert. Version 1 liest die Daten nur von der USB Schnittstelle und ignoriert sie dann, hier hat die CPU keine weiteren Aufgaben zu bewältigen. In der nächsten Version wurden Statusinformationen über den aktuellen Zustand der Openbeacon HW zum Host gesendet, dies sollte die weitere Implementierung erleichtern bzw. spätere Messungen ermöglichen. In dieser Version bewältigte die CPU nicht nur den Empfang der USB Daten, sondern auch ihren Versand. In der letzten Version wurde der Empfang der USB Pakete vervollständigt und die Funktionalität für die drahtlose Schnittstelle hinzugefügt, hier verteilte sich die Rechenzeit auf die drei Threads und beanspruchte die CPU deutlich mehr als in der vorhergehenden Version. Mit dem Anstieg des Rechenaufwandes von Version zu Version verringerte sich der Durchsatz der USB Schnittstelle deutlich. Die letzte, endgültige Version enthält alle nötigen Funktionen, wie das Empfangen und Versenden von Datenpaketen über USB- bzw. drahtlose Schnittstelle und Statusinformationen, die über USB verschickt werden. Aus diesem Grund wurde nur diese Version im weiteren Verlauf dieser Arbeit verwendet.

Für eine flexible Lösung wurde auf den Einsatz der Enhanced ShockBurst Funktion verzichtet, die im nRF24L01 Chip optional aktivierbar ist. Sie übernimmt automatisch den Transport der Datenpakete über Funk und kann kleinere Umschaltzeiten zwischen Empfangs- und Sendemodus nutzen. Dem Entwickler wird die Programmierung dadurch erleichtert, da er die Pakete nur dem Sendepuffer übergeben bzw. vom Empfangspuffer entgegennehmen muss. Die Pakete werden nach dem Empfang mit Hilfe einer Checksumme auf Integrität überprüft und können optional vom nRF24L01 bestätigt werden. Wenn die Bestätigung aktiviert wurde, wird der Versand der Datenpakete automatisch bis zu einem eingestellten Wert wiederholt. Diese Lösung könnte die Funktionalität und damit die Auslastung des Atmel-Prozessors verringern und den Datendurchsatz erhöhen.

4.2 Openbeacon Daemon

Für eine flexible Verwendung der Openbeacon HW wurde der Openbeacon Daemon (ObD) entwickelt. Dieser ermöglicht es beliebigen Programmen über eine Socketverbindung Pakete an den Openbeacon USB Stick zu senden, sie müssen dafür das Click2OBD Protokoll implementieren. In diesem Protokoll werden die zu versendenden Daten mit einem Click2OBD Header versehen, wie es in Tabelle 4.2 dargestellt ist. Der ObD läuft wahlweise im Vordergrund oder im Hintergrund als Daemon. Im Vordergrund oder beim Programmstart können diverse Parameter eingestellt werden. Für den Betrieb der Openbeacon HW dienen die Funkparameter aus der Tabelle 3.1. Der ObD kann dabei mit mehreren Geräten verbunden sein und alle Einstellungen beim Programmstart gelten dabei für alle Geräte. Durch ein Konfigurationstool können die Einstellungen für den Daemon vorgenommen werden, ohne ihn dabei beenden zu müssen. Ein Teil der späteren Messungen wurden durch Parameter wie Datenrate, Paketgröße (USB Schnittstelle) bzw. Übertragungsparameter (drahtlosen Schnittstelle) ohne ein zusätzliches Programm durchgeführt.

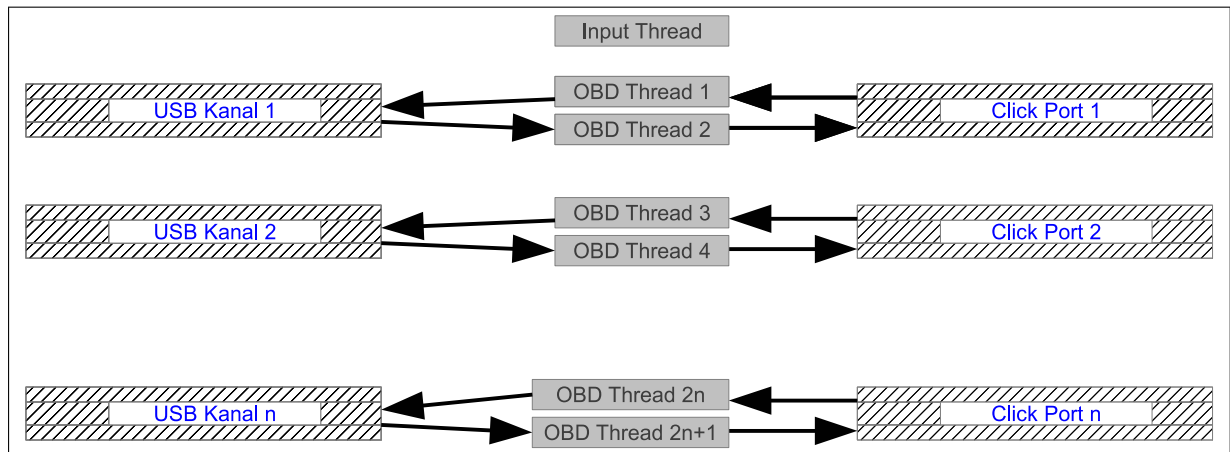


Abbildung 4.4: Softwarearchitektur des ObD

4.3 Click

Da der ObD eine einfache Socket Verbindung für den Transport von Datenpaketen anbietet, sind für Click nur wenige Elemente notwendig, um diese nutzen zu können. Dies ist ein großer Vorteil, da hiermit eine leichtere Anbindung an Click erfolgen kann als über Click Elemente. Für das Click-Framework wurden mehrere Elemente implementiert, welche Ethernet Pakete (mit Zusatzinformationen) entgegen nehmen und daraus Pakete für den ObD erstellen. Zu den Zusatzinformationen gehören Angaben über Kanal, Sendeleistung und Rate. Die Abbildung 4.2 zeigt, wie das Ethernet Paket in ein Click2OBD Paket verpackt wird und wie die entsprechenden Parameter mit zur Hardware gesendet werden.

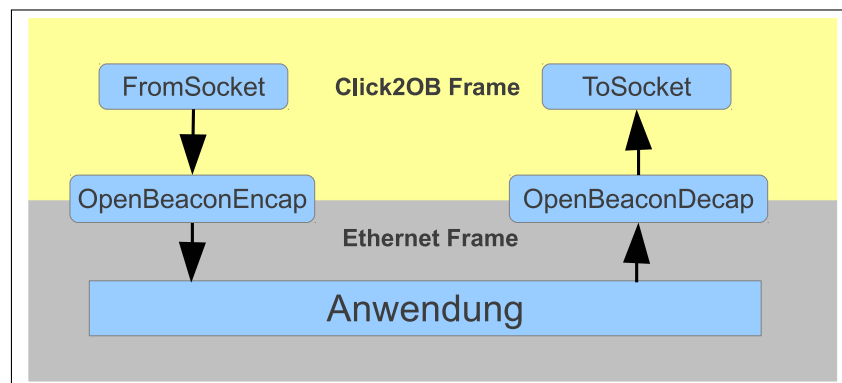


Abbildung 4.5: Anwendungsbeispiel

In der Abbildung 4.5 wird die Nutzung der Elemente dargestellt. Click2OBD Pakete können durch das Element 'FromSocket' vom ObD empfangen und über 'ToSocket' zu diesem gesendet werden. Die Transformation von einem Click2OBD Paket zu einem Ethernet Paket erfolgt dann im Element 'OpenBeaconDecap' und wieder zurück durch das Element 'OpenBeaconEncap'. Die Pakete, welche über die drahtlose Schnittstelle vom Openbeacon USB Stick versendet werden können, dürfen max. 32 Byte lang sein. Um Platz zu sparen, werden die Quell- und Ziel-Mac-Adresse zu einer Adressinformation vereint. In Testumge-

bungen können nur wenige hundert Geräte eingesetzt werden. Aus diesem Grund reichen 2 Byte für die Adressierung innerhalb eines Netzes aus, der Rest wird im Ethernet mit 0 aufgefüllt. Nun werden die beiden niederwertigsten Bytes der Quell-MAC-Adresse (QMA) und Ziel-MAC-Adresse (ZMA) zu einer 5 Byte langen Adresse kombiniert, die ersten beiden Bytes bestehen aus der QMA und die letzten beiden aus der ZMA, das mittlere Byte ist ein 0-Byte. Dieses 0-Byte dient bei der Fehlersuche der Lesbarkeit des Datenpaketes und könnte auch weggelassen werden. Das ermöglicht die direkte Adressierung von mehr als 65000 Openbeacons, was für die meisten Forschungszwecke durchaus ausreichend ist. Für spätere Anwendungen ließen sich die Openbeacons gruppieren und mit unterschiedlichen NIDs versehen. Dies ermöglicht die indirekte Nutzung von mehreren Milliarden Openbeacons ohne Änderungen an der Firmware. Das Element 'OpenBeaconEncap' kapselt Ethernet Pakete in ein Click2OBD Frame, hierzu wird vom Ethernet Paket die QMA und ZMA entfernt und durch die oben beschriebene Kombination ersetzt. Das Element 'OpenBeaconDecap' erzeugt Ethernet Pakete aus einem Click2OBD Frame, hierzu wird die Transformation vom 'OpenBeaconEncap' rückgängig gemacht. Damit die Hardware einfach und flexibel verwendet werden kann, können Parameter wie Kanal, Bitrate und Sendeleistung über vorhandene Elemente im Click konfiguriert werden. Für Bitrate und Sendeleistung können die Elemente 'SetTXRate' und 'SetTXPower', für den Kanal kann das Element 'BRN2SetChannel' verwendet werden. Diese 3 Elemente tragen (für jedes Paket) die Werte als Annotation ein. Im Prinzip kann das auch von anderen oder durch eigene Elemente erledigt werden.

Die Anwendungen im Click verwenden nur noch Ethernet Frames, deren einzige Einschränkung in der Paketgröße von max. 32 Byte liegt. Für größere Pakete müsste daher im Click eine geeignete Fragmentierung erfolgen. Hierfür gibt es Elemente für verschiedene Protokolle, z.B. 'IPFragmenter', 'TCPFragmenter', 'IP6Fragmenter', 'WifiFragment' und das dazugehörige Element für die Defragmentierung. Für Ethernet Pakete direkt gibt es zur Zeit keine Standard Elemente im Click. Bei der Forschung im HWL wurde eine Paketfragmentierung innerhalb des BRN Protokolls implementiert. Diese ermöglicht das Fragmentieren und Defragmentieren von beliebigen Paketen, also auch von Ethernet Paketen, mit einem geringen Overhead.

Kapitel 5

Evaluation

Mit Hilfe der Firmware und der Software für den Host (ObD) ist es möglich den Openbeacon USB Stick an die Geräte des HWL-Netzes anzuschließen und dort mit Hilfe von Click zu verwenden und zu evaluieren. Die Abbildung 5.1 zeigt den grundlegenden Messaufbau der USB Verbindung zwischen Host und Openbeacon USB Stick auf der linken Seite und auf der rechten Seite die Funkverbindung zwischen den Openbeacon USB Sticks. Für die Evaluation der Ende-zu-Ende Kommunikation wurden diese zwei Teilverbindungen getrennt voneinander untersucht, um ihren Einfluss auf die Ende-zu-Ende Kommunikation zwischen den Routern zu ermitteln.

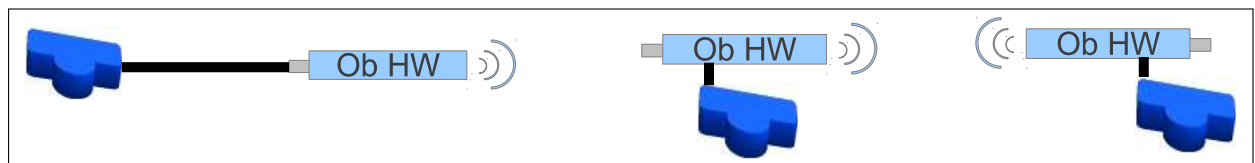


Abbildung 5.1: Messaufbau für die USB Performance (links), für die Funk Performance (rechts)

Der Openbeacon USB Stick kann mit verschiedenen Host Geräten wie z.B Laptop oder WLAN Router verwendet werden. Aufgrund der unterschiedlichen Hardwareleistungen erfolgen alle Messungen sowohl mit einem Laptop mit T5450 Prozessor (Dual Core 1.66 GHz/2GB RAM) bzw. einem Router mit Broadcom BCM95365R (200 MHz/32MB RAM) als Host.

Tabelle 5.1: Übersicht der Messungen

Abschnitt	Besonderheiten	gemessener Datendurchsatz
USB Schnittstelle	ohne drahtlose Schnittstelle	Openbeacon und Host
drahtlose Schnittstelle	geringen externe Interferenzen, ohne USB	zwei Openbeacons (unidirektional)
drahtlose Schnittstelle	geringen externe Interferenzen, ohne USB	zwei Openbeacons (bidirektional)
drahtlose Schnittstelle	mit externe Interferenzen, ohne USB	zwei Openbeacons (unidirektional)
drahtlose Schnittstelle	mit externe Interferenzen, ohne USB	zwei Openbeacons (bidirektional)
drahtlose Schnittstelle	mit externe Interferenzen, ohne USB	drei Openbeacons (bidirektional)
Ende-zu-Ende	ohne externe Interferenzen, mit USB und Click	zwei Openbeacons (unidirektional)
Ende-zu-Ende	ohne externe Interferenzen, mit USB und Click	zwei Openbeacons (bidirektional)
Ende-zu-Ende	mit externe Interferenzen, mit USB und Click	zwei Openbeacons (unidirektional)
Ende-zu-Ende	mit externe Interferenzen, mit USB und Click	zwei Openbeacons (bidirektional)
Ende-zu-Ende	mit externe Interferenzen, mit USB und Click	drei Openbeacons (bidirektional)

Die Tabelle 5.1 zeigt eine Übersicht der durchgeführten Messungen. Die einzelnen Messungen sollen hierbei die unterschiedlichen Einflüsse von USB-Verbindung und Funkverbindung untersuchen. Allen Messungen unter Verwendung Funkverbindung, wurden an verschiedenen Aufstellungsorten durchgeführt. Dadurch sollte der Einfluss von externen Geräten besser berücksichtigt werden. Darüber hinaus wurde durch schrittweises Hinzufügen von Sendern auch der Einfluss von anderen Openbeacons auf den Durchsatz überprüft. Im Abschnitt 'Ende-zu-Ende' wird die Leistungsfähigkeit der kompletten Verbindung untersucht, hierbei wurde eine virtuelle Verbindung zwischen zwei Click Instanzen aufgebaut. Durch Verwendung von verschiedenen Host Hardwarekonfigurationen (Laptop/Router) wird parallel auch der Einfluss Hardware untersucht. Beim Laptop wurden die Verbindung zwischen zwei Openbeacons auf eine kurzen Strecke und in einer Umgebung mit geringen externen Interferenzen gemessen.

In der Testumgebung mit externen Interferenzen wurden 3 Router mit jeweils einem Openbeacon in einem Raum verteilt, sie sollten ohne Hindernisse direkt miteinander kommunizieren können. Am Ende erfolgte eine Überprüfung der Verbindungsqualität bei geringer Datenrate in der Testumgebung, hier wurden 5 Router auf zwei Gebäude verteilt und die Datenübertragung gemessen. Die Testumgebung befand sich innerhalb des HWL, genau gesagt im Institut für Informatik (siehe Anhang 7.3 und 7.4). Im Grundriss werden die Bereiche hervorgehoben, in denen die Messungen erfolgten. Im Institut der Informatik werden für die Studenten WLAN Zugänge betrieben und hier erfolgen auch verschiedene Forschungen mit drahtloser Kommunikation. Die daraus auftretenden Interferenzen der drahtlosen Geräte oder Geräte mit elektromagnetischen Störfeldern (z.B. Mikrowelle) können Einfluss auf die Messungen haben.

5.1 USB Schnittstelle

Für die Evaluation wurde die Messung in zwei Teilschritten vorgenommen, im ersten Schritt wurde die Performance der USB Verbindung gemessen. Der Durchsatz über die USB Verbindung soll zeigen, wie schnell Daten vom PC/Router zum Openbeacon gelangen oder ob hier ein Engpass für die Ende-zu-Ende Kommunikation vorliegt. In der Implementierung werden die Daten in Paketen mit fester Größe verarbeitet, aus diesem Grund sollte der Einfluss der Paketgröße auf den Datendurchsatz gemessen werden. Für die Messanordnung wurde ein PC/Router mit dem Openbeacon USB Stick verbunden, wie es im linken Teil der Abbildung 5.1 gezeigt wird. Für unterschiedliche Paketgrößen sendete der PC/Router, Pakete mit verschiedenen Datenraten über die USB Verbindung (ObD) und erzeugt eine Statistik. Bei den Messungen wurde der maximale Durchsatz bei einer verwendeten Paketgröße ermittelt. Die nachfolgende Abbildung 5.2 zeigt den Durchsatz in Abhängigkeit der verschiedenen Paketgrößen.

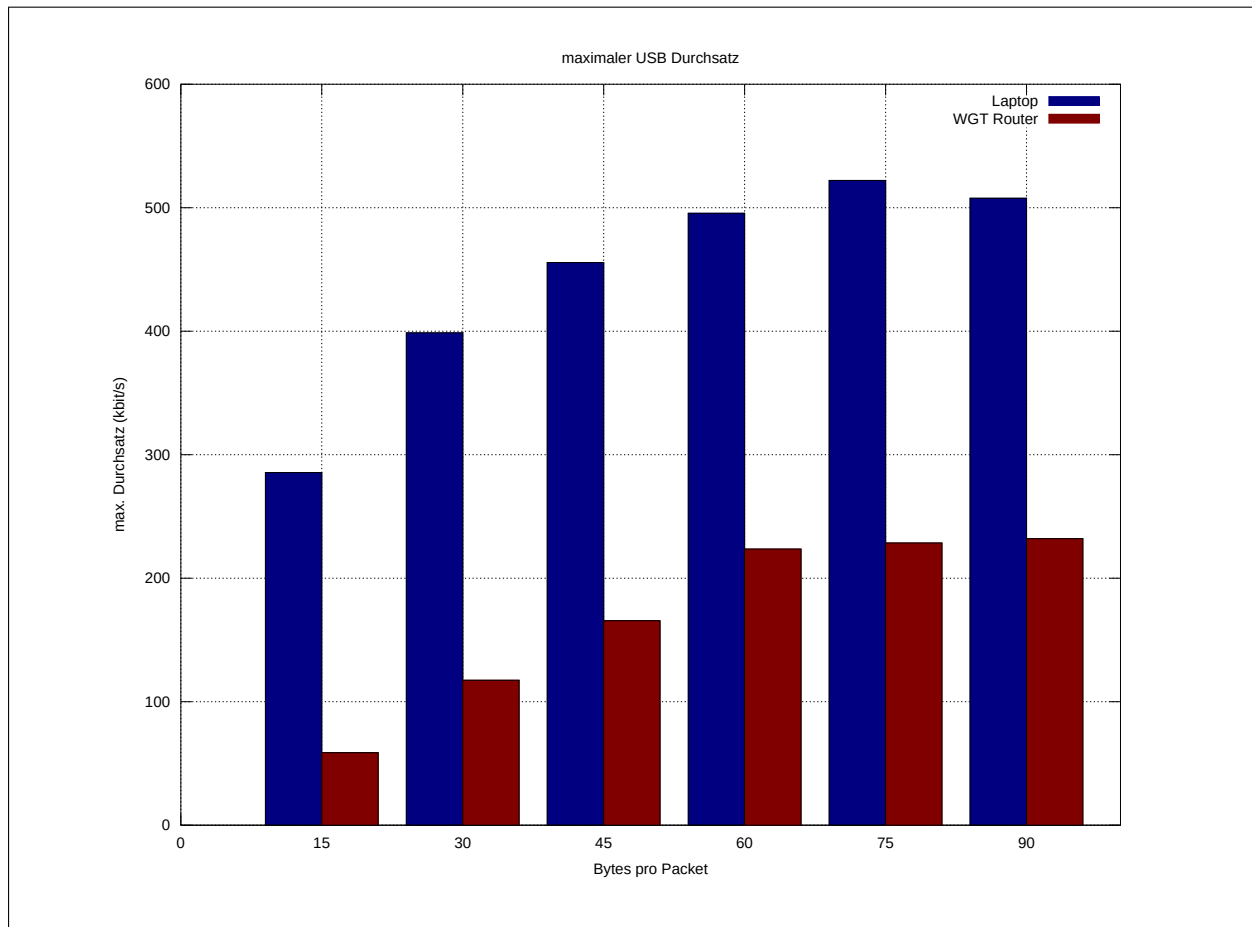


Abbildung 5.2: USB Durchsatz bei verschiedenen Paketgrößen in Kbit/s

Mit zunehmender Paketgröße stieg auch der Durchsatz durch die USB Verbindung an und erreichte sein Maximum bei einer Paketgröße von 75 Bytes. Der Einfluss der Paketgröße auf den Durchsatz ist hier deutlich zu erkennen und erreichte in der Spitze ca. 525 Kbit/s für den Laptop und 225 Kbit/s für den Router. Für diese Ergebnisse könnte es eine Erklärung im USB Standard[16] geben. Dort werden die Daten in einzelnen Frames übertragen, daher können größere Pakete die USB Verbindung besser ausnutzen. Der Unterschied zwischen Laptop (USB 2.0) und Router (USB 1.1) ist hier sehr deutlich und könnte an den unterschiedlichen Versionen des USB Standards liegen.

Bei der drahtlosen Schnittstelle ist die Paketgröße auf 32 Bytes begrenzt, bei diesem Wert konnte ein Durchsatz von ca 410 Kbit/s beim Laptop und ca. 130 Kbit/s beim Router erreicht werden. Bei ausreichend starker Host Hardwareleistung könnte der Durchsatz durch Zusammenfassen von Wireless Paketen zu einem USB Paket erhöht werden. Wenn man beim Laptop zum Beispiel zwei Wireless Pakete mit jeweils 32 Byte zusammen über USB versendet, zeigt Abbildung 5.2 bei einer Paketgröße von 64 Byte eine Steigerung des Durchsatz um 20 Prozent auf einen Wert von 500 Kbit/s.

5.2 drahtlose Schnittstelle

Im vorhergehenden Abschnitt wurde gezeigt welche Datenmenge über die USB Verbindung transportiert werden konnte. In diesem Abschnitt wird die Luftschnittstelle ohne den Einfluss der USB Verbindung genauer untersucht. Hier war von Interesse welche Datenmenge über diese Schnittstelle transportiert und wie die Verbindung sich bei Interferenzen verhält bzw. welche Datenmenge dort transportiert werden konnte. Die Messungen erfolgten in zwei Testszenarien. Zum einen in einer Umgebung mit geringen Interferenzen und in der Testumgebung mit vielen Interferenzen von anderen Geräten. In Tabelle 5.2 ist zu sehen, dass bei den Messungen verschiedene Signalstärken und Bitraten verwendet und für jede Parameterkombination die Datenrate langsam erhöht wurde. Auch bei der drahtlosen Schnittstelle wurde der maximale Durchsatz für die verwendeten Parameter ermittelt. Bei dieser Messung wurde der ObD für die Einstellung der Parameter verwendet, nur die Status- bzw. Konfigurationspakete sind über die USB Verbindung transportiert worden.

Tabelle 5.2: Eingestellte Parameter für die Messungen der drahtlosen Schnittstelle

Parameter	Werte	Einstellzeitpunkt
Paketgröße	32 Byte	fest implementiert
Signalstärke	1-4	Beginn der Messung
Rate	1 Mbit/s, 2 Mbit/s	Beginn der Messung

Messung mit geringen externen Interferenzen

Bei Messungen mit geringen externen Interferenzen konnte die Leistungsfähigkeit der Hardware (Datenrate) ermittelt werden. In diesem Testszenario wurden 2 Openbeacon USB Sticks mit einem Laptop verbunden, die Entfernung zwischen ihnen Betrug wenige Zentimeter. Die Messungen wurden in zwei Schritten durchgeführt. Im ersten Schritt sendete nur ein Openbeacon, während der andere die Pakete nur entgegen nahm. In einem zweiten Schritt sendeten beide Openbeacon USB Sticks.

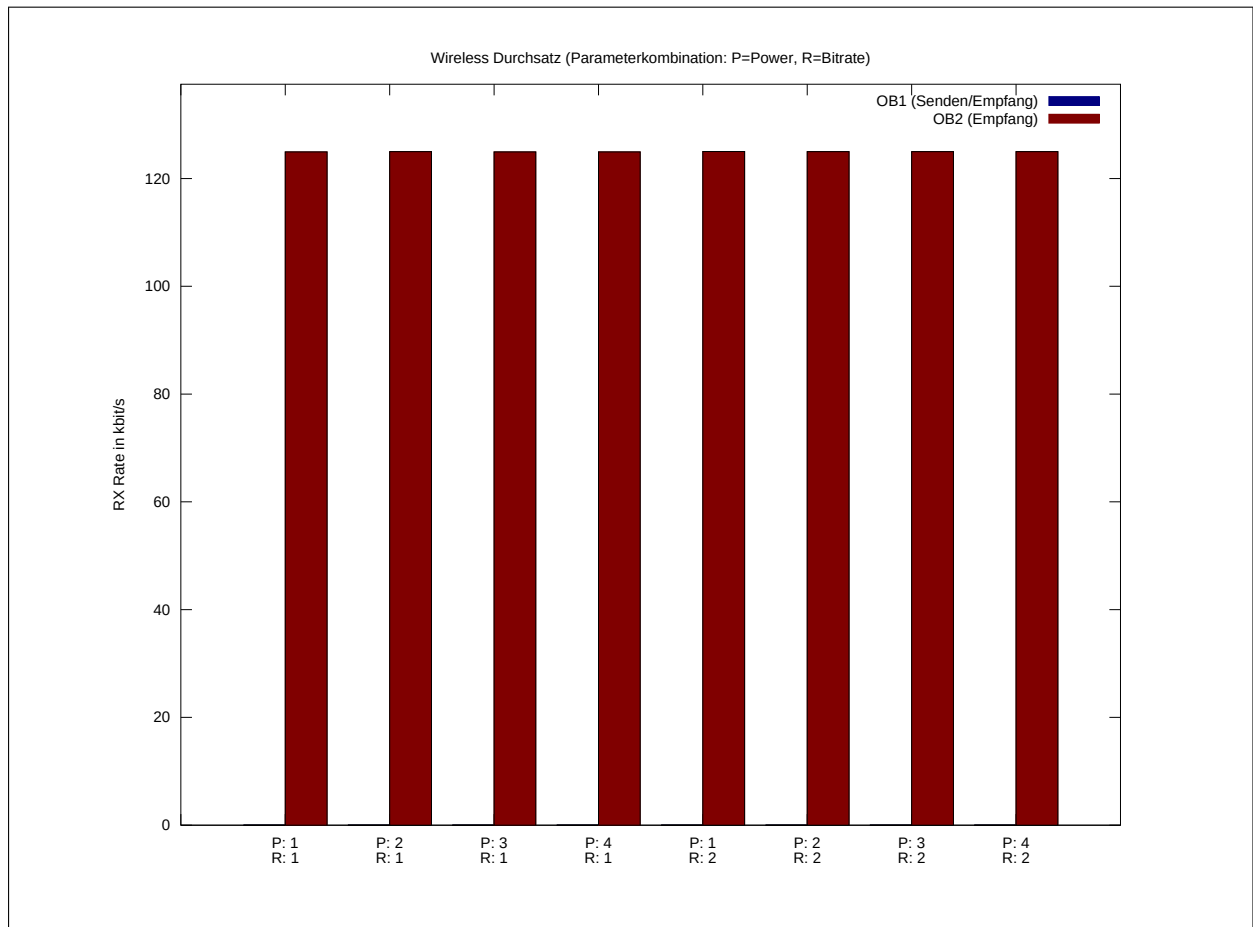


Abbildung 5.3: maximale Paketanzahl beim Laptop

Die Messungen wurden mit verschiedenen Sendeleistungen, Bitraten und verschiedenen Datenraten durchgeführt. Bei der Openbeacon Hardware werden die Daten in 32 Byte großen Paketen drahtlos übertragen. Wenn nur ein Openbeacon USB Sticks sendete, wurde unabhängig von der Bitrate oder Signalstärke in diesen Messungen maximal 125 Kbit/s durch die Firmware empfangen, wie Abbildung 5.3 zu sehen ist. In Abbildung 5.4 werden im Gegensatz dazu, bei gleichzeitigem Senden der Openbeacon USB Sticks, um die 33 Kbit/s von beiden Openbeacon USB Sticks empfangen, unabhängig von der Bitrate oder Signalstärke. Hier wurden nur ca. 55 Prozent der von beiden versandten Daten empfangen, alle übrigen Pakete gingen verloren.

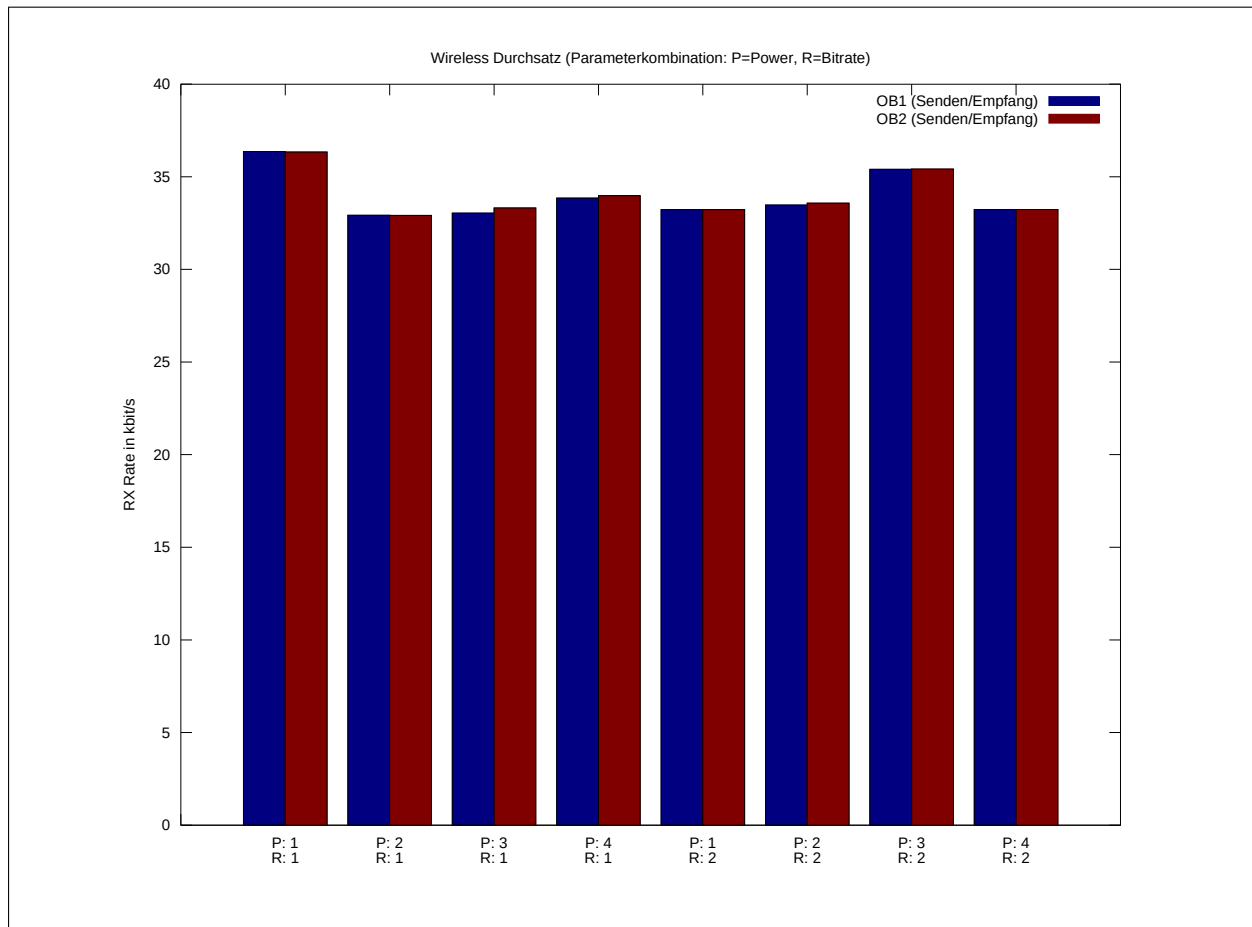


Abbildung 5.4: Empfangene Pakete bei beiden Sendern

Die maximale Datenrate von 125 Kbit/s liegt an der implementierten Umschaltzeit von 1 ms zwischen Empfangs- und Sendemodus. Diese Umschaltzeit wurde für die Stabilität der Firmware eingebaut. In dieser Zeit können keine weiteren Pakete versendet werden. Die maximale Datenrate von 33 Kbit/s bei zwei Sendern liegt ebenfalls an der erwähnten Umschaltzeit, da im Sendemodus nicht empfangen werden kann, und daher die Pakete der anderen Seite verloren gingen. Da nun beide Openbeacon USB Sticks Zeit für das Versenden von Paketen verbrauchen, gehen ihnen mit ansteigender Datenrate immer mehr Pakete des anderen verloren. Aber auch durch auftretende Kollisionen könnten einige Pakete zerstört worden sein.

Messung mit Interferenzen

Der Openbeacon soll für weitere Forschungen in einer Testumgebung eingesetzt werden, in der viele Interferenzen durch externe Geräte vorliegen, da im HWL verschiedene Funktechnologien und mehrere Router eingesetzt werden. Darüber hinaus können dort weitere Störungen auftreten, wie zum Beispiel durch Reflexionen an Wänden oder Metallträgern. In dieser Umgebung sollten daher die Messergebnisse überprüft werden, um eine realistischere Bewertung der Openbeacon Geräte zu erhalten. Der Messaufbau unterscheidet sich durch den getrennten Aufstellungsort der Openbeacons und damit die Überwindung

einer größeren Entfernung. Wie man in Abbildung 5.5 sehen kann, standen drei Openbeacon Geräte mehrere Meter voneinander entfernt und waren jeweils mit einem Router verbunden. Auch hier wurde in mehreren Schritten die Anzahl der Sender erhöht, um vergleichbare Ergebnisse zur vorhergehenden Umgebung zu erhalten.



Abbildung 5.5: Messaufbau im Testraum

Die Abbildung 5.6 zeigt, dass bei einem Sender mit 120 Kbit/s der empfangene Durchsatz nur wenig geringer war als bei der 'Messung mit geringen externen Interferenzen'. Die verwendeten Bitraten oder Signalstärken haben keine Abweichung bei den Messwerten ergeben.

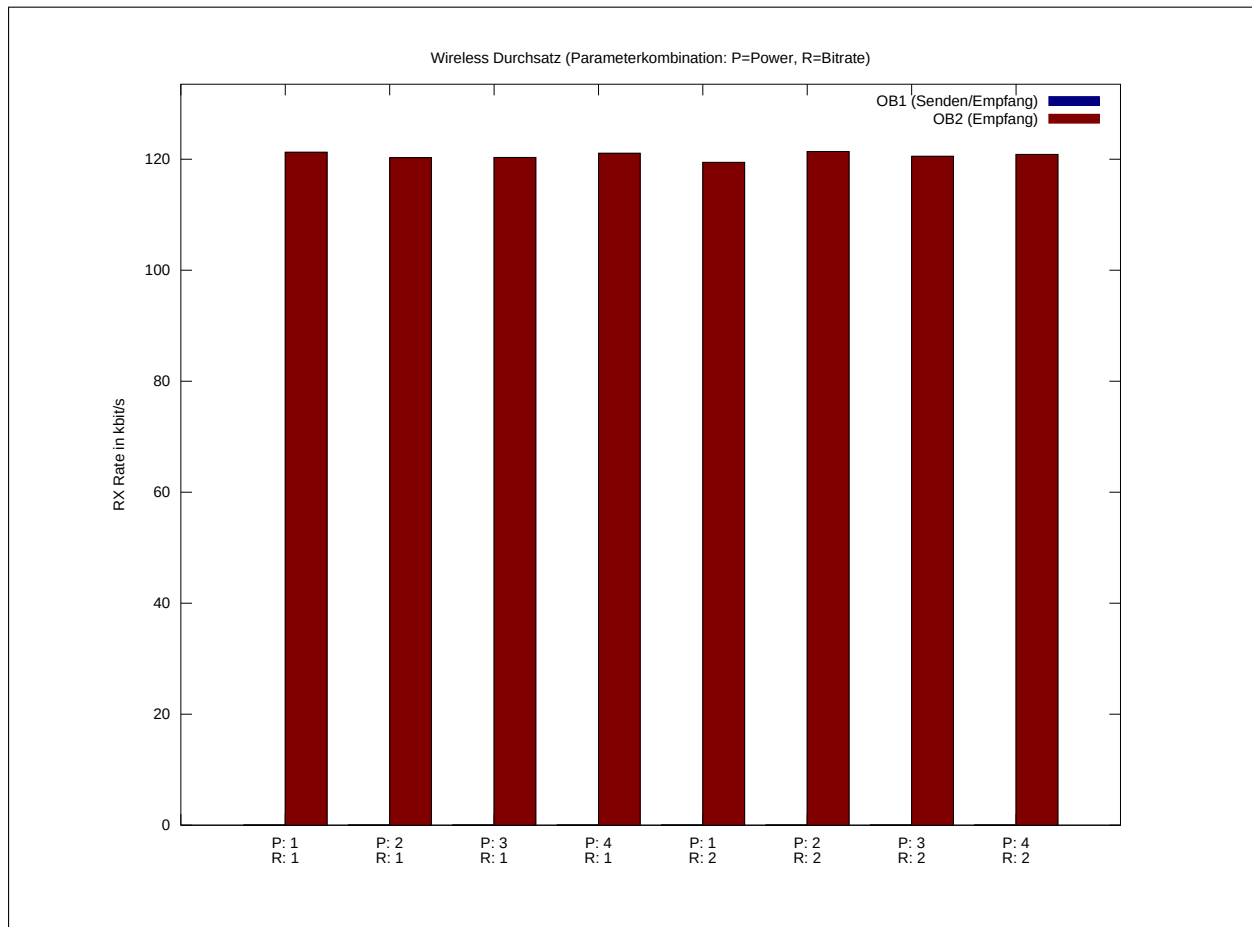


Abbildung 5.6: maximale Paketanzahl im Testraum

Im nächsten Schritt wurden zwei Sender/Empfänger in der Testumgebung verwendet und die Abbildung 5.7 zeigt bei fast allen Parameterkombinationen, dass beide Openbeacon Geräte um 35 Kbit/s empfangen haben. Im Vergleich zur 'Messung mit geringen externen Interferenzen' wurden hier mit ca. 63 Prozent mehr Daten von beiden Openbeacons empfangen. Dies könnte an der größeren Entfernung zwischen den Openbeacons gelegen haben. Gleichzeitig zeigte sich aber eine deutlichere Schwankung der Messwerte, im Vergleich zur 'Messung mit geringen externen Interferenzen'. Dies wird im Vergleich von Abbildung 5.4 mit Abbildung 5.7 deutlich.

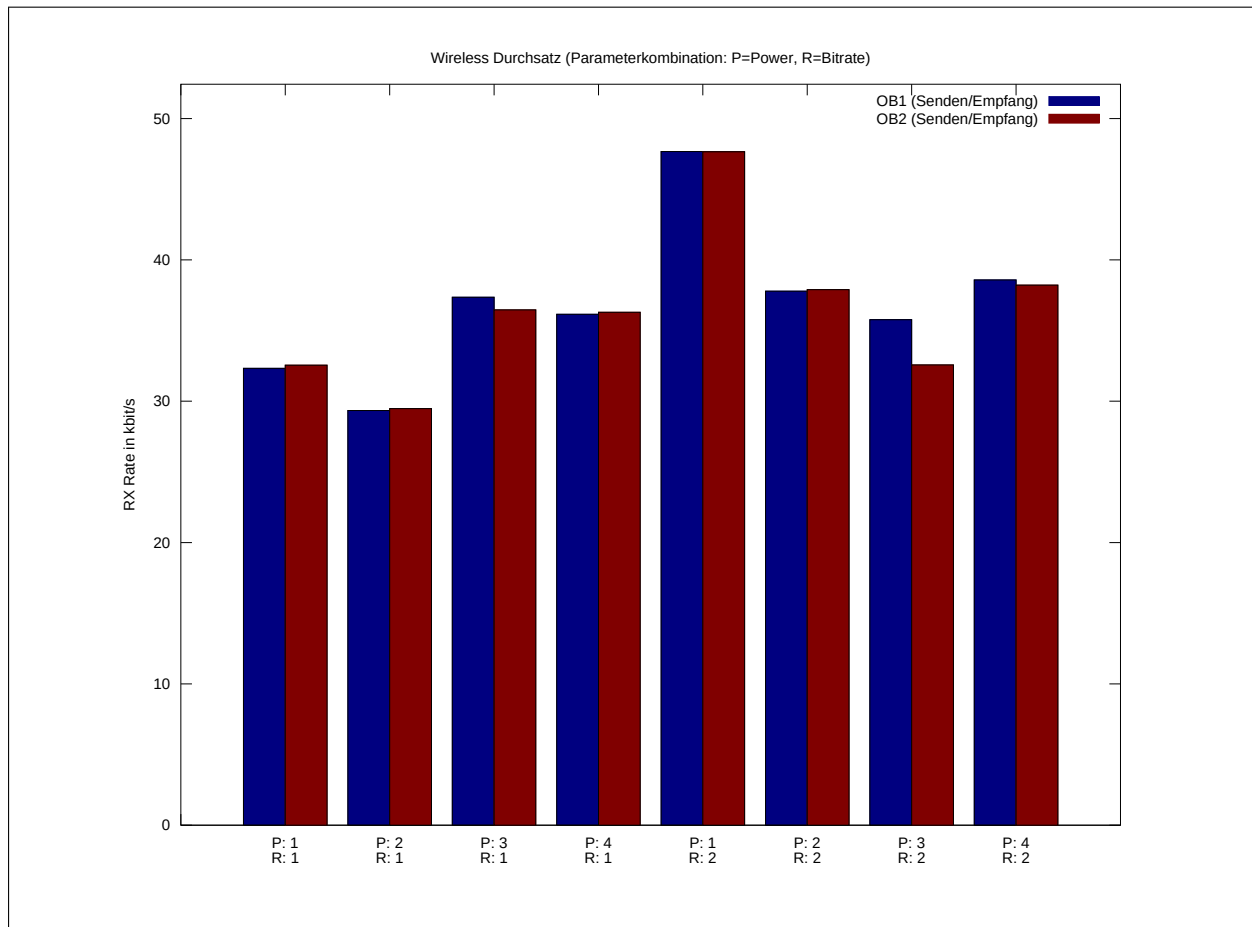


Abbildung 5.7: empfangene Pakete bei 2 Sendern im Testraum

Im letzten Schritt wurden 3 Sender/Empfänger verwendet und Abbildung 5.8 zeigt, dass die Openbeacon Geräte unabhängig von der Signalstärke einen Durchsatz von 49 bzw. 60 Kbit/s empfangen haben. Hier zeigt sich, im Gegensatz zu den vorhergehenden Messungen, ein deutlicher Unterschied zwischen den beiden Bitraten. Bei einer Bitrate von 2 Mbit/s konnten fast 20 Prozent mehr Pakete empfangen werden als bei 1 Mbit/s.

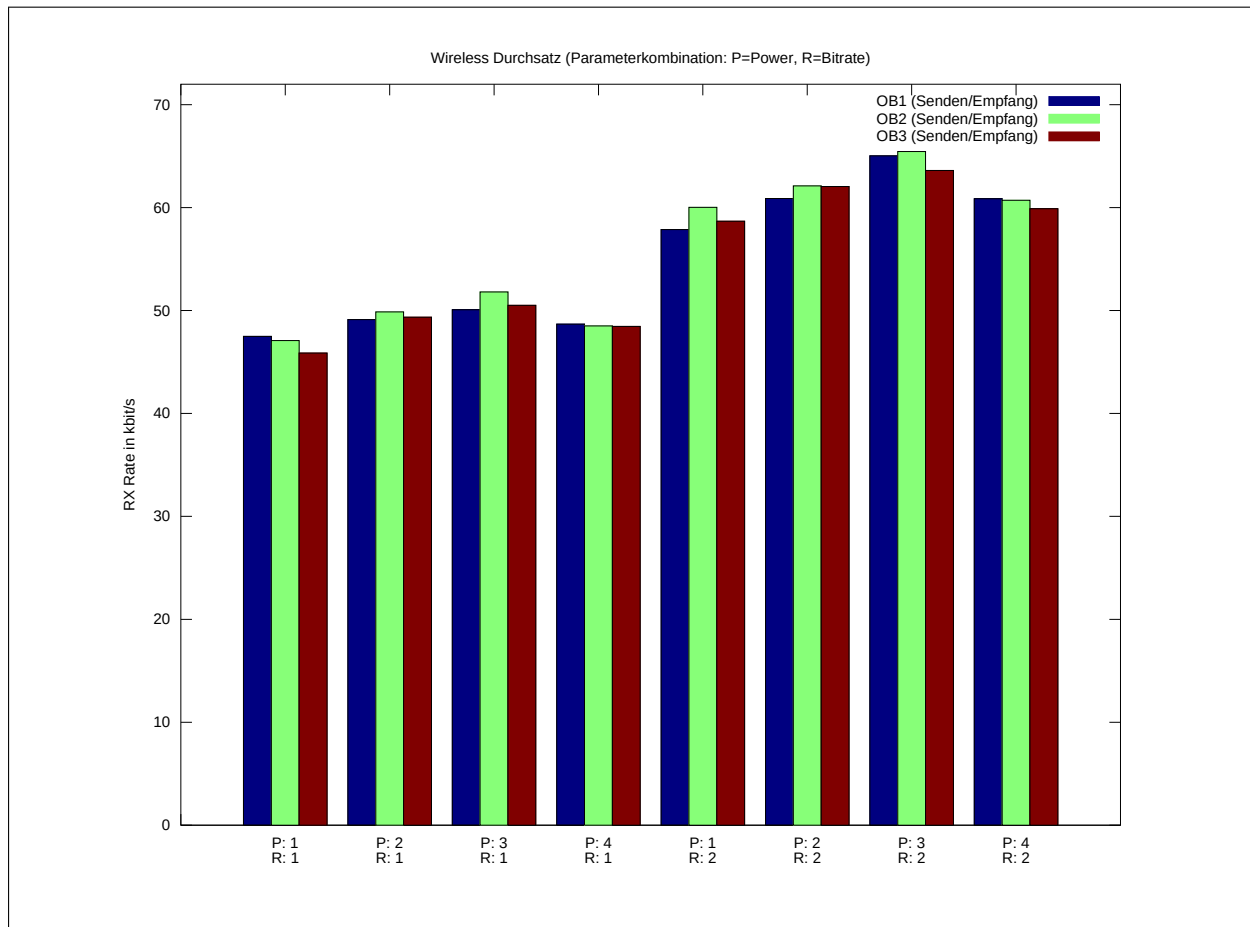


Abbildung 5.8: empfangene Pakete bei 3 Sendern im Testraum

Jeder Openbeacon empfängt die Pakete von zwei anderen Openbeacons und da die Werte alle empfangenen Pakete enthalten, unabhängig vom jeweiligem Sender, erhält man im Durchschnitt Werte von ca. 25 bzw. 30 Kbit/s pro Sender.

Die in diesen Messungen auftretende Reduzierung des Durchsatzes könnte wie im Abschnitt 'Messung mit geringen externen Interferenzen' an der langen Umschaltzeit zwischen Empfangs- und Sendemodus, an den auftretenden Kollisionen zwischen den beteiligten Openbeacons oder an Interferenzen durch externe Geräte gelegen haben.

5.3 Ende-zu-Ende

In den vorhergehenden Abschnitten wurde die Performance der USB und der drahtlosen Schnittstelle getrennt ermittelt. Im folgenden Abschnitt wird die Performance für den gesamten Kommunikationsweg untersucht. Die im Click erzeugten Pakete wurden über die Socketverbindung zum ObD transportiert. Dieser übertrug die Pakete über das serielle Gerät zur Openbeacon Firmware. Das Paket gelangte dann über Funk zu einem anderen Openbeacon und wurde dann über ObD zum Click transferiert. Hierbei

wurde der gleiche Messaufbau wie im Abschnitt 'drahtlose Schnittstelle' verwendet und mit verschiedene Bitraten und Signalstärke getestet. Die Messungen in diesem Abschnitt erfolgten wie im vorhergehenden Abschnitt in zwei Testszenarien.

Tabelle 5.3: Eingestellte Parameter für die Ende-zu-Ende Messungen

Parameter	Werte	Einstellzeitpunkt
Paketgröße	32 Byte	fest implementiert
Signalstärke	1-4	Beginn der Messung
Rate	1 Mbit/s, 2 Mbit/s	Beginn der Messung

Messung mit geringen Interferenzen

Es wurde im Abschnitt 'drahtlose Schnittstelle' eine Messung ohne die Verwendung von Click durchgeführt, in diesem Abschnitt sollte diese Messung unter Verwendung von Click wiederholt werden. Um die Messergebnisse vergleichen zu können, wurde die gleiche Messanordnung gewählt. Zwei Openbeacon Geräte wurden mit einem Laptop verbunden.

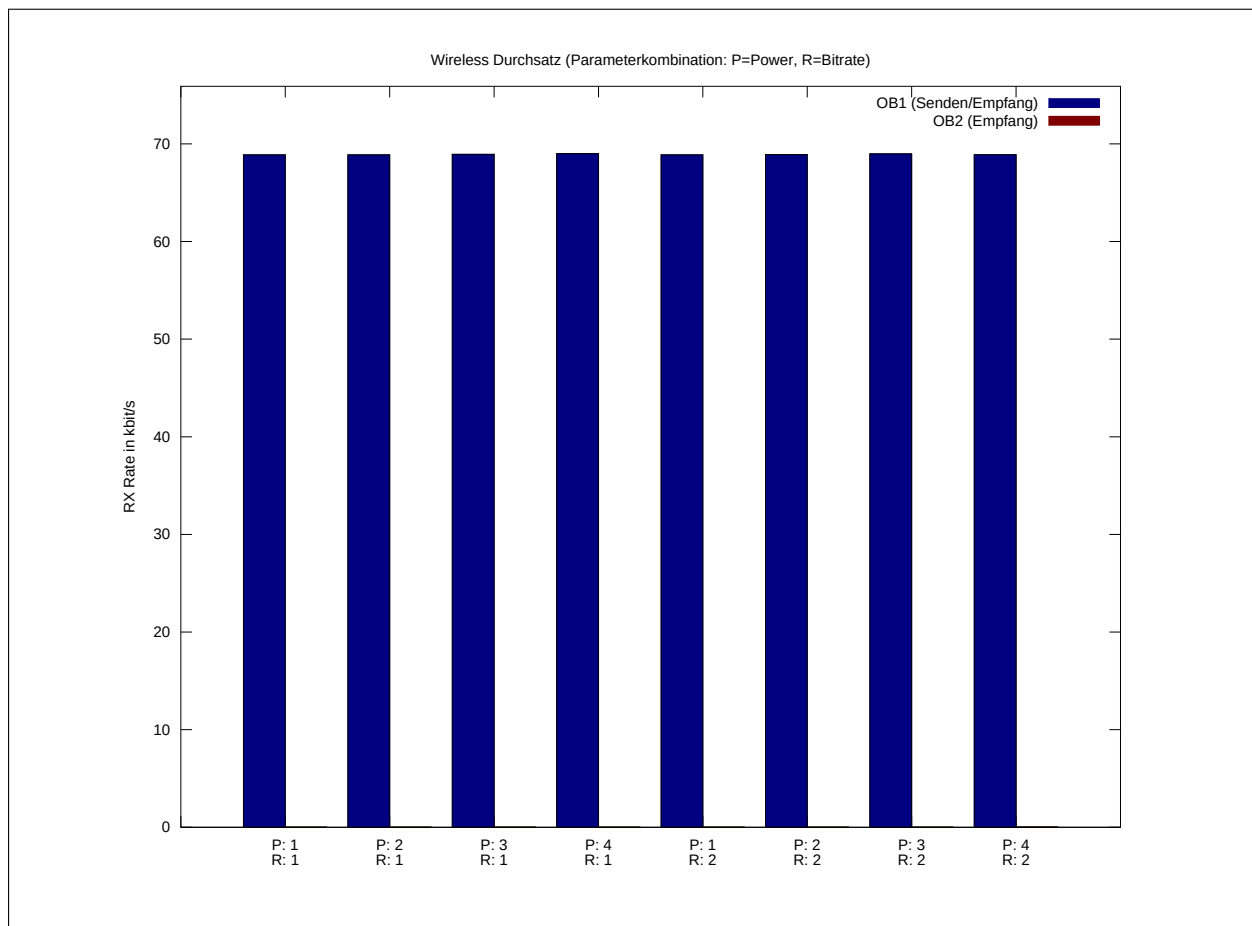


Abbildung 5.9: Gesamtperformance mit Click und einen Sender (Laptop)

Wenn man die Messungen aus Abbildung 5.9 mit Abbildung 5.3 vergleicht, hat sich bei der Verwendung von Click und ObD der Durchsatz erheblich verringert. Bei der Verwendung eines Senders zeigte sich, statt der vorhergehenden 125 Kbit/s, einen Wert von 69 Kbit/s und dies unabhängig von der verwendeten Bitrate oder Signalstärke. Das entspricht einer Reduzierung des Durchsatzes um fast 50 Prozent.

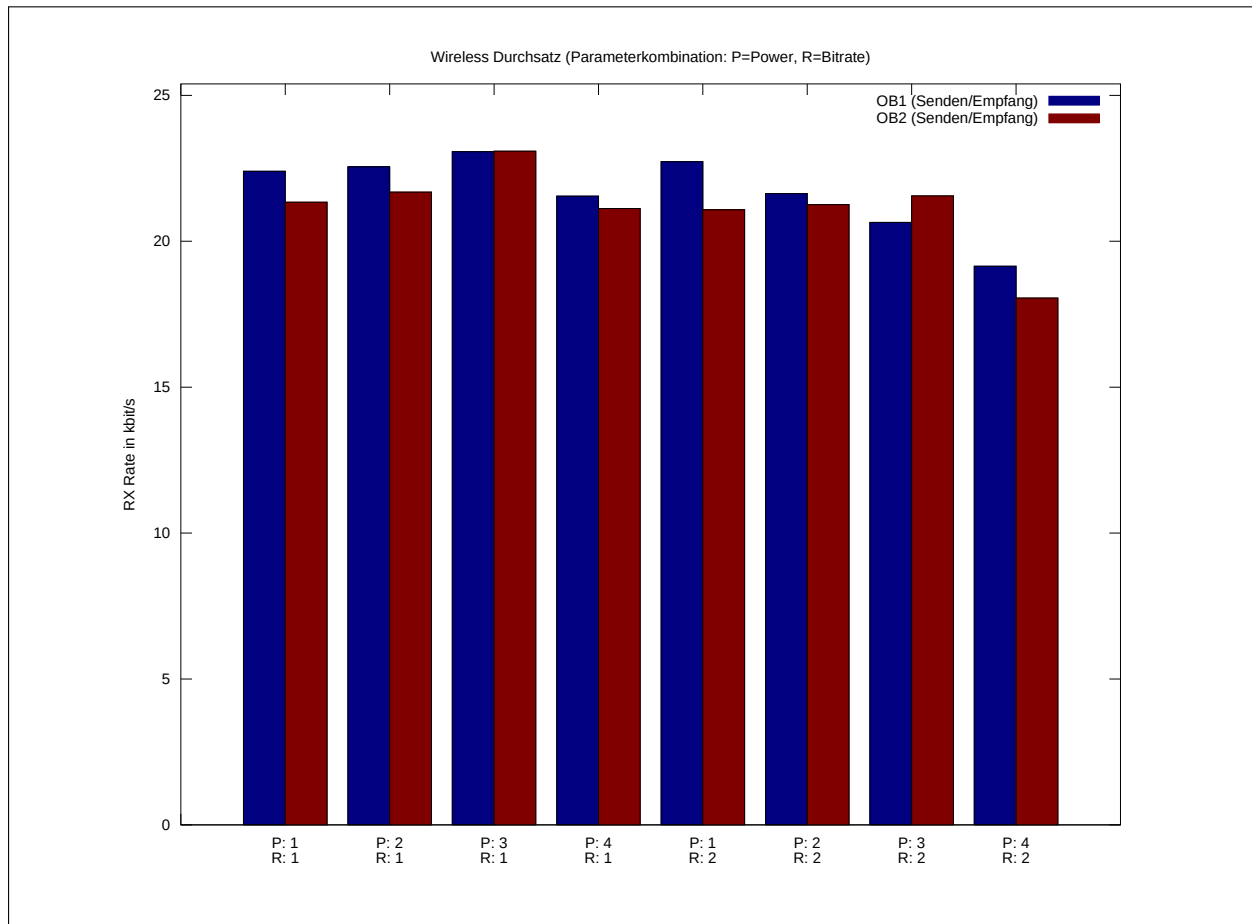


Abbildung 5.10: Gesamtperformance mit Click und zwei Sendern (Laptop)

Ein ähnliches Bild zeigte sich auch bei der Verwendung von zwei Sendern. Hier zeigen Abbildung 5.10 und Abbildung 5.4, dass die Reduzierung der Datenrate hier ebenfalls erheblich war. Statt der 33 Kbit/s wurden mit Click und ObD nur noch 22 Kbit/s erreicht, dies war ein Verlust von 33 Prozent.

Die Messungen der USB Schnittstelle zeigten einen Datendurchsatz von mehr als 210 Kbit/s und bei der drahtlosen Schnittstelle konnten 125 Kbit/s transportiert werden. Auch die Hardware des Laptop sollte genügend Datenpakete für den Transport bereitstellen können. Der Verlust von fast 50 bzw. 33 Prozent, sollte daher im wesentlichen an der begrenzten Hardwareleistung der Openbeacon Geräte gelegen haben. Paketverluste durch Interferenzen sollten bei den beiden Messungen der drahtlosen Schnittstelle ähnlich ausfallen. Für die Bearbeitung des gesamten Datentransfers reichte die Prozessorleistung des Openbeacons nicht aus, da sie hier im Unterschied zu den Einzelmessungen sowohl die Bearbeitung der

USB Pakete als auch den Empfang/Versand der drahtlosen Schnittstelle bewältigen mussten.

Messung mit Interferenzen

In diesem Abschnitt sollte die Messung mit der Verwendung von Click, in der Testumgebung des HWLs wiederholt und verglichen werden. Die Aufstellung der Openbeacon Geräte erfolgte wie in Abbildung 5.5 zu sehen ist. Der Unterschied zur Messung im Abschnitt 'drahtlose Schnittstelle' besteht darin, dass innerhalb des Clicks Pakete erzeugt und über den ObD zum Openbeacon USB Stick geleitet werden. Der Vergleich zwischen Abbildung 5.6 und Abbildung 5.11 zeigt eine deutliche Reduzierung des Durchsatzes, statt 120 Kbit/s waren es in dieser Messung nur 21 Kbit/s.

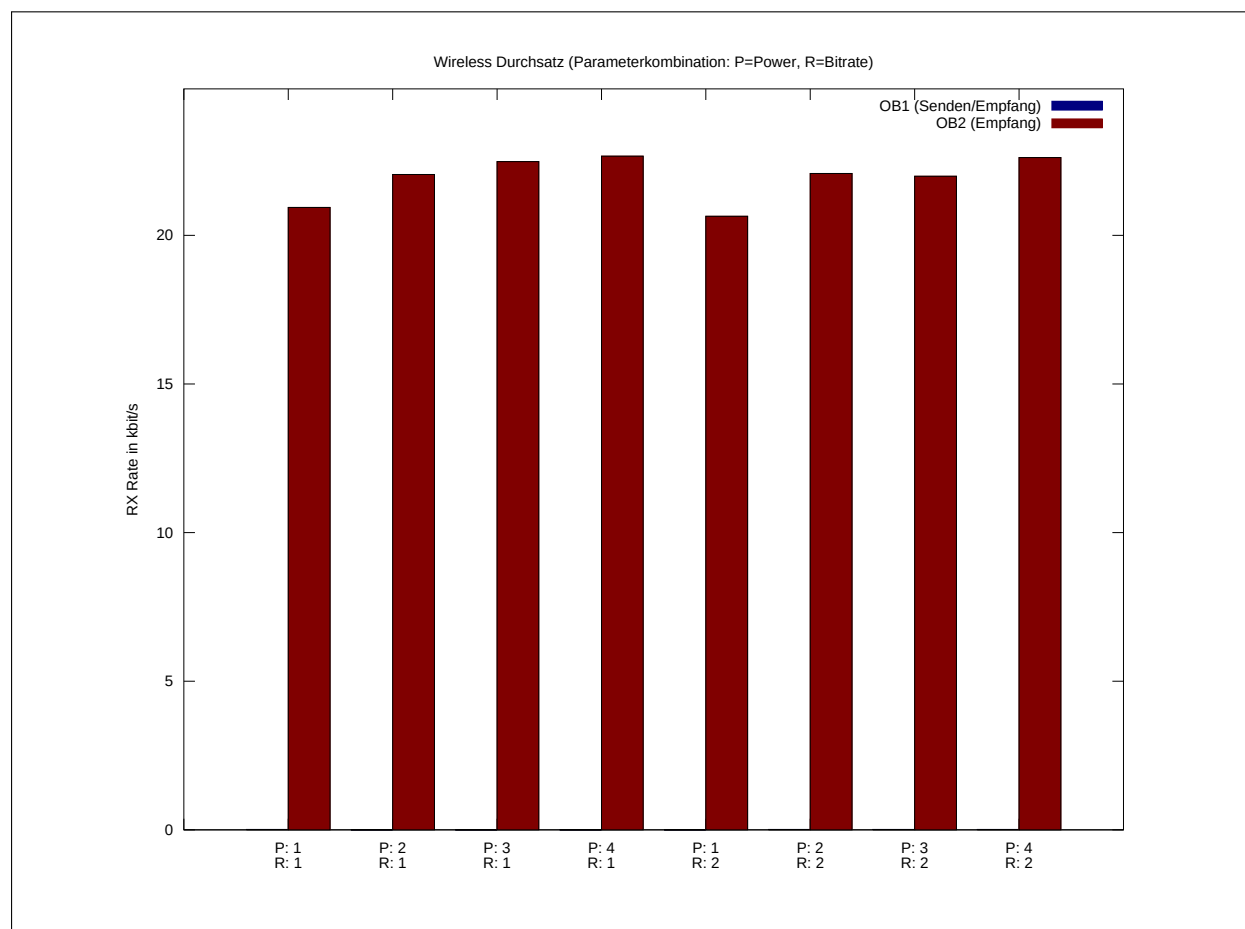


Abbildung 5.11: Gesamtperformance mit Click und einem Sender (Testumgebung)

Bei der USB Schnittstelle konnte ein Durchsatz von 125 Kbit/s erreicht werden und bei der drahtlosen Schnittstelle mit einem Sender konnte ein Durchsatz von 120 Kbit/s erreicht werden. Die Reduzierung in dieser Messung auf 21 Kbit/s könnte als Ursache die geringe Hardwareleistung des Openbeacons haben. Zusätzlich zur drahtlosen Schnittstelle musste der Openbeacon den Datentransfer über die USB Schnittstelle in beide Richtungen ermöglichen. Neben der Leistungsbegrenzung vom Openbeacon Gerät, spielte hier auch die verringerte Hardwareleistung des Routers eine Rolle. Er konnte nicht genügend Datenpakete

erzeugen, da mit dem Click und dem ObD nun deutlich mehr Rechenleistung benötigt wurde, als zur Verfügung stand.

Bei drei Sendern zeigt die Abbildung 5.12 einen Durchsatz von 15-25 Kbit/s, im Vergleich zu Abbildung 5.8 ein Verlust von 60-70 Prozent. Der Verlust könnte wie in der vorhergehenden Messung an der Hardwareleistung des Openbeacons bzw. des Routers gelegen haben. Mögliche Paketverluste durch Kollisionen/Interferenzen können den Durchsatz verringert haben. Der Messungsaufbau unterscheidet sich nur in der zusätzlichen Software, dadurch sollte der Paketverluste durch Interferenzen bei den beiden Messungen der drahtlosen Schnittstelle ähnlich ausfallen. Aus diesem Grund sollte hier die Hardwareleistung für die deutliche Reduzierung des Durchsatzes verantwortlich sein.

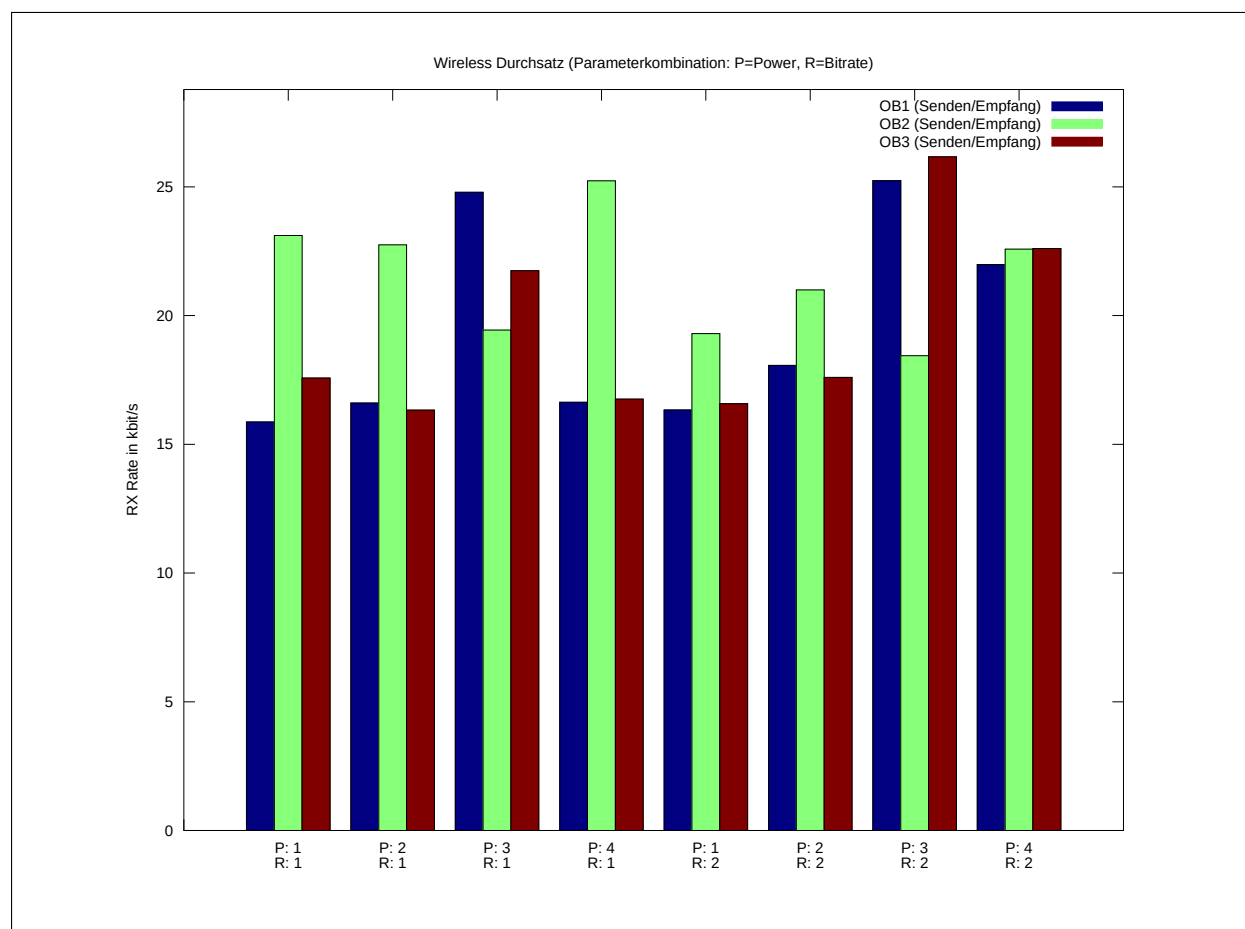


Abbildung 5.12: Gesamtperformance mit Click und drei Sendern (Testumgebung)

5.4 Indoor

In den vorhergehenden Abschnitten wurde der maximale Durchsatz für die USB und der drahtlosen Schnittstelle getrennt und zusammen untersucht. In diesem Abschnitt soll die Reichweite der Openbeacon Geräte in Gebäuden (bei Abschattung) abgeschätzt werden. Die Abbildung 5.13 zeigt wie die 5 Router

(wgt20, wgt71, wgt76, wgt79, wgt81) mit Openbeacon Geräten über zwei Gebäude verteilt wurden. Als Software ist wie im Abschnitt 'Ende-zu-Ende' das Click und der ObD zum Einsatz gekommen. Um die bei der vorhergehenden Übertragung auftretenden Verluste so gering wie möglich zu halten, z.B. durch Interferenzen zwischen den Openbeacons oder fehlender Hardware Ressourcen, wurde mit einer geringen Datenrate gesendet. Für die Messungen wurden die gleichen Parameter wie für die 'Ende-zu-Ende' Messungen verwendet, wie sie in Tabelle 5.3 zu sehen ist.

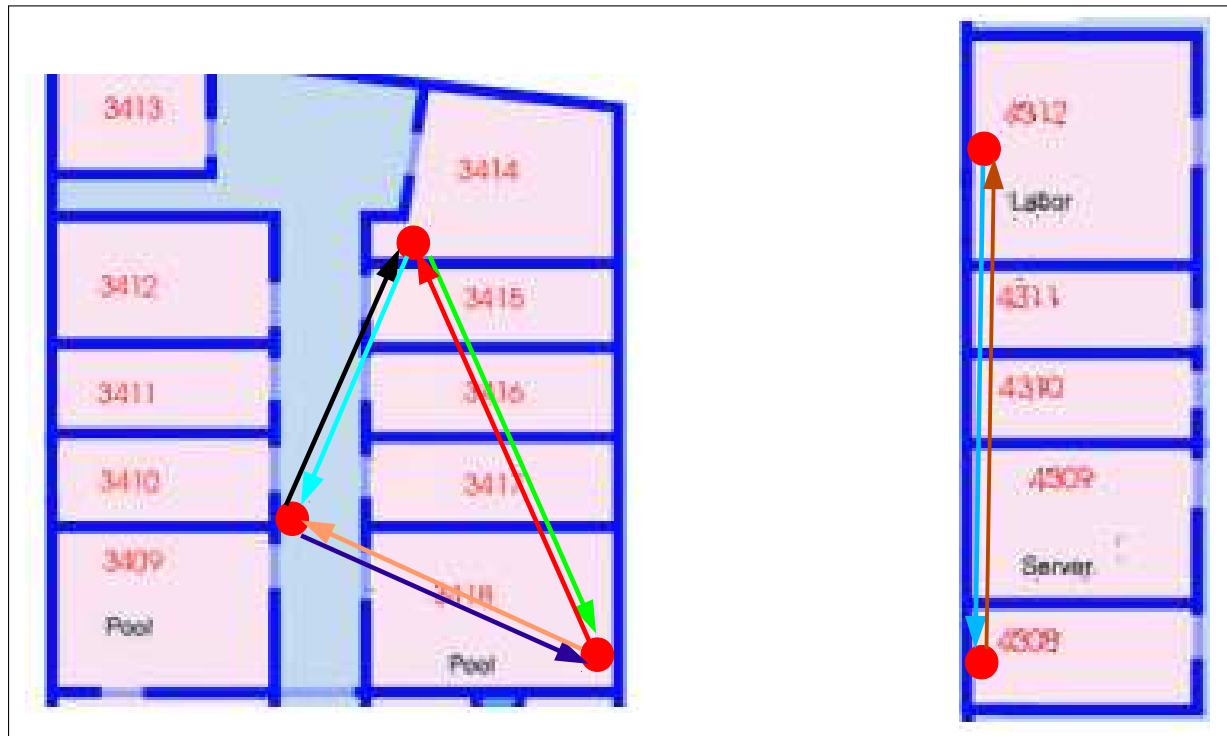


Abbildung 5.13: Messaufbau für die zweite Testreihe

In Abbildung 5.14 wurde die Paketübertragungsrate (packet delivery rate) für die wichtigsten Knotenverbindungen abgebildet. Die Verbindungen zwischen wgt71 und wgt81 lieferten weniger als 0.1 Prozent der versendeten Pakete, hier war die Abschwächung der Signale durch die Wände zu stark. Die Verbindung zwischen wgt71 bzw. wgt81 mit den anderen Knoten war überhaupt nicht vorhanden, hier wurde die Verbindung von den beiden Fenstern abgeschildert. Bei den Verbindungen zwischen wgt20, wgt76 bzw. wgt79 stieg die Paketübertragungsrate mit steigender Sendestärke deutlich an. Darüber hinaus war die Übertragung zwischen wgt20 und wgt76 erst mit höherer Sendestärke möglich, da die Dämpfung durch Wände so stark war, dass die niedrigste Sendestärke nicht reichte. Bei niedriger Sendestärke ist, im Vergleich zur Bitrate von 1 Mbit/s, die Paketübertragungsrate bei einer Bitrate von 2 Mbit/s gesunken. Wobei dieser Unterschied bei höchster Sendeenergie in dieser Messung nur sehr gering war.

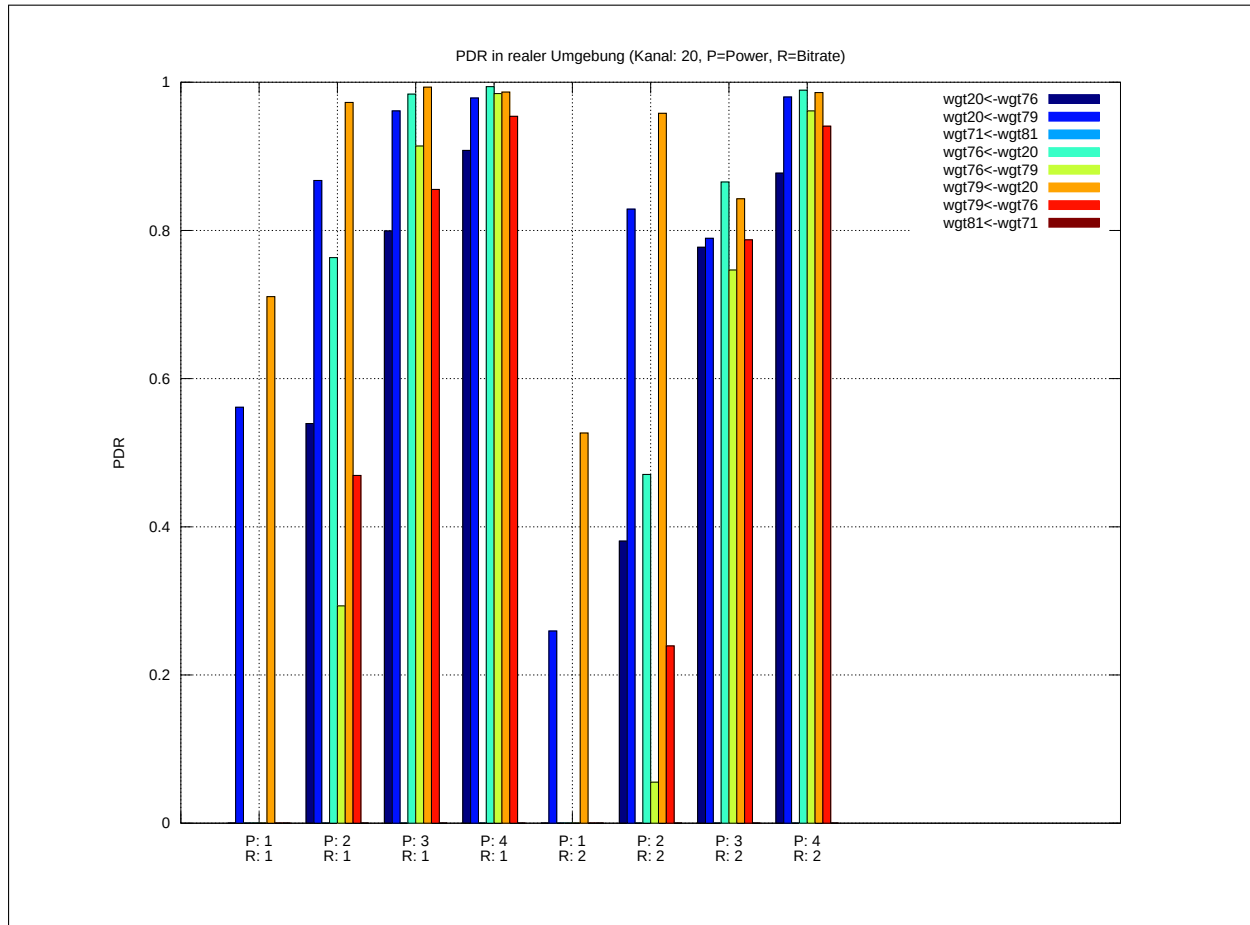


Abbildung 5.14: Verbindungsqualität bei Indoor Messung

Mit höherer Sendeleistung können Signaldämpfungen, wie sie z.B. durch Wände und Türen entstanden, kompensiert und dadurch die Verbindung zwischen den Openbeacons aufgebaut werden. Dabei unterscheidet sich die Signaldämpfung zwischen den beiden Bitraten, dies könnte an der verwendeten Kanalbreite gelegen haben. Bei einer Bitrate von 1 Mbit/s liegt Kanalbreite bei 1 MHz und bei 2 Mbit/s bei 2 MHz. Um die Verbindung zwischen den Openbeacons herstellen zu können, müssen die Abstände zwischen ihnen deutlich geringer als bei vergleichbaren Technologien (z.B. WiFi) sein. Verbindungen zwischen Openbeacon mit Router könnten durch einzelne Openbeacons verbessert werden. Da die Openbeacons ohne Host betrieben werden, könnten sie als Signalverstärker (Repeater) eingesetzt werden. Wenn die Aufstellungsorte der Openbeacons geeignet gewählt werden, lassen sich mit ihnen geeignete Netzwerkstrukturen aufbauen.

Kapitel 6

Fazit

Der Gegenstand dieser Arbeit war die Integration der Openbeacon USB-Sticks in die Entwicklungsumgebung des Humboldt Wireless Lab. Hierzu wurden Elemente für Click, der Openbeacon Daemon als Verbindungsschicht und die Firmware für den Openbeacon entwickelt. Im Fokus lag dabei eine einfache Anwendbarkeit dieser Lösung für Entwickler. Für sie ist es daher leicht möglich kleine Ethernet Paket über den Openbeacon zu verteilen. Die Schnittstelle im Click ist mit einem Element für das Versenden, einem Element für das Empfangen der Datenpakete und den schon vorhandenen Elemente für die Konfiguration von Kanal, Bitrate und Signalstärke leicht zu benutzen.

Um sich einen Überblick über die Leistungsfähigkeit dieser Hardware zu verschaffen, wurden verschiedene Messungen durchgeführt. Für die Funkverbindung wurden Bitraten von 1 bzw. 2 Mbit/s eingestellt und dabei eine max. Datenrate von 125 Kbit/s ermittelt, diese Datenrate könnte über die USB Verbindung zwischen Router und Openbeacon gerade so gesendet werden. Der Atmel-Prozessor war der größere Flaschenhals, hier wird durch die hohe Funktionalität die Datenrate auf weniger als 25 Kbit/s reduziert. Deshalb wird deutlich, dass die Funktionalität auf dem Openbeacon auf das Mindestmaß reduziert werden muss. Auch die Auslastung des Router-Prozessors hat seinen Teil zu der geringeren Datenrate beigetragen. Hier hat man zusätzlich zu einer Optimierung des OpenBeacon Dämons auf dem Host, auch die Möglichkeit die Hardware aufzustocken. Stärkerer Prozessor, größerer Arbeitsspeicher und der USB Standard 2.0 können die Datenrate aber nur um wenige Prozent erhöhen, wie durch die Messung mit dem Laptop deutlich wurde.

Die Datenrate ist sehr stark von der Hardware des Openbeacon abhängig, evtl. werden spätere Hardware Versionen diese Schwachstelle verbessern können. Bei gleichzeitiger Verwendung von mehreren Openbeacons, kann man mit einer Datenrate von 15-20 Kbit/s nur eine kleine Anwendung entwickeln. Hierzu würden Informationsverteilung in kleinem Rahmen gehören, etwa die Mitteilung über Position und momentanem Zustand des Routers, an dem der Openbeacon angeschlossen ist. Ein Beispiel dafür wäre die Organisation einer optimalen Ausnutzung der Hardware des HWLs. Zu den Informationen könnte der momentane Mitarbeiter gehören, der mit dieser Hardware gerade Messungen vornimmt oder Anwendungen testet. Wenn kein anderes Openbeacon das Medium verwendet, kann eine höhere Datenrate von 21 Kbit/s und bei stärkerer Router Hardware von ca. 70 Kbit/s erreicht werden. Hier können Protokolle wie TDMA für die MAC Zugriff entwickelt werden, um diese hohe Datenrate effizienter nutzen zu können.

Die Fähigkeit zu einem Software Defined Radio erhält der Openbeacon durch die Änderungsmöglichkeit in der Firmware. Hier kann ein begrenzter Einfluss auf die Media Access Control Schicht genommen werden. Das Medium kann darauf geprüft werden, ob es gerade belegt ist und darauf reagiert werden.

Allerdings hat man ohne weitere Hardwareunterstützung eine sehr grobe Zeitsteuerung. D.h. man kann als kleinste Einheit bei diesem Prozessor max. das Vergehen 1 ms erkennen und in diesem Intervall die Pakete versenden. Es wäre aber möglich ein Protokoll zu entwickeln, ähnlich wie unterteiltes ALOHA oder TDMA. Es könnten auch Verfahren wie Frequency Hopping Spread Spectrum (FHSS) eingesetzt werden, um die auftretenden Kollisionen zwischen den Openbeacons zu reduzieren. Durch effiziente Ausnutzung der Funkkanäle könnte die Anzahl der Openbeacons pro Kanal reduziert und damit die Datenrate erhöht werden. Darüber hinaus könnte man auch Überlegungen anstellen, wann die einzelnen Pakete bestätigt werden. Hier wäre ein Ansatz mehrere Pakete zusammen zu bestätigen oder die Bestätigung mit dem Senden eines anderen Paketes mit zu übertragen, um weniger Verwaltungspakete zu transportieren. Die Ansätze die man hier mit den Openbeacons untersuchen könnte, würden sicherlich Erkenntnisse bei der Entwicklung neuer Protokolle für bestehende oder neue Funktechnologien liefern.

Kapitel 7

Anhang

Help for obd	
--help	- print this help text
-O[O1] [O2] ... [On]	- USB Device for OpenBeacon HW sample: obd -d 0 1 - for device ttyACM0
und ttyACM1	
-Q[TIME]	- Exittime (0 for no terminate)
-D[microseconds]	- time for display status infos
-g	- send data to openbeacon
-e	- with echo from openbeacon
-p[PACKET_SIZE]	- size of a packet (5...95)
-I[bytes per seconds]	- rate for sending data to openbeacon
-r	- random data activate
-C[channel]	- channel for wireless
-R[rate]	- bitrate for wireless
-P[power]	- power for wireless
-t[DATA_RATE]	- data rate for test (wireless only)
-d	- daemon mode activate
-path	- directory for output-files
-noclick	- deaktiviert Verbindung Click <-> OBd

Abbildung 7.1: Parameterliste des obd

Help for obd-config	
--help	- print this help text
-O[O1] [O2] ... [On]	- USB Device for OpenBeacon HW sample: obd-config -d 0 1 - for device 0
oder 1	
-C[channel]	- channel for wireless
-R[rate]	- bitrate for wireless
-P[power]	- power for wireless
-t[DATA_RATE]	- data rate for test (wireless only)
-path	- directory for output-files

Abbildung 7.2: Parameterliste des obd-config



Abbildung 7.3: Institut für Informatik der HU Berlin

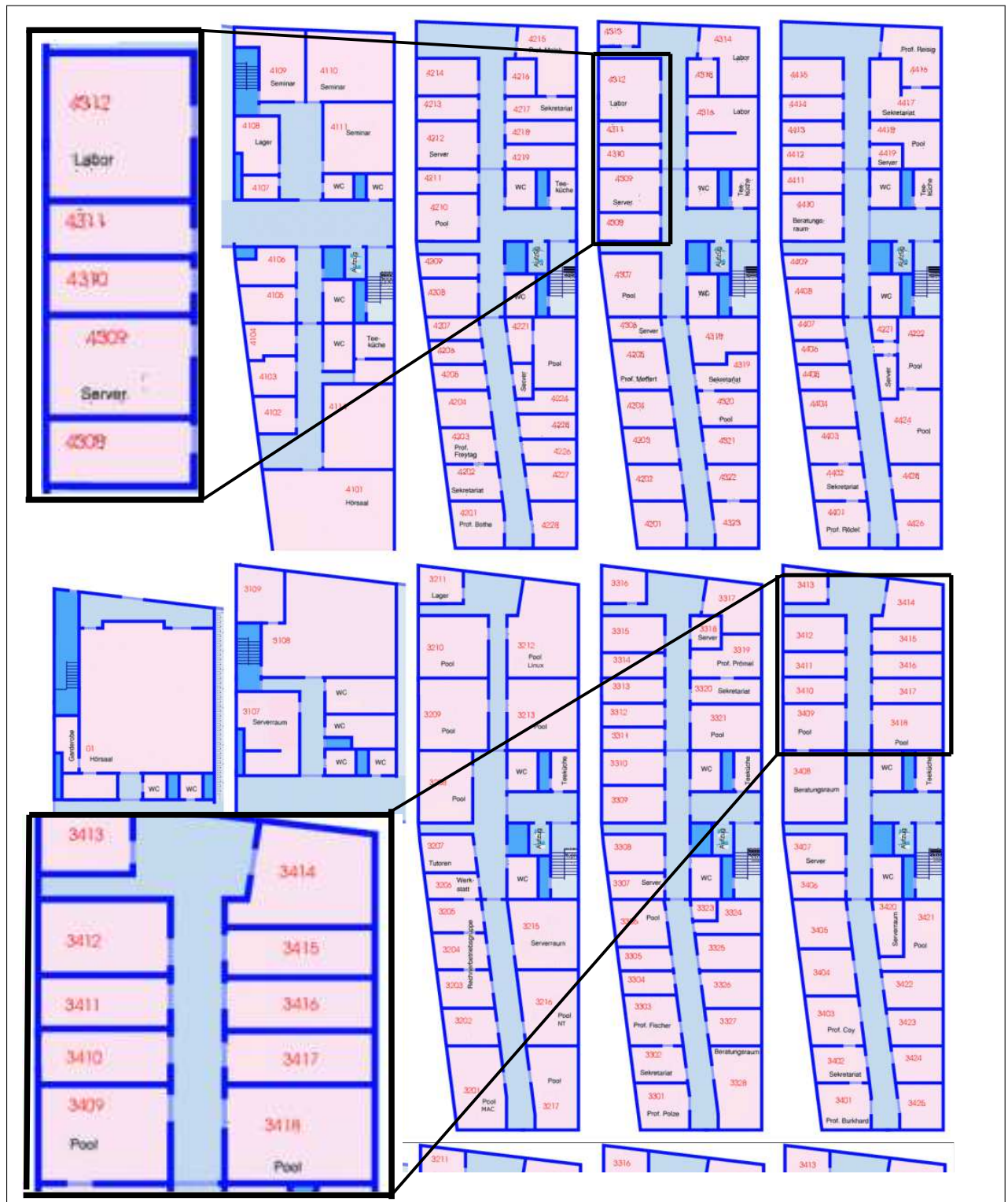


Abbildung 7.4: Grundriss des Institutes für Informatik der HU Berlin

Abbildungsverzeichnis

1.1	Erdbeben Anwendung	3
1.2	Wireless erweiterte Kommunikationsanwendung	4
2.1	ISO/OSI Diagramm	7
3.1	aktuelle Netzwerk Architektur für das HWL	10
3.2	Anordnungsbeispiel für Openbeacon	11
3.3	eine kleine Click Router Anwendung (Auszug)	13
4.1	schematische Darstellung der Anwendung	15
4.2	Protokoll Frames für das Versenden über USB	16
4.3	Softwarearchitektur der Firmware	17
4.4	Softwarearchitektur des ObD	19
4.5	Anwendungsbeispiel	19
5.1	Messaufbau für die USB Performance (links), für die Funk Performance (rechts)	21
5.2	USB Durchsatz bei verschiedenen Paketgrößen in Kbit/s	23
5.3	maximale Paketanzahl beim Laptop	25
5.4	Empfangene Pakete bei beiden Sendern	26
5.5	Messaufbau im Testraum	27
5.6	maximale Paketanzahl im Testraum	28
5.7	empfangene Pakete bei 2 Sendern im Testraum	29
5.8	empfangene Pakete bei 3 Sendern im Testraum	30
5.9	Gesamtperformance mit Click und einen Sender (Laptop)	31
5.10	Gesamtperformance mit Click und zwei Sendern (Laptop)	32
5.11	Gesamtperformance mit Click und einem Sender (Testumgebung)	33
5.12	Gesamtperformance mit Click und drei Sendern (Testumgebung)	34
5.13	Messaufbau für die zweite Testreihe	35
5.14	Verbindungsqualität bei Indoor Messung	36
7.1	Parameterliste des obd	39
7.2	Parameterliste des obd-config	40
7.3	Institut für Informatik der HU Berlin	40
7.4	Grundriss des Institutes für Informatik der HU Berlin	41

Tabellenverzeichnis

3.1	einstellbare Parameter für die Funkübertragung	12
4.1	Firmware Versionen	17
5.1	Übersicht der Messungen	22
5.2	Eingestellte Parameter für die Messungen der drahtlosen Schnittstelle	24
5.3	Eingestellte Parameter für die Ende-zu-Ende Messungen	31

Literaturverzeichnis

- [1] *Informationen zum Calradio*. <http://calradio.calit2.net/calradio1a.htm>, 2012
- [2] Click - Modular Router [Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti]. (2000)
- [3] *ClickWatch - An experimentation framework for communication network test-beds* [Markus Scheidgen, Anatolij Zubow, Robert Sombrutzki]. <http://www-mobile.ecs.soton.ac.uk/home/conference/wcnc2012/papers/p3334-scheidgen.pdf>, 2012
- [4] *Allgemeine Informationen zum CSMA/CA*. <http://de.wikipedia.org/wiki/CSMA/CA>, 2012
- [5] *Allgemeine Informationen zu IEEE_802.15.4*. http://de.wikipedia.org/wiki/IEEE_802.15.4, 2012
- [6] *Allgemeine Informationen zum ISO/OSI Modell*. <http://de.wikipedia.org/wiki/OSI-Modell>, 2012
- [7] *Erdbebenfrühwarnsystem für Istanbul* [Prof. Dr. Joachim Fischer, Prof. Dr. Jens-Peter Redlich, Prof. Dr. Jochen Zschau, Dr. Claus Milkereit, Dr. Matteo Picozzi, Dr. Kevin Fleming, Dipl.-Inf. Björn Lichtblau, Dipl.-Inf. Frank Kühnlenz, Dipl.-Inf. Ingmar Eveslage, Dipl.-Inf. Mihai Brumbulli]. http://sar.informatik.hu-berlin.de/research/publications/SAR-PR-2010-10/SAR-PR-2010-10_.pdf, 2010
- [8] *Allgemeine Informationen zu Kognitiv Radio*. <http://www.esk.fraunhofer.de/de/projekte/CoRa.html>, 2012
- [9] *Allgemeine Informationen zu Openbeacon USB*. http://www.openbeacon.org/OpenBeacon_USB, 2012
- [10] *Openbeacon Source Code*. http://www.openbeacon.org/OpenBeacon_git_repository, 2012
- [11] *OpenWRT Linux distribution for embedded devices*. www.openwrt.org, 2012
- [12] *Allgemeine Informationen zu PCF*. http://de.wikipedia.org/wiki/Point_Coordination_Function, 2012
- [13] *Informationen zu Pololu Wixel*. <http://www.pololu.com/catalog/product/1336>, 2012
- [14] *Informationen zu Microsofts Sora*. <http://research.microsoft.com/en-us/projects/sora/>, 2012
- [15] *Informationen zu Tmote Sky*. <http://www.snm.ethz.ch/Projects/TmoteSky>, 2012
- [16] *Allgemeine Informationen zu USB*. http://de.wikipedia.org/wiki/Universal_Serial_Bus, 2012
- [17] *Informationen zu USRP*. <http://www.ettus.com/products>, 2012