Verlängerung der Reichweite eines WLAN mit Hilfe von portablen Raspberry Pi als Brotkrumen

---

Tim Scherffig

Frankfurt University of Applied Sciences

FB 2: Informatik und Ingenieurwissenschaften

Bachelorthesis

Verlängerung der Reichweite eines WLAN mit Hilfe von portablen Raspberry Pi als Brotkrumen

Eingereicht zum Erlangen des akademischen Grads
Bachelor of Science

Tim Scherffig
Matrikelnummer: 981844

Referent Prof. Dr. Christian Baun
FB 2: Informatik und Ingenieurwissenschaften
Frankfurt University of Applied Sciences

Korreferent Prof. Dr. Matthias Deegener
FB 2: Informatik und Ingenieurwissenschaften
Frankfurt University of Applied Sciences

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen sind, sind als solche kenntlich gemacht. Die Arbeit hat in gleicher Form noch keiner anderen Prüfbehörde vorgelegen.

Frankfurt am Main, 10.07.2019

Tim Scherffig

Abstract

Die vorliegende Bachelorarbeit beschreibt unterschiedliche Implementationen um ein WLAN, mit Hilfe einer Kette aus mobilen Raspberry Pi Einplatinencomputern, zu erweitern. Sie enthält die notwendigen Beschreibungen um eine solche Kette, sowohl mit Repeatern, als auch als Mesh zu konfigurieren. Diese Lösungen werden unterschiedlichen Netzwerktests unterzogen um die Vor- und Nachteile der jeweiligen Lösungen aufzuzeigen. Darüber hinaus enthält sie Informationen zum mobilen Betrieb eines Raspberry Pi mit einer handelsübliche Powerbank, sowie der möglichen Energiesparoptionen des Raspberry Pi.

Abstract (English)

This bachelor thesis describes different implementations to extend a WLAN utilizing a chain of portable Raspberry Pi single-board computers. It contains the necessary descriptions to configure such a chain via repeaters or as a mesh. Network tests are being used to understand and show the advantages and disadvantage of each implementation. Additionally it includes information about operating a Raspberry Pi as a portable device using an ordinary powerbank, as well as methods to conserve energy within the Raspberry Pi.

Danksagung

Hiermit möchte ich mich bei meinen Eltern für die Unterstützung während des Studiums und insbesondere bei dieser Arbeit bedanken.

Mein Dank gilt Prof. Dr. Baun für die Betreuung und Bereitstellung dieses Themas sowie der verwendeten Mittel. Zudem bedanke ich mich bei Prof. Dr. Deegener für die Bereitschaft sich als Korreferent zur Verfügung zu stellen.

Abkürzungsverzeichnis

AP = Access Point
B.A.T.M.A.N. = Better Approach To Mobile Adhoc Networking
ITU = International Telecommunication Union
MTU = Maximum Transmission Unit
PDV = Packet Delay Variation
RPi = Raspberry Pi

Inhaltsverzeichnis

1 Einleitung

Schlechte WLAN-Abdeckung ist ein häufiges Alltagsphänomen. Insbesondere auf größeren Grundstücken kann dies zum Problem werden, wenn sich der Internet-Router im Haus befindet, man aber am anderen Ende des Gartens oder in einem eventuellen Gästehaus noch das WLAN dieses Routers nutzen möchte. Abhilfe schaffen in diesem Fall WLAN-Repeater oder Access-Points, um die Reichweite oder den Datendurchsatz zu verbessern. Befindet sich kein Stromanschluss nah genug an dem Bereich, den man abdecken möchte, sind die Optionen sehr begrenzt. Größer wird dieses Problem wenn man einen größeren oder weiter entfernten Bereich abdecken möchte, als mit einem einzigen WLAN-Repeater erreichbar ist. Soll zum Beispiel das WLAN in einen nah gelegenen Wald verlängert oder das WLAN eines landwirtschaftlichen Betriebs auch auf den umliegenden Feldern nutzbar sein, existieren dafür keine trivialen Möglichkeiten. Der in dieser Arbeit verfolgte Lösungsansatz ist die Nutzung von kostengünstigen Raspberry Pi Einplatinencomputern. Diese werden angelehnt an das Märchen Hänsel und Gretel, wie Brotkrumen verteilt, um ein WLAN zu erweitern.

1.1 Zielsetzung

Das Ziel dieser Arbeit ist es zu prüfen, inwiefern es möglich ist die Reichweite eines WLAN mit Hilfe einer Kette aus Raspberry Pi Einplatinencomputern zu verlängern. Hierfür werden die Raspberry Pi in bestimmten Abständen wie Brotkrumen ausgelegt. Dabei wird jeder Raspberry Pi über einen Akku betrieben, um von stationären Stromquelle unabhängig zu sein.

Es soll geprüft werden, ob und wie dies technisch möglich ist und welche Auswirkungen unterschiedliche Software-Lösungen und WLAN-Hardware auf Reichweite, Datendurchsatz und Zuverlässigkeit einer solchen Kette haben. Des Weiteren wird untersucht wie lang ein Akku einen Raspberry Pi Einplatinencomputer mit Strom versorgen kann und welche Maßnahmen zur Verringerung des Stromverbrauchs sinnvoll möglich sind.

2 Stand der Technik

Dieses Kapitel beschreibt Kriterien für eine Realisierung des gewünschten Systems und bewertet bereits existierende Lösungen für ähnliche Aufgaben.

2.1 Kriterien zur Bewertung existierender Lösungen

Als Kriterien für eine Lösung zählen die Verbindungsqualität (in Form von Datendurchsatz, Zuverlässigkeit und Reichweite) und Akkulaufzeit. Das Ziel ist dabei eine Lösung zu finden die den Kriterien möglichst gut entspricht. Auf diese Art und Weise soll auch herausgefunden werden, welche Ergebnisse mit einer Kette aus Raspberry Pi Einplatinencomputern zu erwarten sind.

2.2 Existierende Lösungen

Fertige kommerzielle Lösungen, die für ein solches Projekt in Frage kämen und ähnliche Aufgaben erfüllen, gibt es bereits. Eine mögliche Lösung ist mobiles Internet, bei dem je nach Standort mit 4G theoretisch bis zu 600 Mbit/s möglich sind[3]. Dies bietet allerdings nur einen einzelnen Zugang zum Internet und kein vollständiges Netzwerk. Für den Fall, dass kein Zugang zu einem anderen Router, der mit dem Internet verbunden ist existiert, kann ein tragbarer 4G-Router theoretisch als Zugangspunkt zum Internet für das Netzwerk benutzt werden. Die leistungsstärksten dieser Modelle können dabei eine Akkulaufzeit von circa 15 Stunden erreichen[14]. Dies ähnelt der Aufgabe allerdings nur insofern, dass sie Internet an einen Ort bringen an dem es ursprünglich nicht verfügbar war. Eine Lösung die näher an einer Kette aus Raspberry Pi Einplatinencomputern ist, wäre beispielsweise der Kortex Xtend Lite. Dabei handelt es sich um einen batteriebetriebener WLAN Repeater mit Unterstützung für einen sogenannten Mesh-Modus[12]. Dieser bietet ein gewisses Maß an Modifizierbarkeit durch den Anschluss von Akkus beliebiger Größe oder einer anderen Antenne sowie durch die quelloffene Firmware.

3 Verwendete Technologien

Das nachfolgende Kapitel ist eine Übersicht über die verwendeten Hardwarekomponenten sowie Software. Es soll einen kurzen Einblick über derren Funktionalität bieten und aufzeigen warum diese ausgewählt wurden.

3.1 Raspberry Pi

Raspberry Pi (RPi) ist eine Reihe von Einplatinencomputer die hauptsächlich für Bastler und zum Lernen von Programmierfähigkeiten entwickelt wurde. Die seit 2012 produzierten RPi umfassen 11 Modelle die unterschiedliche Funktionen und Schnittstellen bieten[22]. In dieser Arbeit fanden sowohl das Model *Raspberry Pi 3 Model B*, als auch das Model *Raspberry Pi 3 Model B+* Verwendung. Im Folgenden sind die für diese Arbeit relevanten Merkmale dieser zwei Modelle aufgelistet[18].

Tabelle 3.1: Raspberry Pi Übersicht

Modell	**RPi 3 Model B**	**RPi 3 Model B+**
CPU	4× Cortex-A53 1.2 GHz	4× Cortex-A53 1.4 GHz
RAM	1 GB	1 GB
Energieverbrauch	400 mA typischer Verbrauch ohne angeschlossene Peripherie	500 mA typischer Verbrauch ohne angeschlossene Peripherie

Beide Modelle bieten zudem integrierte WLAN-Module. Das Modul des RPi 3 B+ bietet zusätzliche zu 2,4Ghz- auch eine Unterstützung für 5Ghz-WLAN. Im Gegensatz zu den kleineren Modellen, wie dem Raspberry Pi Zero, verfügen sie außerdem über vier USB-Anschlüsse. Da teilweise mehrere USB-WLAN-Adapter an einem Raspberry Pi verwendet werden sollen ist dies ein wichtiger Faktor.

3.2 Raspbian

Raspbian ist ein für Raspberry Pi optimiertes Betriebssystem, dass auf der Linux Distribution Debian basiert[19]. Es wird von der Raspberry Pi Foundation in einer Desktop- und einer Lite-Version angeboten[17].

3.3 USB-WLAN-Adapter

Sowohl der *Raspberry Pi 3 B* als auch der *Raspberry Pi 3 B+* bieten einen internen WLAN-Adapter. Da für diese Arbeit jedoch Reichweite und Verbindungsstärke eine entscheidende Rolle spielen, wurde sich entschlossen zusätzlich USB-WLAN-Adapter mit einer externe Antenne zu verwenden. Des Weiteren war bei der Auswahl dieser Adapter insbesondere die Kompatibilität zu Raspberry Pi und Raspbian wichtig. Ausgewählt wurden der *TP-Link TL-WN722N V.3* und der *Edimax EW-7612UAn V2 Adapter*. Die für hier wichtigen, vom Hersteller angegebenen Spezifikation dieser beiden Adapter stellen sich dabei wie folgt dar.[2, 13]

Tabelle 3.2: WLAN-Adapter Spezifikationen

	TP-Link TL-WN722N V.3	**Edimax EW-7612UAn V2**
Antennengewinn	4dBi	3dBi
WLAN-Standards	IEEE802.11b/g/n	IEEE802.11b/g/n
Höchste Signalrate	11n: Bis zu 150 Mbit/s (dynamisch)	11n (20 MHz): MCS0-15 (bis) zu 144 Mbit/s) 11n (40MHz): MCS0-15 (bis zu 300 Mbit/s)

TP-Link teilt bei der Signalrate nicht zwischen 20 und 40 Mhz auf, somit ist nicht klar was gemeint ist. Da 20 Mhz allerdings häufiger verwendet wird, ist anzunehmen, dass sich die 150 Mbit/s auf 20 Mhz beziehen.

3.4 B.A.T.M.A.N. Advanced

B.A.T.M.A.N. Advanced (im Folgenden in der Schreibweise *batman-adv*) ist eine Implementation des Routing Protokolls B.A.T.M.A.N. (Better Approach To Mobile Adhoc Networking), dass auf Netzwerkschicht 2 arbeitet und von der Freifunk-Community entwickelt wurde, um das zuvor genutzte Protokoll OSLR (Optimized Link State Routing) zu ersetzen.[25, 23]

Es agiert dabei als vermaschtes Netzwerk und emuliert einen virtuellen Netzwerks-witch aller verbundenen Knoten. Das Protokoll arbeitet dezentral, jeder Knoten kennt nur seine direkten Nachbarn, welche jeweils mit einer Metrik versehen werden. *Batman-adv* findet in dieser Arbeit Verwendung, da es ein weit verbreitetes Protokoll und seit Version 2.6.38 teil des Linux-Kernels ist. Zudem is es nicht notwendig, dass der WLAN-Adapter den Standard 802.11s für Mesh-Netzwerke unterstützt.[24]
Als Konfigurations- und Debug-Tool für *batman-adv* steht batctl zur Verfügung[20].

3.5 iperf

Iperf ist eine quelloffene Software zum Testen des Datendurchsatzes von Netzwerken. Ein Computer agiert dabei als Server auf den sich der zweite Computer als Client verbindet. Die Software kann TCP- und UDP-Datenströme erzeugen und bietet diverse Einstellungsmöglichkeiten. So sind unter anderem Tests mit mehreren gleichzeitigen Verbindungen, sowie Einstellungen zu Dauer von Tests und Bandbreiteneinstellungen für UDP-Tests möglich.[9]

4 Grundlagen

Das folgende Kapitel beschreibt die notwendigen Schritte für den Aufbau einer Netzwerkkette aus Rasberry Pi Einplatinencomputern. Es enthält eine Beschreibung der Installation von Rasbian auf dem RPi, sowie der Erstellung eines Access-Points(APs). Teil ist außerdem ein Basistest der beiden USB-WLAN-Adapter um Vergleichswerte zu gewinnen.

4.1 Raspberry Pi Grundsetup

Das folgenden Kapitel beschreibt das Grundsetup eines Raspberry Pi Einplatinencomputers anhand des *Raspberry Pi 3 B* und *Raspbian Stretch Lite (2019-04-08)*. Die verwendete Hardware und Software umfasst:

Hardware

- Raspberry Pi 3 B V.2
- TP-Link TL-WN722N V.3 USB-WLAN-Adapter
- Edimax EW-7612UAn V2 USB-WLAN-Adapter
- USB-Netzteil mit einer Stromstärke von mindestens 2 Ampere oder eine vergleichbare Stromversorgung
- Ravpower 20.000mAh Powerbank RP-PB006
- Micro-USB Kabel zur Verbindung mit dem Netzteil
- microSD-Karte
- microSD-Kartenlesegerät
- Tastatur und Monitor

- Ein PC auf Basis von Windows, Linux oder macOS
- USB-Verlängerungskabel
- Netzwerkkabel(optional)

Software

- Raspbian Stretch Lite (2019-04-08)
- balenaEtcher oder vergleichbares Programm zum Brennen von Images auf microSD-Karten
- WinRAR oder vergleichbares Packprogramm

Um zunächst die microSD-Karte mit dem Betriebssystem zu bespielen Zuerst wird die microSD-Karte in das Lesegerät gesteckt und dieses mit dem PC verbunden. In *BalenaEtcher* wählen nun die Schaltflächen *Select image* und *Select drive* das jeweilige Raspbian-Image, in diesem Fall *2019-04-08-raspbian-stretch-lite.img*, sowie die microSD-Karte aus. Ein Klick auf *Flash!* startet den Vorgang des Brennens. Ist dieser abgeschlossen kann die Karte entnommen und in den passenden Steckplatz auf dem Raspberry Pi gesteckt werden. Der Raspberry Pi wird mit Tastatur, Monitor, sowie dem Netzteil verbunden und so eingeschaltet. Die amerikanische Tastatur ist als Standard festgelegt. Somit ist auf einer deutschen Tastatur darauf zu achten *z* statt *y* und *ß* statt - zu drücken. Um sich nach dem ersten Start einzuloggen, sind der Standardbenutzername *pi* und das Passwort *raspberry*. Der Befehl `sudo raspi-config` öffnet nun die Grundeinstellungen von Rasbian.

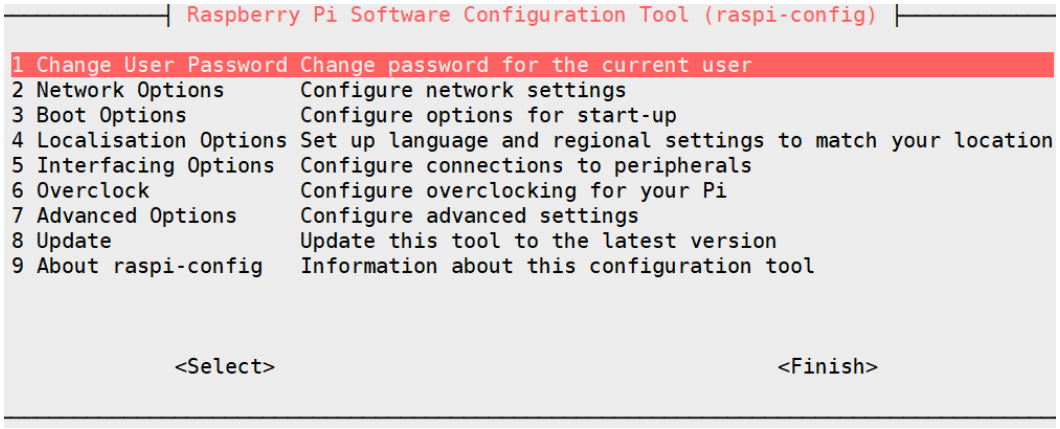

Abbildung 4.1: Raspberry Pi Software Configuration Tool

Abbildung 4.1 zeigt das Hauptmenü der Einstellungen in dem folgende Einstellungen vorgenommen werden:

- Localisation Options → Change Timezone → Europe → Berlin
- Localisation Options → Change Keyboard Layout → Generic 105-key (Intl) PC → Other → German → German → The default for the keyboard layout → No compose key
- Localisation Options → Change Wi-fi Country → DE Germany
- Interfacing Options → SSH → Yes
- Change User Password → Die Auswahl eines Passworts wird dem Nutzer überlassen
- Advanced Options → Expand Filesystem

Optional:

- Network Options → Hostname → Aussagekräftiger Name für den Raspberry Pi

Steht kein Netzwerkkabel zur Verfügung um den RPi mit einem Internet-Router zu verbinden, kann der RPi auch wie folgt die Verbindung mit einem WLAN herstellen:

- Network Options → Wi-fi → *SSID* → *Passwort*

Die Befehle `sudo apt-get update` und `sudo apt-get upgrade` installieren die neusten Updates für existierende Pakete und schließen damit dieses Grundsetup ab.

4.2 USB WLAN-Adapter Setup

Raspbian Stretch liefert Treiber für beide WLAN-Adapter bereits mit, allerdings ist der mitgelieferte Treiber des TL-WN722N-Adapters nicht mit *hostapd* kompatibel. Somit ist es notwendig einen anderen Treiber zu installieren. Der benötigte Treiber hat den Namen *8188eu* und ist über das offizielle Raspberry Pi Forum, beziehungsweise über die Webseite von Fars Robotics erhältlich[15, 16].

Es existiert zudem ein Installationsskript, welches den passenden Treiber direkt herunterlädt und installiert. Die folgenden drei Befehle laden diese Skript herunter und starten es:

```
sudo wget http://downloads.fars-robotics.net/wifi-drivers/install-wifi
  -O /usr/bin/install-wifi
sudo chmod +x /usr/bin/install-wifi
sudo /usr/bin/install-wifi
```

Der interne WLAN-Adapter des RPi Model 3 B findet hier keine Verwendung. Um ihn zu deaktiviert wird die Datei */etc/modprobe.d/raspi-blacklist.conf* geöffnet und zwei neue Zeilen eingetragen. Diese Zeilen setzen die Treiber dieses Adapters auf die Blacklist.

```
blacklist brcmfmac
blacklist brcmutil
```

4.3 Aufsetzen eines Access-Points

Dieses Kapitel beschreibt die Installation und Konfiguration eines Access-Points unter Raspbian Stretch. Die verwendete Software ist hierbei *hostapd* für den Access-Point und *dnsmasq* als DNS- und DHCP-Server. Zudem sind *dhcpcd* als DHCP-Client und *wpa_supplicant* zum Verbinden zu WLAN-Netzwerken in Verwendung. Diese beiden sind bereits Teil der Standardinstallation von Raspbian und müssen nicht installiert werden. Der folgende Befehl installiert die benötigten Pakete.

```
sudo apt-get install dnsmasq hostapd -y
```

Als nächstes werden die Dienste gestopt.

```
sudo systemctl stop dnsmasq
sudo systemctl stop hostapd
```

Die Konfigurationsdatei für *dnsmasq* ist */etc/dnsmasq.conf*. Die nachfolgenden Zeilen sind in die Datei einzufügen:

Festlegen der zu konfigurierenden Schnittstelle:

```
interface=wlan0
```

Einstellen des DHCP-Adressbereichs, sowie der Netzwerkmaske und Lease-Zeit:

```
dhcp-range=192.168.1.2,192.168.1.20,255.255.255.0,24h
```

Die Einstellung der Standard-Route:

```
dhcp-option=3,192.168.1.1
```

Drei neue Zeilen am Ende der Datei */etc/dhcpcd.conf* konfigurieren *dhcpcd*:

```
interface wlan0
static ip_address=192.168.1.1/24
nohook wpa_supplicant
```

Die erste Zeile legt die verwendete Schnittstelle fest. Die zweite Zeile definiert die IP-Adresse und Netzwerkmaske der Schnittstelle. Um zu verhindern, dass sich die Schnittstelle mit einem existierenden WLAN zu verbinden versucht, muss in der dritten Zeile noch *wpa_supplicant* für diese Schnittstelle deaktiviert werden. Die nun folgende Konfiguration von *hostapd* ist, aufgrund der Vielzahl von Einstellungsmöglichkeiten, im nächsten Unterkapitel 4.3.1 zu finden.

Im Anschluss an diese Konfiguration ist es wichtig die gerade installierten Dienste neuzustarten.
`sudo systemctl reload dnsmasq` startet dnsmasq neu.
Um hostapd starten zu können muss der Dienst erst demaskiert und aktiviert werden.
`sudo systemctl unmask hostapd`
`sudo systemctl enable hostapd`
`sudo systemctl start hostapd` kann nun den Dienst starten.

4.3.1 Hostapd Konfiguration

Zuerst ist der Speicherort der Konfigurationsdatei zu definieren. Dafür wird in der Datei */etc/default/hostapd* die Zeile `#DAEMON_CONF` durch `DAEMON_CONF="/etc/hostapd/hostapd.conf"` ersetzt.

Die Einstellungen für *hostapd* sind in der Datei */etc/hostapd/hostapd.conf* zu finden. Tabelle 4.1 bietet eine Auflistung und Erläuterung der verwendeten Einstellungen.

Tabelle 4.1: Übersicht über verwendete Optionen der hostapd.conf

Einstellung	**Beschreibung**
`interface=wlan0`	Auswahl der Schnittstelle für den Access-Point
`driver=nl80211`	Auswahl des Treibers
`ssid=ssidname`	Name der SSID des Access Points
`ieee80211d=1`	Aktivieren der Limitierung auf bestimmte Kanäle und Beschränkung der Sendeleistung basierend auf den im *country_code* festgelegten Werte des jeweiligen Landes
`country_code=DE`	Einstellung des Ländercodes
`hw_mode=g`	Festlegen des Operationsmodus auf IEEE 802.11g. g ist dabei notwendige Option um IEEE 802.11n zu nutzen. a ist die notwendige Option für IEEE 802.11ac
`ieee80211n=1`	Aktivieren von IEEE 802.11n
`ieee80211ac=1`	Aktivieren von IEEE 802.11ac
`require_vht=1`	Aktivieren der Unterstützung von Very High Throughput(VHT)
`channel=7`	Auswahl des verwendeten Kanals
`macaddr_acl=0`	MAC-Adressen-Filter wird deaktiviert solang sich die anfragende MAC-Adresse nicht auf der Deny-Liste befindet
`auth_algs=1`	Aktivierung von Open System Authentication
`ignore_broadcast_ssid=0`	SSID wird auf sichtbar gestellt
`wpa=2`	Aktivieren von WPA2-Verschlüsselung
`wpa_key_mgmt=WPA-PSK`	
`wpa_pairwise=TKIP`	
`rsn_pairwise=CCMP`	
`wpa_passphrase=passwort`	Einstellung des Passworts für WPA
`wmm_enabled=1`	Aktivierung von WMM als notwendige Voraussetzung für volle HT Funktionalität
`ht_capab=[SHORT-GI-40] [HT40-][DSSS_CCK-40]`	HT-Einstellungen des Access-Points. Für Kanal 1 bis 9 kann [HT40+] verwendet werden. Sollte der gewählte Kanal zwischen 5 und 13 liegen ist [HT40-] zu wählen. Dabei ist zu erwähnen, dass hostapd eine 40 Mhz Kanalbreite zwar unterstützt, aber selbstständig deaktiviert wenn andere Funknetzwerke diese Kanäle bereits belegen

Es folgen die verwendeten Konfigurationen für sowohl einen 2,4- als auch einen 5Ghz-Access-Point:

```
interface=SCHNITTSTELLE
driver=nl80211
ssid=SSID
ieee80211d=1
country_code=DE
hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GI-40][HT40+][DSSS_CCK-40]
channel=7
wmm_enabled=1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=PASSWORT
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

hostapd.conf für 2,4Ghz Access-Point

```
interface=SCHNITTSTELLE
driver=nl80211
ssid=SSID
ieee80211d=1
country_code=DE
hw_mode=a
ieee80211n=1
ieee80211ac=1
channel=36
require_vht=1
wmm_enabled=1
macaddr_acl=0
auth_algs=1
wpa=2
wpa_passphrase=PASSWORT
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
```

hostapd.conf für 5Ghz Access-Point

4.4 Basistests

Um einen Ausgangswert für beide WLAN-Adapter zu erhalten, sollen Basistests durchgeführt werden. Diese Tests sind dafür gedacht, um die Leistung dieser Adapter auf unterschiedlichen Entfernungen zu messen und zu ermitteln welcher Adapter besser geeignet ist.

4.4.1 Gemessene Werte

Die gemessenen Werte sind Bandbreite, verlorene Pakete und Packet Delay Variation(PDV). PDV ist die Differenz zwischen der Dauer die mehrere Pakete brauchen um ihr Ziel zu erreichen[10]. Braucht beispielsweise Paket 1 100 ms um das Ziel zu erreichen und Paket 2 120 ms ist die PDV 20 ms. PDV kann ein wichtiger Faktor bei Echtzeitanwendungen wie Voice-over-IP sein. Die Grenzwerte für PDV sind je nach Quelle unterschiedlich definiert. Die International Telecommunication Union(ITU) definiert einen Grenzwert von 50 ms für ihre höchsten Quality-of-Service Klassen[11], während Cisco einen Grenzwert von 30 ms festlegt[5].

4.4.2 Installation von iperf3

Die verwendete Testsoftware ist iperf3. Die neuste Version von iperf3 kann nur manuell installiert werden und ist auf der offiziellen Webseite erhältlich[8]. Die für Raspbian und den RPi 3 B V.2 kompatible Version, ist die Version für Ubuntu ARMhf beziehungsweise Debian ARMhf. Die Befehle um die dafür benötigten Pakete herunterzuladen und zu installieren sind:

```
1 wget https://iperf.fr/download/ubuntu/libiperf0_3.1.3-1_armhf.deb
2 wget https://iperf.fr/download/ubuntu/iperf3_3.1.3-1_armhf.deb
3 sudo dpkg -i libiperf0_3.1.3-1_armhf.deb iperf3_3.1.3-1_armhf.deb
```

4.4.3 Testaufbau

Es wird die Verbindung zwischen zwei RPi 3 B V.2 gemessen. Der erste RPi agiert dabei als Access-Point, zu dem sich der zweite RPi als Client verbindet. Beide Adapter durchlaufen zwei Testreihen. Bei der Ersten befinden sich beide Geräte auf dem Boden. Die Zweite Testreihe soll die Verbindung im Optimalfall testen bei der zwischen den Antennen beider Geräte eine offene Luftlinie besteht. Da eine möglichst ebene und potentiell sehr lange Strecke benötigt wird, sowie um den Einfluss von anderen Funknetzwerken zu minimieren, wurde als Ort für den Test ein Feldweg in der Nähe von Bad Vilbel ausgewählt. Die Abstände zwischen Tests

beträgt bei der ersten Testreihe 5 Meter. In der zweiten Testreihe befinden sich beide 150cm über dem Boden. Die RPi sind dabei an der Spitze eines Stabes befestigt der in den Boden gesteckt wird (Siehe Abbildung 4.2).

Abbildung 4.2: Raspberry Pi auf einem Stab

Die Abstände zwischen Tests betragen hierbei 25 Meter. Ursprünglich war vorgesehen die Test hier ebenfalls in Abständen von 5 Meter durchzuführen, allerdings stellte sich nach 25 Metern heraus, dass sich die Verbindungsqualität nicht erkennbar verschlechtert hatte. Somit wurden deutlich höhere Abstände von 25 Metern gewählt.

Auf dem Client-RPi läuft iperf3 als Dienst. Mit diesem verbindet sich der Access-Point-RPi. Ein Computer ist über SSH mit dem AP-RPi verbunden um den Test zu starten. Es finden sowohl TCP- als auch UDP-Tests statt. Jeder Test läuft über einen Zeitraum von jeweils 30 Sekunden. Für UDP-Tests ist die Bandbreite auf 10 Gigabit/s limitiert um die Verbindung über diese Einstellung nicht auszubremsen. Die Messung liefert Ergebnisse über die Bandbreite bei TCP und UDP, sowie die verlorenen Pakete und PDV bei UDP. Die genauen für die Tests verwendeten Befehle sind dabei
`iperf3 -c Ziel-IP -V -t 30 --logfile Dateiname`
für den TCP-Test und
`iperf3 -c Ziel-IP -V -t 30 -u -b 10G --logfile Dateiname`
für den UDP-Test. Der Befehl `iperf3 -s -D` startet einen iperf3-Server als Daemon. Die verwendeten Parameter sind in Tabelle 4.2 zu finden.

Tabelle 4.2: Verwendete iperf3 Startparameter

Parameter	**Beschreibung**
`-c Ziel-IP`	Start von iperf3 als Client verbunden zu Ziel-IP
`-V`	Ausführlichere Ausgabe
`-t 30`	Testzeitraum in Sekunden
`-u`	UDP-Test
`-b 10G`	Bandbreite für UDP-Tests in bit/s. Hier steht 10G für 10Gigabit/s.
`--logfile Dateiname`	Speichern von Testergebnissen in einer Datei
`-s`	Start von iperf3 als Server
`-D`	Start von iperf3 als Daemon

4.4.4 Ergebnisse

Dieses Unterkapitel bietet eine Zusammenfassung der Ergebnisse der oben beschriebenen Tests. Zudem enthält es einen Vergleich der beiden verwendeten USB-WLAN-Adapter.

Ergebnisse bei freier Luftlinie

Die Ergebnisse bei den Tests bei einer freien Luftlinie zeigen, dass, bei einer Entfernung von weniger als 50 Metern, der Adapter von Edimax eine höhere Bandbreite bietet. Bei jeglichen höherer Entfernung erzielt der Adapter von TP-Link bessere Ergebnisse. Dies ist sowohl in den TCP- als auch in den UDP-Tests erkennbar (Siehe Tabelle 4.3). Beide Adapter zeigen, dass bei einer bestimmten Entfernung die Bandbreite deutlich sinkt. Bei dem Adapter von Edimax sind dies die erwähnten 50 Meter. Bei dem Adapter von TP-Link beträgt die Entfernung 375 Meter.

Tabelle 4.3: Basistests: Bandbreite Luftlinie

Bandbreite TCP (Mbit/s)

Entfernung in m	TP-Link	Edimax
25	80,7	122
50	78,3	73
75	81,2	82,3
100	81,5	69,7
125	79,7	56
150	78,7	56
175	76,7	50
200	76,9	46,9
225	76,1	52,1
250	80,9	45,4
275	80,9	50,9
300	64,7	36,5
325	74,5	35
350	69,5	28,5
375	25,5	16,3
400	15,8	12,2
425	10,1	4,39
450	13,8	
475	17,9	
500	18,1	
525	17,1	
550	13	
575	11,7	
600	9,12	

Bandbreite UDP (Mbit/s)

Entfernung in m	TP-Link	Edimax
25	93,8	127
50	94,3	93,4
75	96,4	90,7
100	94,7	82
125	91,8	62,1
150	89,2	62,7
175	90,7	61,7
200	90,3	55,2
225	87,4	56,8
250	94,1	50,9
275	93,4	56,9
300	94,5	36,8
325	79,7	34,3
350	66,1	30,8
375	32,1	18,8
400	20,5	14
425	10,4	2,76
450	14,1	
475	17,5	
500	21,3	
525	20	
550	14	
575	14,1	
600	7,94	

Die Bandbreite bei den UDP-Tests ist dabei bei beiden Adaptern, abgesehen von jeweils zwei Entfernungen, leicht höher als bei den TCP-Tests. Im Schnitt sind dies ≈15,9% bei TP-Link und ≈11,9% bei Edimax. Erwähnenswert ist, dass jeweils einer der zwei schlechteren Werte beider Adaptern auf der höchsten gemessenen Entfernung liegt. Es zeigt sich bei dem Adapter von Edimax, nach dem starken Abfall bei 50 Metern, eine sehr gleichmäßige Senkung der Bandbreite bis zu einer Entfernung von 425 Metern (Siehe Abbildung 4.3). Während der Adapter von TP-Link bis 350 Meter eine fast gleichbleibende Bandbreite, zwischen 80 und 70 Mbit/s, liefert ist der Abfall bei 375 Metern auf 25,5 Mbit/s sehr deutlich. Dennoch liegt er selbst dort noch mehr als 50% vor dem Adapter von Edimax.

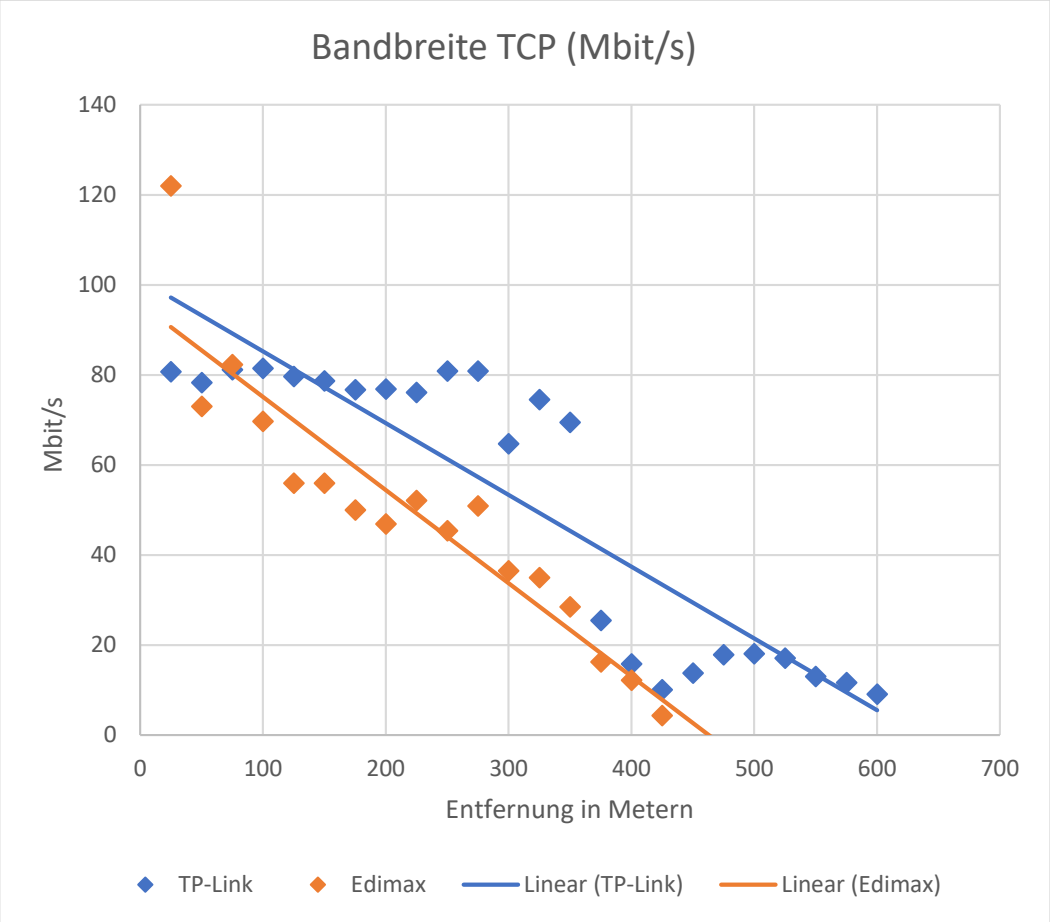

Abbildung 4.3: Basistests: Bandbreite TCP (Mbits/s) Graph Luftlinie

Bei den verlorenen Paketen wurde, ausgenommen von drei Tests, 0% gemessen. Der höchste Wert war dabei der 300 Meter Test des TP-Link mit einem Wert von 0,018% verlorenen Paketen. Bei der gemessenen Packet Delay Variation (PDV) ist ein sehr ähnliches Bild zu erkennen wie bei der Bandbreite (Siehe Tabelle 4.4). Bei den gleichen Entfernungen, bei denen die Bandbreite sinkt, steigt die PDV.

Tabelle 4.4: Basistests: PDV in ms (UDP)) Luftlinie

Entfernung in m	TP-Link	Edimax
25	0,651	0,43
50	0,757	1,603
75	0,525	0,533
100	0,57	0,684
125	0,81	1,538
150	0,849	0,995
175	0,639	1,208
200	0,897	1,09
225	0,834	1,494
250	0,572	1,405
275	0,611	1,682
300	1,261	2,176
325	1,088	2,806
350	0,724	2,352
375	2,061	3,986
400	5,885	4,784
425	5,597	12,254
450	4,943	
475	6,861	
500	3,316	
525	4,093	
550	4,774	
575	7,886	
600	9,403	

Auch hier ist ein fast gleichmäßiger Anstieg bei dem Adapter von Edimax und ein deutlicher Anstieg bei dem Adapter von TP-Link zwischen 350 und 400 Metern zu erkennen (Siehe Abbildung 4.4). Dort steigt der Wert zuerst von 0,724 ms auf 2,061 ms und dann auf 5,885 ms.

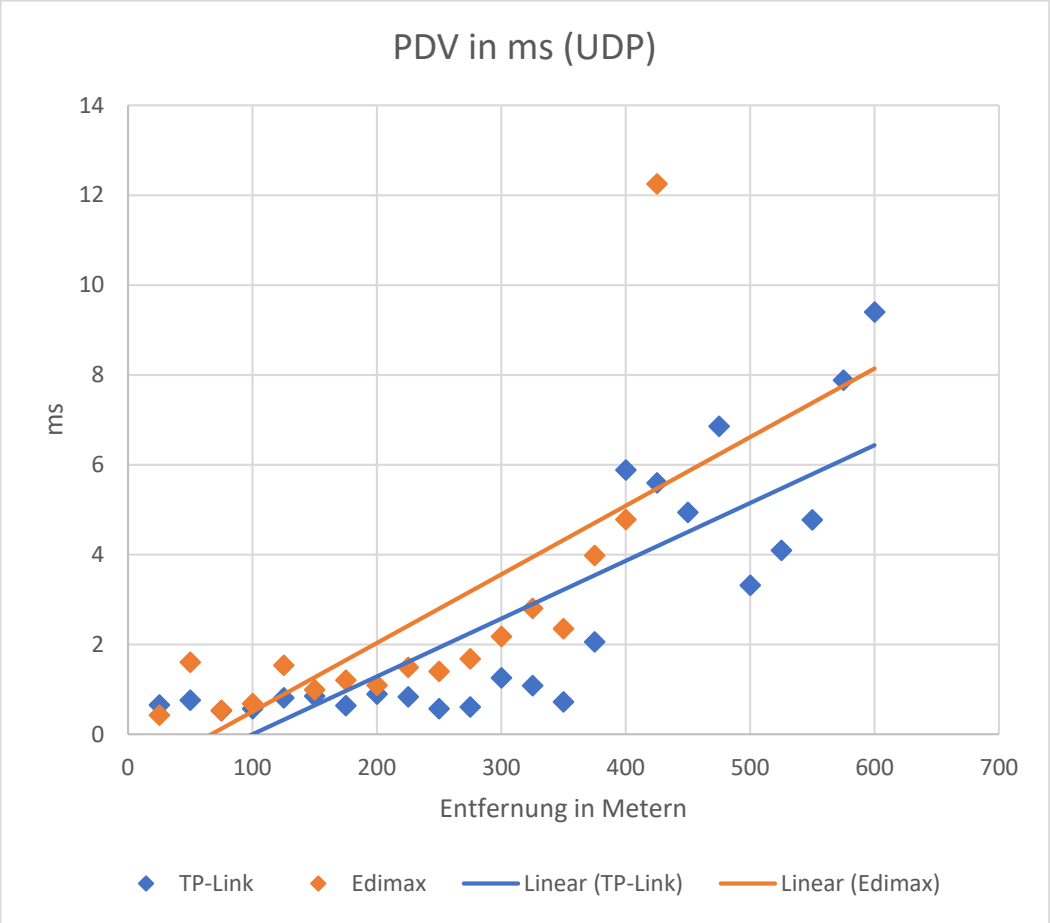

Abbildung 4.4: Basistests: PDV in ms (UDP) Graph Luftlinie

Ergebnisse auf dem Boden

Die Messung der Bandbreite ergibt, dass bei jeder Entfernung der TP-Link Adapter über dem Edimax Adapter liegt. Dies gilt sowohl für TCP- als auch für UDP-Tests (Siehe Tabelle 4.5). Es ist zu erkennen, dass die Bandbreite teilweise bei höheren Entfernungen besser ist als bei Kürzeren. Dies ist zum Beispiel bei den 40 Meter Tests beider Adapter, im Vergleich zu ihren 35 Meter Test, zu erkennen. Die UDP-Tests des TP-Link-Adapters zeigen eine ≈14,7% höhere Bandbreite als die TCP-Tests. Für den Adapter von Edimax ist hier ein anderes Bild erkennbar. Bei sechs Entfernungen sind die Werte des UDP-Tests höher und bei vier Entfernungen niedriger. Im Schnitt ist dies nur eine ≈0,2% höhere Bandbreite die der TCP-Tests.

Tabelle 4.5: Basistests: Bandbreite Boden

Bandbreite TCP (Mbit/s)

Entfernung in m	TP-Link	Edimax
5	82,2	66
10	81,9	52,8
15	79	44,1
20	78,4	40,3
25	48,3	32,8
30	39,1	14,9
35	25,2	15,3
40	27,9	19,5
45	14,6	9,8
50	19,7	9,31
55	7,1	2,51
60	10,1	
65	9,85	
70	7,13	
75	3,34	
80	3,85	

Bandbreite UDP (Mbit/s)

Entfernung in m	TP-Link	Edimax
5	95,7	68,4
10	95,7	64,4
15	93,1	38,3
20	85,3	32
25	54,1	22,3
30	40,3	23,2
35	24,1	18,3
40	31,3	17,9
45	16,6	10,5
50	22,8	12,5
55	10,2	0
60	15,9	
65	9,39	
70	13,6	
75	5,44	
80	3,33	

Die Bandbreite des TP-Link Adapter bleibt bis 20 Meter sehr stabil bei circa 80 Mbit/s und sinkt dann bei 25 Metern deutlich auf 48,3 Mbit/s (Siehe Abbildung 4.5).

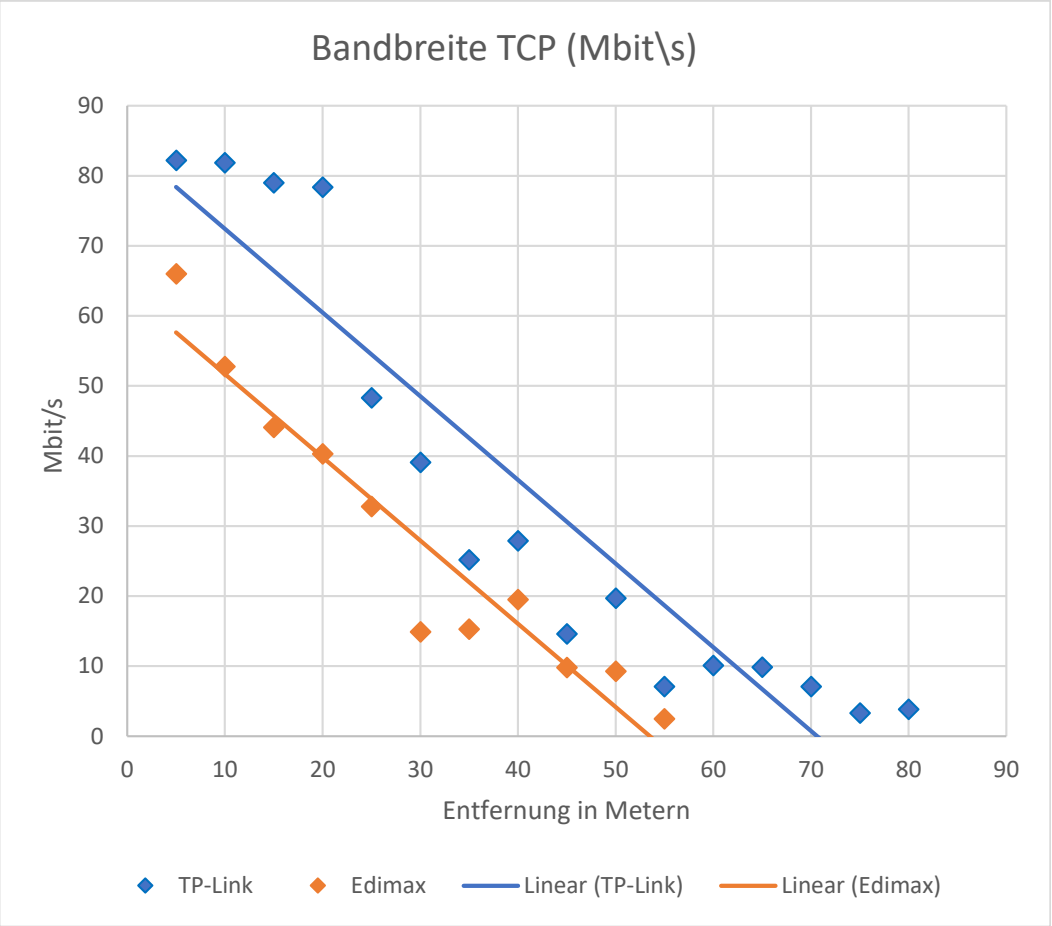

Abbildung 4.5: Basistests: Bandbreite TCP (Mbits/s) Graph Boden

Die verlorenen Pakete liegen, ausgenommen von drei Tests, bei 0%. Der höchste Wert war dabei der 80 Meter Test des TP-Link, mit einem Wert von 0,13% verlorenen Paketen.

Für den TP-Link Adapter beginnt die PDV bei 0,534 ms auf 5 Metern und steigt bis zu einem Maximum von 14,13 ms auf einer Entfernung von 75 Metern (Siehe Tabelle 4.6). Bei dem Edimax Adapter gibt es drei Werte die deutlich höher als der Rest sind. Einer dieser Werte liegt mit 15,834 ms bereits auf der kürzesten gemessenen Entfernung. Wie auch bei der Bandbreite erkennbar, liegen die gemessenen Werte für PDV teilweise bei kürzeren Entfernungen über denen bei höheren Entfernungen.

Tabelle 4.6: Basistests: PDV in ms (UDP) Boden

Entfernung in m	TP-Link	Edimax
5	0,534	15,834
10	0,76	1,858
15	0,606	1,845
20	0,764	1,398
25	1,019	3,681
30	1,204	12,827
35	2,696	3,294
40	3,14	6,783
45	2,124	7,597
50	6,882	25,663
55	8,1	0
60	4,753	
65	12,176	
70	6,371	
75	14,13	
80	11,206	

Trotz nicht immer steigender Werte ist an der Trendlinie der PDV zu erkennen, dass der Wert des TP-Link-Adapters mit zunehmender Entfernung steigt (Siehe Abbildung 4.6). Auch die Trendlinie des Edimax-Adapters zeigt, trotz drei sehr hoher Werte an unterschiedliche Stellen, immer noch eine eindeutige Steigung.

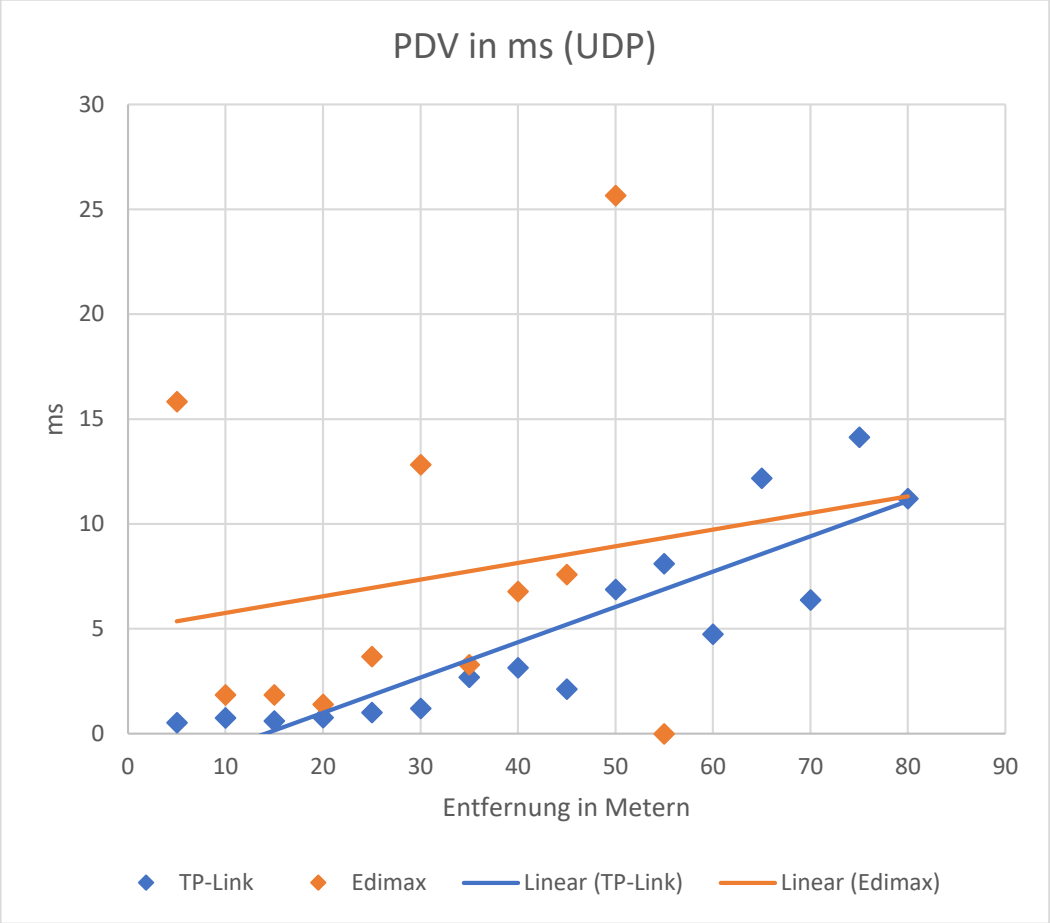

Abbildung 4.6: Basistests: PDV in ms (UDP) Graph Boden

Bewertung

Vergleicht man die Ergebnisse der Tests mit freier Luftlinie und der Tests auf dem Boden kann man einige Schlüsse ziehen. Erstens ist der Adapter von TP-Link bei nahezu allen Tests dem Adapter von Edimax überlegen. Nur unter 50 Metern auf freier Luftlinie fallen die Ergebnisse des Edimax-Adapters besser aus. Außerdem erreicht der TP-Link-Adapter eine Reichweite von 600 Metern und der Edimax-Adapter nur eine von 425 Meter. Da beides wichtige Kriterien für diese Arbeit sind, werden im Folgenden nur die Ergebnisse des TP-Link-Adapters weiterbewertet. Im Vergleich zu den Luftlinien-Tests sinkt die Reichweite deutlich wenn sich die Antennen auf dem Boden befinden. Die weiteste gemessene Reichweite liegt bei

freier Luftlinie auf 600 und auf dem Boden bei 80 Metern. Die UDP-Ergebnisse sind bei den Luftlinie-Tests um ≈15,9% und bei den Boden-Tests um ≈14,7% höher als die TCP-Ergebnisse. Sowohl bei den Tests über Luftlinie als auch am Boden bleibt die Bandbreite anfangs sehr stabil bei ≈80 Mbit/s. Auf Luftlinie sinkt diese dann deutlich auf 25,5 Mbits/s bei 375 Metern. Bei den Boden-Tests ist das Gleiche auf 48,3 Mbit/s bei 25 Metern zu erkennen. Sind die Antennen auf dem Boden platziert, ist zudem zu erkennen, dass die Bandbreite zum Teil bei kürzeren Entfernungen schlechter ist als bei längeren. Dies zeigt, dass selbst kleinste Höhenunterschiede große Auswirkung auf die Verbindungsqualität haben können. In der Praxis kann selbst eine wenige Zentimeter hohe Unebenheit oder ein Büschel Grass, auf dem sich die Antenne befindet, nennenswerte Auswirkungen haben. Sowohl die deutlich besseren Werte bei offener Luftlinie, als auch die Schwankungen wenn sich die Antennen auf dem Boden befinden, zeigen dass, zusammen mit der Entfernung, die Platzierung der Antenne einen großen Einfluss auf die Bandbreite und Verbindungsqualität hat. Aufgrund der besseren Werte des TP-Link-Adapters in nahezu allen Tests wird er in allen nachfolgenden Tests ausschließlich verwendet.

5 Implementierung einer Raspberry Pi-Kette

Diese Kapitel behandelt die Implementierung einer Kette aus RPi. Die ersten zwei Unterkapitel beschreiben zum einen die Implementation mit Hilfe von 2,4Ghz-Repeatern sowie, zum anderen, mit 2,4Ghz und 5Ghz-Repeatern im Wechsel. Das dritte Unterkapitel beschreibt die Implementierung einer Kette mit Hilfe eines batman-Meshs. Die hier gezeigten Implementierungen basieren teilweise auf unterschiedlichen Anleitungen und Artikeln aus dem Internet. Die vollständige Liste ist in den Quellen zu finden [6, 7, 4, 21].

5.1 Implementierung mit 2,4Ghz Repeatern

Vorraussetzung für einen Repeater sind zwei Schnittstellen. In Rasbian sind WLAN-Schnittstellen mit wlan# benannt, wobei die Nummerierung von 0 beginnt. Die Schnittstelle *wlan0* soll als Access-Point dienen, mit dem sich das nächste Glied der Kette verbinden kann, während die Schnittstelle *wlan1* eine Verbindung zum vorherigen Access-Point in der Kette herstellt. Der RPi leitet dann die Pakete von *wlan0* an *wlan1* weiter (Siehe Abbildung 5.1).

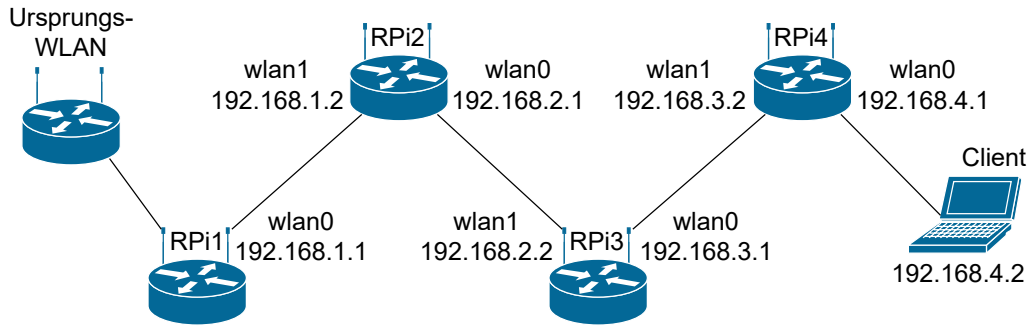

Abbildung 5.1: Netzwerkdiagramm: 2,4Ghz Repeater-Kette

Dieses Kapitel setzt die in Kapitel 4.1 bis Kapitel 4.3 beschriebene Grundkonfiguration und Access-Point Installation voraus. Um, aufbauend auf dieser Konfiguration, einen RPi als Repeater zu konfigurieren ist ein zweiter USB-WLAN-Adapter nötig. Zuerst ist es notwendig eine Verbindung mit dem WLAN herzustellen, dass erweitert werden soll. In Raspbian ist dafür *wpa_supplicant* zustän-

dig. Die Datei in der die Verbindungsdaten für WLANs gespeichert sind ist */etc/wpa_supplicant/wpa_supplicant.conf*. Ein neues WLAN kann in folgender Form der Datei hinzugefügt werden.

```
network={
    ssid="SSID"
    psk="Passwort"
}
```

Für jeden Repeater ist jeweils die SSID und Passwort des vorherigen Access-Points einzutragen. Da jeder Repeater seinen eigenen Access-Point besitzt, mit dem sich der nächste Repeater verbindet, sind für jeden Repeater eigene Werte für die IP-Adresse, Netzwerkmaske der Schnittstelle, sowie für DHCP festzulegen. Für die Schnittstelle sind diese an das Ende der Datei */etc/dhcpcd.conf* anzufügen.

```
interface wlan0
static ip_address=192.168.5.1/24
nohook wpa_supplicant
```

Die IP-Adresse unter `static ip_address` muss für jeden Repeater einzigartig sein. Es ist zu empfehlen eine fortlaufende Nummerierung zu wählen, um später den jeweiligen Repeater leichter identifizieren zu können. Entsprechend des Netzes das in der Schnittstelle konfiguriert ist, müssen die Werte des DHCP-Servers in der Datei */etc/dnsmasq.conf* angepasst werden. Ist beispielsweise `static ip_address=192.168.5.1/24` definiert, muss das Ende von */etc/dnsmasq.conf* wie folgt aussehen:

```
interface=wlan0
dhcp-range=192.168.5.2,192.168.5.20,255.255.255.0,24h
dhcp-option=3,192.168.5.1
```

Um Interferenzen zwischen den Repeatern zu minimieren sollten die Access-Points der Kette zudem nicht den gleichen Kanal nutzen. Bei einer Kanalbreite von 20 Mhz ist ein Abstand von 4 freien Kanälen notwendig. Bei einer Kanalbreite von 40 Mhz ein Abstand von 8 freien Kanälen. Hier werden abwechselnd Kanal 1 und 11 genutzt. Es ist zu beachten, dass, bei einer Änderung des Kanals, eventuell auch die HT-Einstellung geändert werden muss (Siehe Kapitel 4.3.1). Zudem muss für jeden Repeater eine eigene SSID und Passwort definiert sein. Wie auch bei der IP-Adresse der Schnittstelle, ist es empfehlenswert eine fortlaufende oder zumindest eindeutige Bezeichnung für die SSID jedes Repeaters zu wählen. In diesem Fall wird für jeden Repeater die SSID *chain#* genutzt, wobei die jeweilige Nummer in der Kette # ersetzt.

All diese Einstellungen sind in der Datei */etc/hostapd/hostapd.conf* zu finden.

```
interface=wlan0
driver=nl80211
ssid=SSID
ieee80211d=1
country_code=DE
hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GI-40][HT40+][DSSS_CCK-40]
channel=1
wmm_enabled=1
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=PASSWORT
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Die Beschreibung der jeweiligen Optionen findet sich in Kapitel 4.3.1. Der erste Repeater der Kette hätte so beispielsweise einen Access-Point mit der SSID *chain1* und der IP-Adresse *192.168.1.1/24*. Zuletzt ist die Paketweiterleitung zu konfigurieren. Dazu muss zuerst in der Datei */etc/sysctl.conf* die Zeile
`#net.ipv4.ip_forward=1` auskommentiert, beziehungsweise durch
`net.ipv4.ip_forward=1`
ersetzt werden um das Weiterleiten generell zu erlauben. Anschließend ist eine Weiterleitung zwischen *wlan0* (dem AP) und *wlan1* (Verbindung zum vorherigen Glied in der Kette) in der Firewall einzurichten. In Raspbian ist dafür *iptables* zuständig. Dies geschieht mit den Befehlen
`sudo iptables -table nat -append POSTROUTING -out-interface wlan1 -j MASQUERADE` für die POSTROUTING Kette und
`sudo iptables -append FORWARD -in-interface wlan0 -j ACCEPT`
für die FORWARD Kette.
Der Befehl `sudo sh -c "iptables-save < /etc/iptables.ipv4.nat"` speichert diese Regeln in der Datei */etc/iptables.ipv4.nat*.
Um die Regeln bei jedem Neustart des RPi automatisch wiederherzustellen, ist es notwendig *crontab* zu konfigurieren. *Crontab* ist ein Programm zur Automatisierung und wird mit `sudo crontab -e` geöffnet. Die Zeilen
`@reboot sudo iptables-restore < /etc/iptables.ipv4.nat`
`@reboot sudo iperf3 -s -D`
sind der geöffneten Datei anzufügen. Die erste Regel stellt die zuvor erstellten Regeln beim Neustart wieder her, während die zweite Regel einen iperf3 Server als Daemon startet.

5.2 Implementierung mit 2,4Ghz und 5Ghz Repeatern

Dieses Kapitel beschreibt eine Kette aus Repeatern mit sich abwechselnden 2,4- und 5Ghz Netzen (Siehe Abbildung 5.2). Dafür werden die in Kapitel 5.1 verwendeten RPi 3 B durch RPi 3 B+ ersetzt. Diese besitzen bereits ein internes 5Ghz-Funkmodul. Dieses Kapitel setzt die in Kapitel 4.1 bis Kapitel 4.3 beschriebene Grundkonfigu-

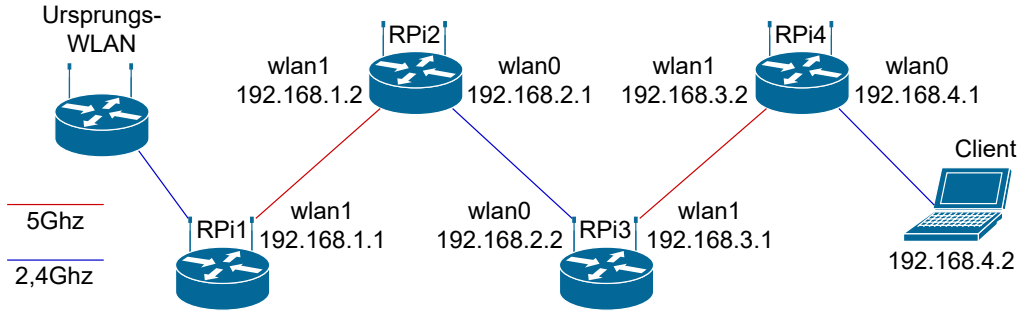

Abbildung 5.2: Netzwerkdiagramm: 2,4/5Ghz Repeater-Kette

ration und Access-Point Installation voraus. Es wird sich zu Nutzen gemacht, dass Rasbian Stretch, beim Anschluss eines externen WLAN-Adapters, diesen immer an den Anfang der WLAN-Nummerierung stellt. Somit ist der externe Adapter immer *wlan0* und das interne Modul *wlan1*. Im Vergleich zur Implementierung mit ausschließlich 2,4Ghz-WLANs gibt es einige Unterschiede. Im Speziellen gibt es zwei unterschiedliche Konfigurationen, je nachdem ob der jeweilige RPi einen 2,4- oder 5Ghz-Access-Point aufbaut. Konfiguration 1 findet Verwendung bei RPi1 und RPi3. Konfiguration 2 bei RPi2 und RPi4. Es ist zu beachten, dass sich auch diese Konfigurationen jeweils noch leicht unterscheiden. Die IP des APs jedes Gerätes muss mit der richtigen, einzigartigen Zahl versehen sein (Siehe Abbildung 5.2). Außerdem sollten sich nicht überschneidende Frequenzbereiche gewählt werden, damit sich die WLANs nicht gegenseitig stören können. Es kann der Beschreibung in Kapitel 5.1 gefolgt werden. Die Änderungen sind im Folgenden beschrieben. In den Dateien */etc/dnsmasq.conf* und */etc/dhcpcd.conf* ist für *interface* die jeweils verwendete Schnittstelle des AP einzutragen (*wlan0* bei 2,4Ghz-AP und *wlan1* bei 5Ghz-AP). Die Paketweiterleitung ist so zu konfigurieren, dass als *-out-interface* die Schnittstelle definiert ist, die dem Ursprungs-WLAN näher ist, sowie als *-in-interface* die jeweils andere Schnittstelle. Sollte, wie in Kapitel 4.2 des Grundsetups beschrieben, der interne WLAN-Adapter deaktiviert sein, sind dieses Änderungen in der Datei */etc/modprobe.d/raspi-blacklist.conf* rückgängig zu machen um ihn wieder zu aktivieren.

Schlussendlich folgt hier die Konfiguration der Datei */etc/hostapd/hostapd.conf* für 5Ghz-Access-Points.

```
interface=wlan1
driver=nl80211
ssid=SSID
ieee80211d=1
country_code=DE
hw_mode=a
ieee80211n=1
ieee80211ac=1
channel=36
require_vht=1
wmm_enabled=1
macaddr_acl=0
auth_algs=1
wpa=2
wpa_passphrase=PASSWORT
wpa_key_mgmt=WPA-PSK
rsn_pairwise=CCMP
```

Die Konfiguration für 2,4Ghz-Access-Points ist in Kapitel 5.1 zu finden. Die Beschreibung der jeweiligen Optionen findet sich in Kapitel 4.3.1.

5.3 Implementierung mit einem batman-Mesh

In diesem Kapitel wird die Erweiterung eines WLANs mit Hilfe eines batman-Meshs beschrieben. Einer der Knoten agiert dabei als Gateway, der das Mesh mit dem zu verlängernden WLAN verbindet. Ein weiterer Knoten ist zusätzlich als Access-Point konfiguriert, mit dem sich der Client verbinden kann. Die Mesh-Knoten besitzen eine Schnittstelle *wlan0* die mit einer virtuellen batman-Mesh-Schnittstelle *bat0* verknüpft ist. Sowohl Gateway-Knoten, als auch der AP-Knoten, besitzen eine weitere Schnittstelle *wlan1*. Bei dem Gateway-Knoten dient diese der Verbindung mit dem Ursprungs-WLAN. Bei dem AP-Knoten ist sie als Access-Point konfiguriert mit dem sich der Client verbindet (Siehe Abbildung 5.3).

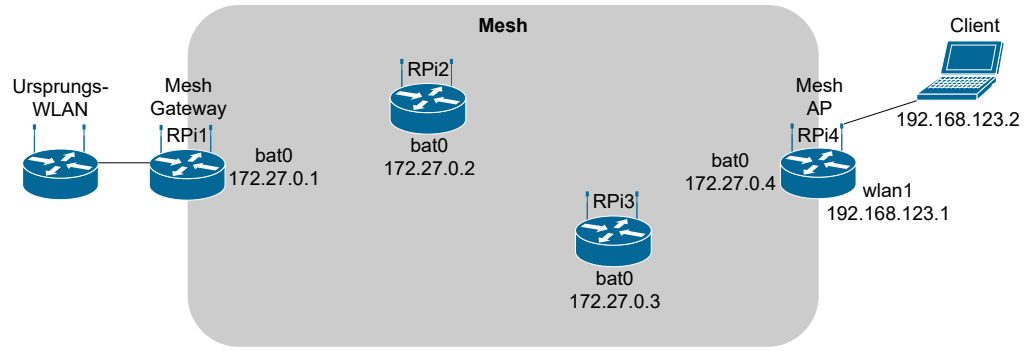

Abbildung 5.3: Netzwerkdiagramm: Mesh-Kette

Die folgende Implementierung setzt das Grundsetup von Kapitel 4.1 bis Kapitel 4.2 voraus. Es ist allerdings wichtig den Befehl *sudo apt-get upgrade* am Ende von Kapitel 4.1 nicht auszuführen, da es sonst zu Kompatibilitätsproblemen mit batman kommt. Zuerst ist es nötig die für git notwendigen Pakete zu installieren.

```
sudo apt-get install libnl-3-dev libnl-genl-3-dev git -y
```

Anschließend kann batctl mit folgenden Befehlen heruntergeladen und installiert werden.

```
git clone https://git.open-mesh.org/batctl.git
cd batctl
sudo make install
```

5.3.1 Mesh-Gateway einrichten

Vorerst ist es notwendig eine Verbindung mit dem WLAN herzustellen, dass erweitert werden soll. In Raspbian ist dafür *wpa_supplicant* zuständig. Die Datei in der die Verbindungsdaten für WLANs gespeichert sind ist */etc/wpa_supplicant/wpa_supplicant.conf*. Das Hinzufügen eines neuen WLANs geschieht in folgender Form:

```
network={
    ssid="SSID"
    psk="Passwort"
}
```

Anschließend ist es notwendig die beiden Schnittstellen *wlan0* und *wlan1* zu konfigurieren. Dafür müssen die Einstellungen der Schnittstellen in zwei Dateien vorgenommen werden. Die Datei */etc/network/interfaces* ist um die nachfolgenden Zeilen zu erweitern:

```
auto lo
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

Die zweite Datei die erweitert werden muss ist */etc/dhcpcd.conf*:

```
denyinterfaces wlan0
interface wlan0
nohook wpa_supplicant
```

Um nun batman zu konfigurieren und zu starten müssen einige Einstellungen vorgenommen und Befehle ausgeführt werden. Da dies bei jedem Neustart des RPi erneut ausgeführt werden muss, wird ein Skript namens */usr/local/bin/batmanstart.sh* dafür erstellt.

```
1  sleep 5s
2  sudo modprobe batman-adv # batman-adv aktivieren
3  sudo ip link set wlan0 down # wlan0 deaktivieren
4  sudo ifconfig wlan0 mtu 1532
5  sudo iwconfig wlan0 mode ad-hoc
6  sudo iwconfig wlan0 essid mesh-ssid
7  sudo iwconfig wlan0 ap any
8  sudo iwconfig wlan0 channel 1
9  sleep 1s
10 sudo ip link set wlan0 up # wlan0 wieder aktivieren
11 sleep 1s
12 sudo batctl if add wlan0 # wlan0 als batman Schnittstelle hinzufuegen
13 sleep 1s
14 sudo ifconfig bat0 up # bat0 Schnittstelle aktivieren
15 sleep 5s
16 sudo ifconfig bat0 172.27.0.1/16 # Jeder RPi braucht eine eigene IP-
       Adresse fuer die bat0 Schnittstelle
17 sleep 1s
18 # Paketweiterleitung aktivieren und einstellen fuer die Weiterleitung
       zwischen bat0 und wlan1
19 sudo sysctl net.ipv4.ip_forward=1
20 sudo iptables -t nat -A POSTROUTING -o wlan1 -j MASQUERADE
21 sudo iptables -A FORWARD -i wlan1 -o bat0 -m state --state RELATED,
       ESTABLISHED -j ACCEPT
22 sudo iptables -A FORWARD -i bat0 -o wlan1 -j ACCEPT
```

Um Pakete durch das Mesh zu transportieren stellt batman allen Paketen seinen eigenen Header vorran. Daher wird empfohlen die MTU auf mindestens 1528 Byte zu erhöhen, um zu verhindern, dass größere Pakete fragmentiert werden (Zeile 4)[26]. Bei der hier verwendeten Version von Raspbian ist es allerdings nicht möglich die MTU auf mehr als 1500 zu stellen. Nur ältere Versionen, wie zum Beispiel Jessie, die nicht auf Raspbian Stretch basieren bieten diese Option. Der Modus der Schnittstelle ist auf *ad-hoc* zu setzen (Zeile 5). Die SSID des Mesh muss auf allen Knoten gleich sein (Zeile 6). Der Access-Point, zu dem sich verbunden werden soll, kann auf *any* gestellt sein (Zeile 7). Der Kanal der Schnittstelle muss auf 1 gestellt sein, da sonst batman selbstständig versucht die Einstellung auf Kanal 1 zu ändern (Zeile 8).
Um das Skript bei jedem Neustart des RPi automatisch zu starten ist es notwendig *crontab* zu konfigurieren. *Crontab* ist ein Programm zur Automatisierung und wird mit `sudo crontab -e` geöffnet. Die Zeilen

```
@reboot sudo bash /usr/local/bin/batmanstart.sh
@reboot sudo iperf3 -s -D
```

sind der geöffneten Datei anzufügen. Die erste Regel startet das zuvor erstellte Script beim Neustart und die zweite Regel startet einen iperf3 Server als Daemon.

5.3.2 Mesh-Knoten einrichten

Die Knoten sind sehr ähnlich eingerichtet wie das Gateway. Deshalb sind hier nur die Unterschiede zu Kapitel 5.3.1 aufgezeigt. *Wpa_supplicant* wird nicht konfiguriert. Die Dateien */etc/network/interfaces*, */etc/dhcpcd.conf* und */usr/local/bin/batmanstart.sh* sind anders zu konfigurieren. Es kann dabei der Anleitung in Kapitel 5.3.1 mit den nachfolgenden Änderungen in den verwendeten Dateien gefolgt werden.

```
auto lo
iface lo inet loopback

iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
```

/etc/network/interfaces

```
interface wlan1
static ip_address=192.168.123.1/24
nohook wpa_supplicant
```

/etc/dhcpcd.conf

```
sleep 5s
sudo modprobe batman-adv # batman-adv aktivieren
sudo ip link set wlan0 down # wlan0 deaktivieren
sudo ifconfig wlan0 mtu 1532
sudo iwconfig wlan0 mode ad-hoc
sudo iwconfig wlan0 essid mesh-ssid
sudo iwconfig wlan0 ap any
sudo iwconfig wlan0 channel 1
sleep 1s
sudo ip link set wlan0 up # wlan0 wieder aktivieren
sleep 1s
sudo batctl if add wlan0 # wlan0 als batman Schnittstelle hinzufuegen
sleep 1s
sudo ifconfig bat0 up # bat0 Schnittstelle aktivieren
sleep 5s
sudo ifconfig bat0 172.27.0.1/16 # Jeder RPi braucht eine eigene IP-
    Adresse fuer die bat0 Schnittstelle
sleep 1s
```

```
18 # Paketweiterleitung aktivieren und einstellen fuer die Weiterleitung
       zwischen bat0 und wlan1
19 sudo iptables -t nat -A POSTROUTING -o bat0 -j MASQUERADE
20 sudo iptables -A FORWARD -i bat0 -o wlan1 -m state --state RELATED,
       ESTABLISHED -j ACCEPT
21 sudo iptables -A FORWARD -i wlan1 -o bat0 -j ACCEPT
22 sleep 2s
23 sudo ip route add default via 172.27.0.1
```

/usr/local/bin/batmanstart.sh

Zusätzlich zu der Paketweiterleitung ist eine Default-Route zur IP-Adresse der *bat0* Schnittstelle des Mesh-Gateways einzurichten. Dies geschieht in der letzten Zeile des *batmanstart.sh* Skripts.

Access-Point für Mesh-Knoten einrichten

Siehe Kapitel 4.3 um den Access-Point für einen Mesh-Knoten einzurichten. Zu verändern ist die Schnittstelle von *wlan0* zu *wlan1*. Um auch eine Verbindung zum Internet herstellen zu können, ist zusätzlich ein DNS-Nameserver einzutragen. Dies geschieht in der Datei */etc/resolvconf.conf* mit der folgenden Zeile und einem beliebigen Nameserver.

`name_servers=8.8.8.8`

6 Netzwerktests der Implementierung

Das folgende Kapitel behandelt die Tests der drei Implementierungen der RPi-Kette. Es ist dabei aufgeteilt zwischen Tests auf offenem Gelände und im Wald. Zu jedem Test werden zunächst der Testaufbau geschildert und anschließend die Ergebnisse ausgewertet. Auf Grund der deutlich schlechteren Ergebnisse während den Bodentest in Kapitel 4.4.4 gibt es hier nur noch Tests mit RPi in erhöhter Position.

6.1 Kette auf offenem Gelände

Zuerst gilt es Tests auf offenem Gelände durchzuführen um die höchstmöglichen Werte festzustellen die erzielt werden können. Zudem erlauben diese Werte es eine Kette mit einer einzelnen direkten Verbindung zu vergleichen.

6.1.1 Testaufbau

Der Testaufbau für die Repeater-Kette lautet wie folgt. Vier Rpi sind als Repeater eingerichtet (Siehe Kapitel 5.1 und 5.2). Mit dem letzten Repeater der Kette ist ein Notebook verbunden, von dem aus die Tests durchgeführt werden (Siehe Abbildung 6.1).

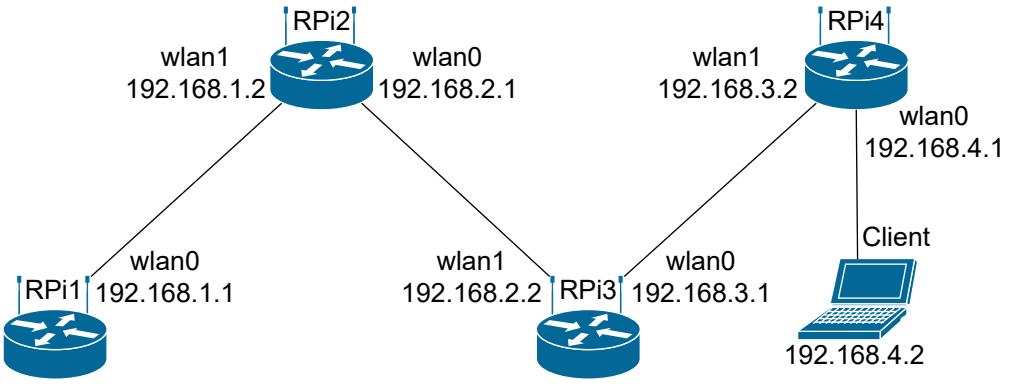

Abbildung 6.1: Netzwerkdiagramm: Repeater-Kette Testaufbau

Der Testaufbau der Mesh-Kette ist gleich zu Abbildung 5.3 in Kapitel 5.3. Die Tests finden auf einem Feldweg in der Nähe von Bad Vilbel statt um den Einfluss von anderen WLANs zu minimieren (Siehe rot markierte Strecke in Abbildung 6.2). Jeder

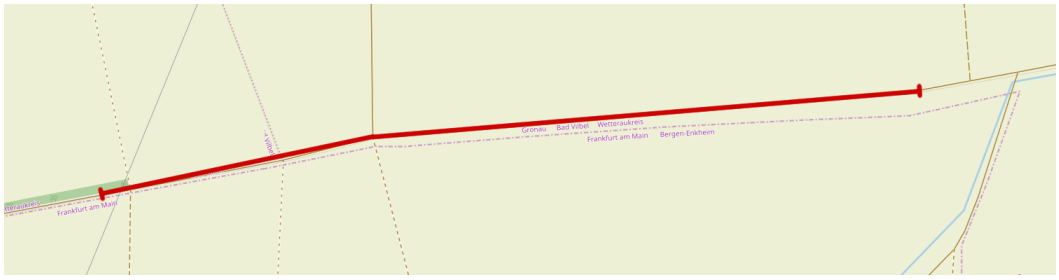

Abbildung 6.2: Karte des Ortes der Tests

RPI ist auf einem 150cm langen Stab befestigt der in den Boden gesteckt ist. Die Entfernung zwischen den RPi während der Tests liegt bei jeweils n. Die Werte für n sind 50, 100, 150, 200 und 300 Meter. Für die 2,4/5Ghz-Tests finden die internen WLAN-Adapter und damit auch die internen Antennen für 5Ghz Verwendung. Da diese Antennen eine größere Richtwirkung als die externen Antennen der USB-WLAN-Adapter haben, wird darauf geachtet, dass die Seite der Platine, an der sich die Antenne befindet, in Richtung des vorherigen 5Ghz-Senders, beziehungsweise des nächsten Empfängers, zeigt. Die Messung findet zwischen dem Client und RPi1 statt. Der Client für diese Tests ist ein Microsoft Surface Pro 3. Auf Grund von Problemen bei den Tests mit 2,4- und 5Ghz Repeatern, konnte dort nicht der interne WLAN-Adapter des Surface Pro 3 genutzt werden. Stattdessen kommt bei diesen Tests ein TP-Link WLAN-Adapter zum Einsatz. Dieser hat eine höhere Übertragungsrate als der interne Adapter, was in die Bewertung der Ergebnisse mit einfließt. Für jede Entfernung finden sechs Tests statt:

- Einzelne Verbindung (TCP)
- Zwei gleichzeitige Verbindungen (TCP)
- Fünf gleichzeitige Verbindungen (TCP)
- Einzelne Verbindung (UDP)
- Zwei gleichzeitige Verbindungen (UDP)
- Fünf gleichzeitige Verbindungen (UDP)

Die 100-Meter-Tests beinhalten zudem zusätzliche Tests zu den anderen RPi, um herauszufinden wie stark jeder einzelne Hop die Verbindungsqualität beeinflusst. Die verwendete Testsoftware ist iperf3. Dessen Installationsanleitung findet sich in Kapitel 4.4.2. Die betrachteten Werte sind die Bandbreite bei TCP und UDP, sowie die verlorenen Pakete und PDV bei UDP. Basierend auf den verwendeten Parametern

der Basistests in Kapitel 4.4.3 ist die Dauer jedes Tests 20 Sekunden. Zudem beginnt jeder TCP-Test erst zwei Sekunden nach dem Start der Übertragung um die Slow-Start Phase von TCP zu vermeiden[1]. Die genauen, für diese Tests verwendeten, Befehle sind dabei

```
iperf3 -c Ziel-IP -V -O 2 -P 1 -t 20 --logfile Dateiname
```

für die TCP-Tests und

```
iperf3 -c Ziel-IP -V -t 20 -P 1 -u -b 10G --logfile Dateiname
```

für die UDP-Tests. Mit dem Befehl `iperf3 -s -D` kann iperf3 als Server-Daemon gestartet werden. Die verwendeten Parameter sind in Tabelle 6.1 beschrieben.

Tabelle 6.1: Netzwerktests: Verwendete iperf3 Startparameter

Parameter	**Beschreibung**
`-c Ziel-IP`	Start von iperf3 als Client verbunden zu Ziel-IP
`-V`	Ausführlichere Ausgabe
`-O 2`	Test beginnt erst zwei Sekunden nach Start der Verbindung
`-P 1`	Anzahl der parallelen Verbindungen
`-t 30`	Testzeitraum in Sekunden
`-u`	UDP-Test
`-b 10G`	Bandbreite für UDP-Tests in Bit/s. Hier steht 10G für 10 Gigabit/s
`--logfile Dateiname`	Speichern von Testergebnissen in einer Datei
`-s`	Start von iperf3 als Server
`-D`	Start von iperf3 als Daemon

6.1.2 Ergebnisse

In diesem Kapitel wird n als Bezeichnung für die Entfernung zwischen den RPi benutzt. $n = 100$ bedeutet so zum Beispiel, dass die Entfernung von jedem RPi zum Nächsten 100 Meter beträgt. Vor der Betrachtung der genauen Ergebnisse der Tests, ist zu sagen, dass bei beiden Repeater-Implementationen Ergebnisse bis zu einer Entfernung von $n = 300$ zwischen den RPi existieren, während die Mesh-Implementation bei einer Entfernung von $n = 200$ keine Verbindung mehr aufbauen konnte. Da als Limit für UDP 10 Gigabit/s festgelegt ist, ist es möglich, dass iperf3 versucht deutlich mehr Pakete über UDP zu senden als das Netzwerk verkraften kann. Dies kann zu einer sehr hohen Rate an verlorenen Paketen führen. Deshalb sind immer, wenn von UDP-Bandbreite die Rede ist, die verlorenen Pakete bereits abgezogen, eingerechnet und somit Teil der UDP-Bandbreite. Darüber hinaus sind die Ergebnisse von Tests mit mehreren gleichzeitigen Verbindungen als Summe oder Durchschnittswert zusammengefasst. Zuerst wird die TCP-Bandbreite bei einer Entfernung von $n = 100$ bei den unterschiedlichen Hops verglichen (Siehe Tabelle 6.2). Die Ergebnisse der 2,4Ghz-Repeater-Kette zeigen dabei, dass sich die Band-

Tabelle 6.2: Luftlinie: TCP Bandbreite in Mbit/s der unterschiedlichen Hops auf 100 Metern

Einzelne Verbindung

Hopanzahl	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
2 Hop	15,70	53,6	1,78
3 Hop	7,08	41,50	5,24
4 Hop	3,720	30,400	1,940

Zwei gleichzeitige Verbindungen

Hopanzahl	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
2 Hop	14,70	58,20	1,68
3 Hop	6,45	45,00	4,14
4 Hop	3,460	35,400	1,050

Fünf gleichzeitige Verbindungen

Hopanzahl	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
2 Hop	5,66	60,10	1,83
3 Hop	5,45	42,60	7,71
4 Hop	3,560	37,400	1,520

breite mit jedem Hop in etwa halbiert. Dies ist ein erwartbares Ergebnis, da diese Kette in Halb-Duplex operiert, wobei die Hälfte der Bandbreite jeweils für Senden und Empfangen genutzt wird. Bei der 2,4Ghz/5Ghz-Kette sinkt die Bandbreite mit jedem Hop im Schnitt um circa 11,5 Mbit/s. Der Unterschied zwischen zwei und vier Hops beträgt, unabhängig von der Anzahl an gleichzeitigen Verbindungen, 23

Mbit/s. Andere Ergebnisse zeigen sich bei der Mesh-Kette. Sobald sich die Pakete ab dem zweiten Hop durch das Mesh bewegen sinkt die Bandbreite deutlich. Einer der möglichen Gründe dafür ist, dass das Mesh auf IEEE802.11g, statt dem IEEE802.11n des Access-Points des ersten Hops, operiert. Zudem ist die Bandbreite bei drei Hops höher als bei zwei und vier. Da die Pakete von batman-adv geroutet werden und nicht wie bei der Repeater-Kette eine feste Route haben, ist es somit möglich, dass diese Verbindung Zwischenstationen überspringt. Es ist zu erkennen, dass die Bandbreite der 2,4Ghz/5Ghz-Kette, prozentual gesehen, weniger durch jeden einzelnen Hop beeinflusst wird als die 2,4Ghz-Kette. Die nächsten betrachteten Ergebnisse sind die der Bandbreite der TCP-Tests (Siehe Tabelle 6.3). Ob ein, zwei oder fünf

Tabelle 6.3: Luftlinie: TCP Bandbreite in Mbit/s auf unterschiedlichen Entfernungen

Einzelne Verbindung

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	3,930	39,200	5,450
100	3,720	30,400	1,940
150	0,419	20,400	0,524
200	0,472	18,100	
300	0,367	16,900	

Zwei gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	5,090	46,700	6,400
100	3,460	35,400	1,050
150	0,315	18,400	0,577
200	0,314	18,600	
300	0,472	17,100	

Fünf gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	3,670	47,000	5,560
100	3,560	37,400	1,520
150	0,262	21,900	0,577
200	0,262	17,200	
300	0,315	17,000	

gleichzeitige Verbindungen bestehen hat augenscheinlich keinen großen Einfluss auf die Bandbreite von 2,4Ghz-Repeatern und des Meshs. Unterschiede existieren zwar, jedoch ist kein regelmäßiges Muster erkennbar, was darauf hindeutet, dass diese Unterschiede auf schwankende Verbindungsqualität und Messabweichungen zurückzuführen sind. Einzig bei den 2,4Ghz/5Ghz-Repeatern ist, bei 50/100 Metern und mehreren gleichzeitigen Verbindungen, eine Steigerung von etwas über 10% gegenüber der einzelnen Verbindung erkennbar. Wie auch bereits bei den Basistests

in Kapitel 4.4 gibt es eine Entfernung bei der die Bandbreite deutlich sinkt. Dies ist zu erkennen bei den Ergebnissen der 150-Meter-Tests und besonders ausgeprägte bei 2,4Ghz-Repeatern sowie dem Mesh. Im Allgemeinen ist sehr deutlich zu sehen, dass die Ergebnisse der 2,4Ghz/5Ghz-Repeater weit über denen der anderen zwei Varianten liegen (Siehe Abbildung 6.3). Auch im schlechtesten Fall ist die Bandbreite

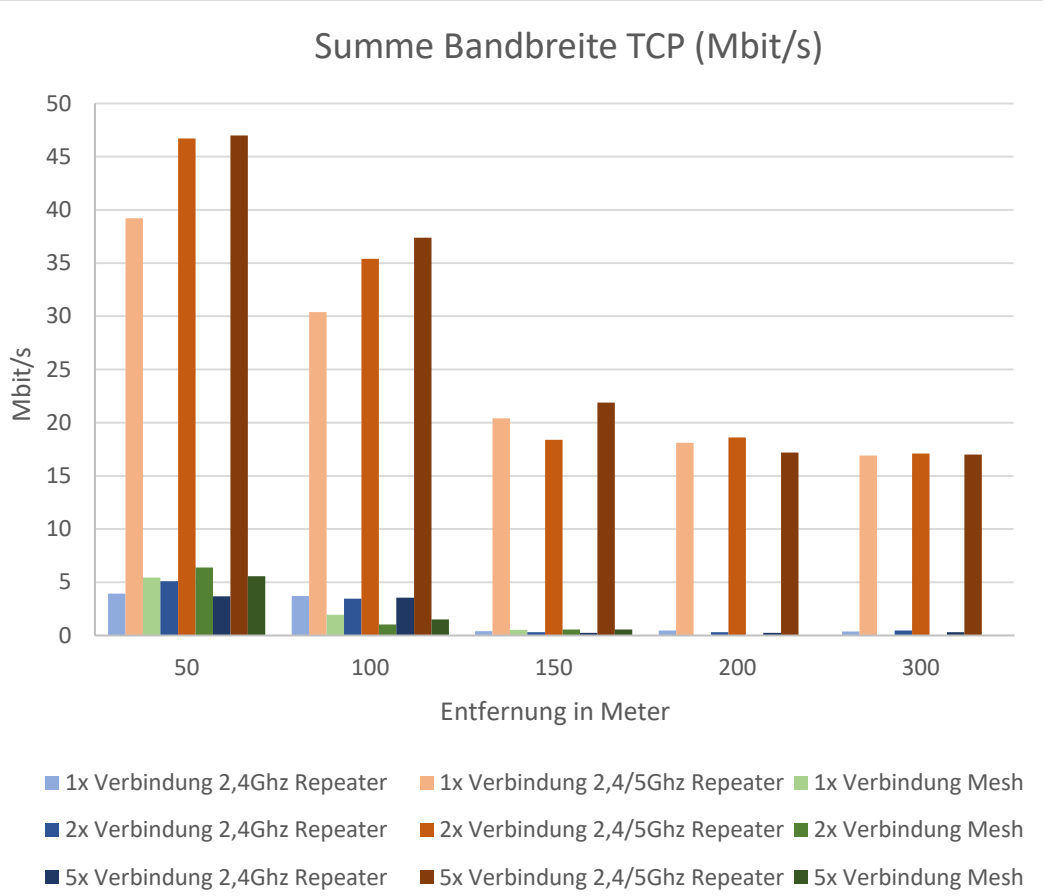

Abbildung 6.3: Luftlinie: TCP Bandbreite in Mbit/s auf unterschiedlichen Entfernungen

noch mehr als dreimal und im besten Fall mehr als 88-mal höher als die der Anderen. Selbst unter Einbeziehung der Tatsache, dass bei den 2,4/5Ghz-Tests ein leistungsstärkerer WLAN-Adapter beim Client zum Einsatz kam, sind diese Ergebnisse sehr eindeutig.

Wie schon bereits bei den Basistests in Kapitel 4.4 ist die UDP-Bandbreite in nahezu allen Tests höher als die TCP-Bandbreite (Siehe Tabelle 6.4). In einzelnen Fällen wie

Tabelle 6.4: Luftlinie: UDP Bandbreite in Mbit/s auf unterschiedlichen Entfernungen

Einzelne Verbindung

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	3,970	46,115	10,647
100	2,946	36,184	2,094
150	1,373	24,146	0,978
200	1,500	16,205	
300		21,411	

Zwei gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	4,095	48,702	10,342
100	3,405	18,540	2,136
150	1,541	21,932	0,932
200	1,663	16,747	
300		18,625	

Fünf gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	4,433	49,907	10,423
100	2,595	15,884	2,196
150	2,003	20,036	0,935
200	2,002	13,581	
300	1,473	20,091	

dem Mesh-Test mit fünf gleichzeitigen Verbindungen ist die Bandbreite fast doppelt so hoch. Auch bei den meisten Fällen in denen TCP eine höhere Bandbreite liefert als UDP, ist diese nur um einstellige Prozente höher. Die zwei großen Ausnahmen hierbei sind die Ergebnisse der 100-Meter-Tests mit 2,4/5Ghz-Repeatern, bei denen, mit fünf gleichzeitigen Verbindungen, die UDP-Bandbreite weniger als die Hälfte der TCP-Bandbreite beträgt. Es zeigt sich nur bei der Mesh-Implementation ein eindeutiger Vorteil für UDP im Vergleich zu TCP. Dort ist die UDP-Bandbreite in jedem Fall höher und in sieben von neun Fällen mindestens ≈62% höher als die TCP-Bandbreite.

Die PDV bleibt bei den 2,4/5Ghz-Repeatern stets unter dem Grenzwert von 50 ms der ITU[11] und übersteigt selbst den Grenzwert 30 ms von Cisco[5] nur bei den 300-Meter-Tests mit fünf gleichzeitigen Verbindungen (Siehe Tabelle 6.5). Mit reinen

Tabelle 6.5: Luftlinie: PDV in ms auf unterschiedlichen Entfernungen

Einzelne Verbindung

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	23,989	1,103	4,93
100	97,242	9,772	39,329
150	53,903	2,606	128,511
200	169,422	5,092	
300		3,959	

Zwei gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	20,811	2,286	8,07
100	70,192	3,353	47,756
150	220,822	5,978	1253,693
200	107,339	5,235	
300		8,029	

Fünf gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50	59,869	2,901	19,983
100	392,384	7,402	2155,987
150	5210,833	8,677	282273,811
200	58531,416	12,002	
300	21813,134	29,304	

2,4Ghz-Repeatern liegt die PDV nur bei 50-Meter-Tests mit einzelner oder zweifacher Verbindung unter den Grenzwerten und steigt bei den 300-Meter-Tests mit fünf gleichzeitigen Verbindungen auf ein Maximum von 21813,134 ms. Die Werte des Meshs liegen bei 50 Metern unterhalb beider Grenzwerte und bei 100 Metern mit einzelner, sowie zweifacher Verbindung unter den Grenzwerten der ITU. Bei den restlichen Tests werden die Grenzwerte, bis zu einem Maximum von 282273,811 ms, überschritten. Im Allgemeinen zeigen die Werte, dass die PDV sowohl mit zunehmender Entfernung, als auch mit der Anzahl gleichzeitiger Verbindungen steigt. Einzig die 2,4/5Ghz-Implementierung bleibt bei höheren Entfernungen und mehreren gleichzeitigen Verbindungen unterhalb der Grenzwerte.

6.2 Kette in Wald

Die zweiten Testreihen finden in einem Wald statt um in einer Umgebung zu testen, in der die Verbindung der RPi eingeschränkter als auf offenem Feld ist.

6.2.1 Testaufbau

Die Tests finden im Bad Vilbeler Wald statt. Der Testaufbau dieser Tests ist gleich dem Aufbau in Kapitel 6.1.1 mit den nachfolgenden Unterschieden. Die RPi befinden sich nicht auf einem Stab, sondern in einer Plastiktüte in welcher sich Löcher für die Antennen befinden. Diese werden an Ästen aufgehangen, die ohne Hilfsmittel leicht erreichbar sind (Siehe Abbildung 6.4). Es wurde darauf geachtet, dass auf der Sichtlinie zwischen den Antennen möglichst wenige Bäume liegen. Da sich in einem

Abbildung 6.4: RPi in Wald

Wald nicht immer in exakten Abständen Bäume mit erreichbaren Ästen befinden, sind die getesteten Abstände zwischen den RPi:

- Test 1: 50m-50m-50m
- Test 2: 100m-70m-100m
- Test 3: 150m-100m-125m

Der Client befand sich bei diesen Tests direkt neben dem letzten Kettenglied. Außerdem wurde, sofern die Verbindung es zuließ, bei jedem der drei Tests, noch ein weiterer Test durchgeführt, bei dem sich der Client 100 Meter von dem letzten Kettenglied entfernt befindet. Auf der Karte in Abbildung 6.5 sind die drei unterschiedlichen Teststrecken eingezeichnet.

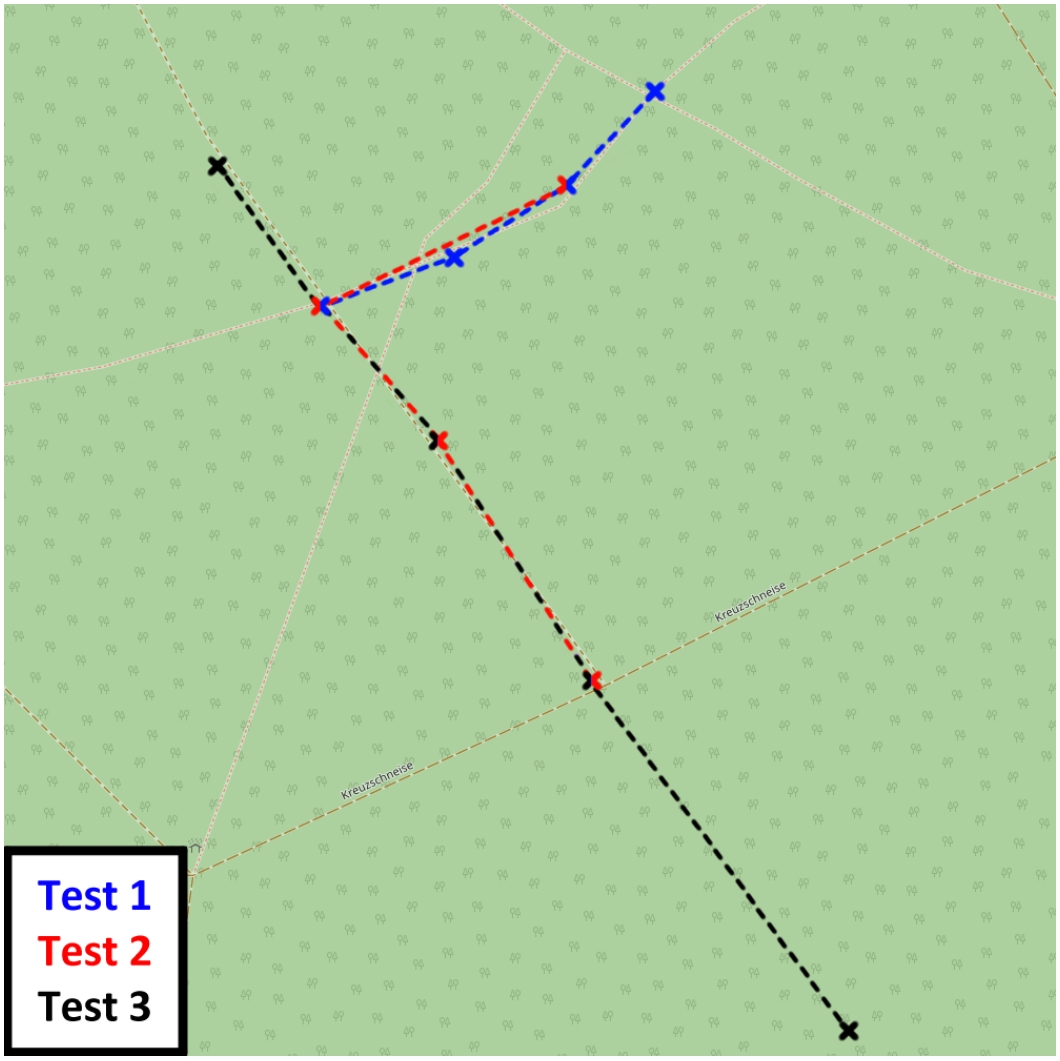

Abbildung 6.5: Karte der Teststrecken im Wald

6.2.2 Ergebnisse

Wie auch bei den Tests auf offener Fläche sind immer, wenn von UDP-Bandbreite die Rede ist, die verlorenen Pakete bereits abgezogen, eingerechnet und somit Teil der UDP-Bandbreite. Die Ergebnisse von Tests mit mehreren gleichzeitigen Verbindungen sind als Summe oder Durchschnittswert zusammengefasst. Zudem konnte die Mesh-Implementation keine Verbindung bei Test 3 aufbauen und die Messung mit fünf gleichzeitigen Verbindungen bei Test 2 nicht durchführen. Basiernd auf der Anzahl gleichzeitiger Verbindungen ist kein regelmäßiges Muster bei der TCP-Bandbreite sichtbar (Siehe Tabelle 6.6). Sowohl bei der 2,4Ghz-Kette als auch bei der Mesh-Kette ist ein Sinken der Bandbreite auf längeren Entfernungen zu erkennen.

Tabelle 6.6: Wald: TCP Bandbreite in Mbit/s auf unterschiedlichen Entfernungen

Einzelne Verbindung

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	2,050	12,000	1,780
50-50-50 + 100	1,570	12,900	1,520
100-70-100	1,840	7,030	1,310
100-70-100 + 100	1,100	9,120	0,524
150-100-125	0,629	29,000	
150-100-125 + 100	0,577	20,000	

Zwei gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	1,470	14,700	4,090
50-50-50 + 100	0,787	8,920	1,100
100-70-100	1,730	10,100	0,472
100-70-100 + 100	1,470	5,870	0,524
150-100-125	1,470	33,000	
150-100-125 + 100	0,262	23,200	

Fünf gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	1,260	18,000	3,250
50-50-50 + 100	1,310	13,400	0,734
100-70-100	1,680	10,400	0,210
100-70-100 + 100	1,470	7,500	
150-100-125	0,944	33,200	
150-100-125 + 100	0,577	21,500	

Dies ist nicht der Fall bei Werten der 2,4/5Ghz-Kette. Dort sind die Werte der höchsten Entfernung mehr als doppelt so hoch wie die Ergebnisse der Kürzesten. Zudem liegt die Bandbreite der 2,4/5Ghz-Kette deutlich über der, der beiden anderen Implementierungen (Siehe Abbildung 6.6).

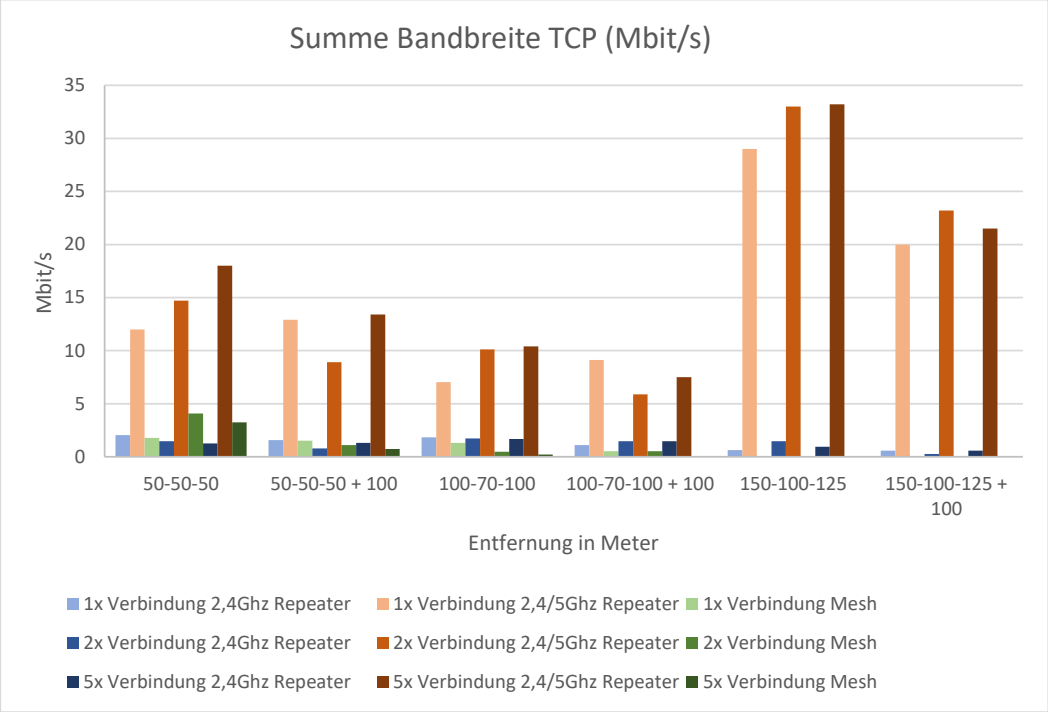

Abbildung 6.6: Wald: TCP Bandbreite in Mbit/s auf unterschiedlichen Entfernungen

Wie auch bei den anderen Tests ist die UDP-Bandbreite in den meisten Fällen höher als die TCP-Bandbreite (Siehe Abbildung 6.7). Je höher die Bandbreite im Allgemeinen ist, desto kleiner ist die Differenz zwischen TCP und UDP. Dies ist besonders auffällig bei den 2,4/5Ghz-Tests mit *150-100-125* Metern. Dort sind zwei der drei Fälle zu finden, in denen TCP vor UDP liegt. Des Weiteren beträgt die UDP-Bandbreite dort maximal das ≈1,2 fache der TCP-Bandbreite.

Tabelle 6.7: Wald: UDP Bandbreite in Mbit/s auf unterschiedlichen Entfernungen

Einzelne Verbindung

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	4,582	15,193	5,641
50-50-50 + 100	1,517	19,900	4,704
100-70-100	4,633	12,729	4,329
100-70-100 + 100	2,106	15,663	1,889
150-100-125	5,020	34,646	
150-100-125 + 100	2,833	24,087	

Zwei gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	4,477	23,896	0,801
50-50-50 + 100	1,714	17,027	4,254
100-70-100	7,110	14,959	1,502
100-70-100 + 100	2,107	15,971	1,657
150-100-125	2,592	23,128	
150-100-125 + 100	2,353	20,493	

Fünf gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	5,084	21,590	3,358
50-50-50 + 100	1,970	19,367	4,176
100-70-100	7,030	6,134	8,482
100-70-100 + 100	1,985	15,549	
150-100-125	2,893	34,300	
150-100-125 + 100	2,247	23,393	

Die PDV bleibt bei den 2,4/5Ghz-Repeatern, mit Ausnahme von einem Fall, stets unter dem Grenzwert von 50 ms der ITU[11] und dem Grenzwert 30 ms von Cisco[5]. Die 2,4Ghz-Repeater bleiben nur bei den *50-50-50* und *100-70-100* Tests, mit einzelnen oder zweifachen Verbindungen, unter den Grenzwerten und übersteigen sie sonst bis zu einem Maximum von 606,455 ms. Das Mesh liegt bei den 50-50-50 und 50-50-50+100 Tests, mit einer Ausnahme, unterhalb der Grenze und sonst darüber. Im Allgemeinen steigt die PDV besonders mit mehreren gleichzeitigen Verbindungen.

Tabelle 6.8: Wald: PDV in ms auf unterschiedlichen Entfernungen

Einzelne Verbindung

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	9,504	2,480	6,611
50-50-50 + 100	138,400	0,745	7,429
100-70-100	6,705	6,241	37,690
100-70-100 + 100	45,418	2,281	72,597
150-100-125	13,894	1,557	
150-100-125 + 100	22,232	1,935	

Zwei gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	23,230	7,629	9,180
50-50-50 + 100	117,262	1,835	23,503
100-70-100	9,484	9,954	59,661
100-70-100 + 100	61,620	2,293	47,559
150-100-125	92,229	3,228	
150-100-125 + 100	93,490	2,084	

Fünf gleichzeitige Verbindungen

Entfernung in m	2,4Ghz Repeater	2,4/5Ghz Repeater	Mesh
50-50-50	80,398	18,768	17,100
50-50-50 + 100	395,573	4,190	94,847
100-70-100	14,558	86,258	106,018
100-70-100 + 100	606,455	23,517	
150-100-125	294,280	6,784	
150-100-125 + 100	211,633	3,629	

6.3 Gesamtbewertung

Vergleicht man die drei Implementationen ist es eindeutig, dass die 2,4/5Ghz-Kette die mit Abstand besten Werte in sowohl Reichweite, Bandbreite als auch Stabilität (in Form von PDV) bietet. Hier ist es wichtig nicht zu vergessen, dass bei den 2,4/5Ghz-Tests, statt dem internen WLAN-Adapter, einer der externen USB-WLAN-Adapter von TP-Link verwendet wurde und damit die Bandbreite bei diesen Tests leicht höher ausfällt. Im Gegensatz dazu haben die 5Ghz-Adapter kleinere Antennen als die externen 2,4Ghz-Adaptern. Allerdings sind die Ergebnisse der 2,4/5Ghz-Kette so deutlich besser als die Ergebnisse der beiden anderen Implementation (Zum Beispiel ≈3,59 bis ≈88,55 fache TCP-Bandbreite), dass dies zu vernachlässigen ist.

Die Ergebnisse der 2,4Ghz-Kette zeigen die Begrenzung einer Implementation die in Halb-Duplex operiert. Da jeder Hop die Bandbreite halbiert ist es bei solch einer Implementation wichtig, eine möglichst lange Strecke mit jedem Hop abzudecken. Selbst bei den hier getesteten vier Hops liegt die Bandbreite, bei Entfernungen von mehr als 100 Metern pro Hop, bereits unter 1Mbit/s. Dies ist nur sehr bedingt nutzbar, insbesondere wenn über solche eine Kette das Internet genutzt werden soll.

Die Mesh-Kette zeigt bei Entfernungen von 50 Metern pro Hop eine höhere und bei 100 Metern pro Hop eine gleichwertige Bandbreite wie die 2,4Ghz-Kette. Höhere Entfernungen sind nur bei einer offenen Fläche möglich und selbst dort liegt die Grenze bei 150 Metern. Selbst ein einzelner Hop im Mesh senkt die Bandbreite auf unter 10 Mbit/s. Einige Dinge sind hierbei zu bedenken. Die MTU konnte, bei der hier verwendeten Version von Raspbian, nicht auf die mindestens empfohlenen 1528 erhöht werden, was zu einer erhöhten Anzahl von Paketen im Mesh führen kann. Zudem unterstützen die Mesh-Knoten nur IEEE802.11b. Darüber hinaus ist eine Kette, bei der ein Knoten nur den jeweils nächsten und vorherigen Knoten erreichen kann, nicht der optimale Aufbau für ein Mesh.

Die PDV liegt nur bei der 2,4/5Ghz-Kette zuverlässig unter den ITU-Grenzwerten[11] von 50 ms sowie, abgesehen von zwei Fällen, auch unter den Cisco-Grenzwerten[5] von 30 ms. Die PDV der 2,4Ghz-Kette überschreitet diese Grenzwerte selbst bei einfachen Verbindungen ab 100 Metern auf offenem Gelände und steigt bei den Tests auf bis zu 21813,134 ms. Ähnliches gilt für die Mesh-Kette, die auf 100 Metern mit einzelnen und zweifachen Verbindungen knapp unter den Grenzwerten der ITU bleibt und sie, bei höheren Entfernungen, mit bis zu 282273,811 ms deutlich überschreitet. Den größten Einfluss auf PDV hat die Anzahl der gleichzeitigen Verbindungen. Dies zeigt sich insbesondere bei fünf gleichzeitigen Verbindungen.

Bis zu einer Entfernung von 600 Metern (200 Meter pro Hop) liegt selbst die Bandbreite des Basistest aus Kapitel 4.4 um mehr als einen Faktor von 20 über der Bandbreite sowohl der 2,4Ghz- als auch der Mesh-Kette. Dasselbe gilt ebenso für die Werte von PDV. Bei den Messungen im Wald, sowie den Basistests, zeigt sich wie wichtig die Positionierung der Antenne ist. Während der Basistests konnte, für einen einzelnen Hop, maximal eine Entfernung von 80 Metern überbrückt werden, sofern sich der RPi auf dem Boden befindet. Dies steht im Gegensatz zu dem auf einem Stab befestigten RPi, der selbst bei einer Entfernung von 600 Metern noch eine Verbindung aufbauen konnte. Test 2 im Wald ergab eine niedrigere Bandbreite als Test 3, obwohl die reine Strecke insgesamt 105 Meter kürzer ist. Da in einem Wald viele Objekte den Weg versperren können, ist es nicht immer möglich einen hohen, offenen Platz für die Antenne zu finden. Die Antennen sollten somit, soweit es möglich ist, weit oben und mit möglichst freier Luftlinie zur vorherigen und nächsten Antenne angebracht sein.

7 Energieverbauch des Raspberry Pi

Dieses Kapitel beschäftigt sich mit dem Energieverbrauch eines RPi sowie den Möglichkeiten Energie zu sparen. Es beinhaltet mögliche Energiesparoptionen und einen Vollzeittest der Implementationen mit Akkus.

7.1 Energiesparoptionen

Es bieten sich mit einem RPi eine Hand voll Möglichkeiten Energie zu sparen. Die getesteten Optionen und ihre Energieersparnis finden sich in Tabelle 7.1. Gemessen

Tabelle 7.1: Getestete Energiesparoptionen und ihr Einfluss

Option	**Energieersparnis**
Status-LEDs deaktivieren	Kein messbarer Unterschied
HDMI-Anschluss deaktivieren	20mA
Untertakten	Kein messbarer Unterschied

wurde die Stromstärke in Milliampere mit Hilfe zweier USB-Strommessgeräten der Marken Logilink und Portapow. Die einzige messbare Energiesparoptionen ist das Deaktivieren des HDMI-Anschlusses. Es ist theoretisch noch möglich andere Funktionen des RPi, wie die USB-Anschlüsse, zu deaktivieren. Allerdings werden diese hier benötigt. Große Energieverbraucher sind USB-Geräte wie die verwendeten USB-WLAN-Adapter. Der Energieverbrauch liegt dabei zwischen 60mA und 180mA, je nach derzeitiger Auslastung des Adapters. Um den HDMI-Anschluss zu deaktivieren kann der Befehl `/usr/bin/tvservice -o` verwendet werden.

7.2 Tests

In den folgenden Unterkapiteln ist die Messung des Energieverbrauchs, in Relation zu dem verwendeten Akku, beschrieben.

7.2.1 Testaufbau

Es werden sechs verschiedene Tests durchgeführt um die Akkulaufzeit der unterschiedlichen Implementationen zu testen. Dabei soll es für jede der drei Implementationen jeweils einen Test unter Last und einen Test bei Inaktivität geben. Ein einzelner Knoten der jeweiligen Kette ist an einen voll aufgeladenen Akku angeschlossen. Der verwendete Akku für diese Tests ist eine Ravpower 20.000mAh Powerbank RP-PB006.

Für die Last-Tests erzeugt ein dauerhaft laufender iperf3-Test einen Datenstrom der, im Fall der beiden Repeater-Ketten, von einem iperf3-Client (RPi1) über den zu testenden RPi2 zu einem iperf3-Server (RPi3) fließt (Siehe Abbildung 7.1). Im Fall

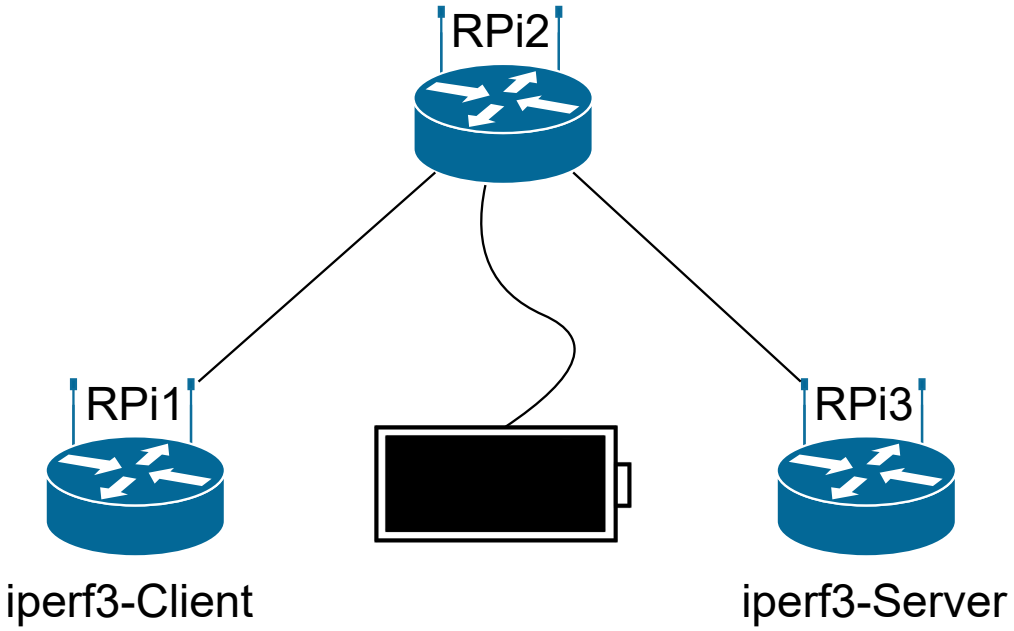

Abbildung 7.1: Testaufbau des Akkutests

der Mesh-Kette agiert der zu testende RPi selbst als iperf3-Server, an den ein zweiter Mesh-Knoten die Daten überträgt.

Ein Skript schreibt einmal pro Minute das derzeitige Datum in eine Datei. Damit ist erkennbar zu welcher Uhrzeit der RPi des letzte Mal Strom hatte.

7.2.2 Ergebnisse

Die kürzeste Akkulaufzeit bietet die 2,4/5Ghz-Implementation mit 31 Stunden, 55 Minuten und 22 Sekunden bei Inaktivität und 19 Stunden, 13 Minuten und 57 Sekunden unter Last. Die nächstlängere Akkulaufzeit erreicht die 2,4Ghz-Implementation mit 40 Stunden, 10 Minuten und 27 bei Inaktivität (≈25,85% länger als 2,4/5Ghz) und 40 Stunden, 10 Minuten und 27 Sekunden unter Last (≈71,92% länger als 2,4/5Ghz). Mit 52 Stunden, 25 Minuten und 56 Sekunden bei Inaktivität (≈64,25% länger als 2,4/5Ghz) und 46 Stunden, 59 Minuten und 33 Sekunden unter Last (≈144,34% länger als 2,4/5Ghz), liegt die längste erreichte Laufzeit bei der Mesh-Implementation (Siehe Tabelle 7.2).

Tabelle 7.2: Maximale Akkulaufzeit mit unterschiedlichen Implementierungen

	Akku ist leer in
2,4Ghz-Repeater Last	33 Stunden, 3 Minuten und 54 Sekunden
2,4Ghz-Repeater Inaktivität	40 Stunden, 10 Minuten und 27 Sekunden
2,4/5Ghz-Repeater Last	19 Stunden, 13 Minuten und 57 Sekunden
2,4/5Ghz-Repeater Inaktivität	31 Stunden, 55 Minuten und 22 Sekunden
Mesh Last	46 Stunden, 59 Minuten und 33 Sekunden
Mesh Inaktivität	52 Stunden, 25 Minuten und 56 Sekunden

Sowohl die 2,4Ghz- als auch die Mesh-Implementation verwenden den RPi 3 B, welcher von sich aus einen geringeren Energieverbrauch hat als der, von der 2,4/5Ghz-Implementation verwendete, RPi 3 B+. Dies zeigt sich auch in den Ergebnissen. Das Mesh verwendet nur einen (zwei als Gateway oder als Knoten mit Access-Point) statt der zwei USB-WLAN-Adapter der 2,4/5Ghz-Implementation. Dies macht sich auch in den Ergebnissen bemerkbar. Unter Last liegen die Implementationen weiter auseinander als bei Inaktivität. Ursache dafür ist der erhöhte Energieverbrauch bei höherer Bandbreite.

8 Fazit

Ein WLAN mit Hilfe von Raspberry Pi zu verlängern ist durchaus möglich. Von den drei getesteten Implementationen zeigte sich die Mesh-Implementation als schlechteste Option. Die erreichte Reichweite war am Geringsten, da die Verbindung bereits bei über 150 Metern pro Hop abbrach. Zudem lag die erreichte Bandbreite und Zuverlässigkeit sogar unter der einer einzelnen, direkten Verbindung. Als einzigen Vorteil zeigt sich der geringste Stromverbrauch aller Implementationen.

Die 2,4Ghz-Implementation zeigt sich als valide Option für eine Kette mit wenigen Hops. Da es sich dabei um eine Kette handelt die auf Halb-Duplex basiert, wird auch die Bandbreite mit jedem Hop mindestens halbiert. Des Weiteren sinkt die Bandbreite der Kette bei einer Entfernung von mehr als 100 Metern pro Hop um circa einen Faktor von 10 auf unter 1Mbit/s. Auf diesem Wert ist sie dann bei bis zu 300 Metern pro Hop stabil. Somit ist die erreichte Reichweite höher als die des Meshs und bietet einen Vorteil gegenüber einer direkten Verbindung mit einem einzelnen Repeater. Allerdings ist die Bandbreite so niedrig, dass sie nur bedingt nutzbar ist.

Die 2,4/5Ghz-Implementation löst diese Halb-Duplex-Probleme indem abwechselnd ein 2,4- und 5Ghz-WLAN verwendet wird. Jeder Hop senkt dabei die Bandbreite nur um ≈11,5Mbit/s. Selbst bei nur 2 Hops ist dieser Vorteil deutlich erkennbar.

Die Einrichtung beider Repeater-Ketten ist einfach zu bewältigen. Schwieriger gestaltet sich die eines batman-Meshs. Es gibt mehrere Kompatibilitätsprobleme, sowohl mit verschiedenen Versionen von Raspbian, als auch mit Updates gewisser Linux-Pakete. Einstellungsmöglichkeiten wie das Erhöhen der MTU auf über 1500 und eine Verwendung eines anderen WLAN-Kanals als Kanal 1 sind in der verwendeten Version von Raspbian-Stretch nicht möglich.

Der mobile Betrieb über eine Powerbank ist problemlos durchführbar. Je höher die Bandbreite der jeweiligen Implementation, desto größer auch der Stromverbrauch. Die Laufzeit lag dabei zwischen 19 und 52 Stunden, je nach Implementation und Auslastung. Die 2,4/5Ghz-Kette verbraucht am meisten Strom. Dies könnte in Zukunft allerdings eventuell reduziert werden indem, statt des internen WLAN-Adapters des RPi 3 B+, ein RPi 3 B mit einem 5Ghz-fähigen USB-WLAN-Adapter versehen wird. Auch andere Stromsparoptionen, wie die Nutzung eines Solarpanels

oder eines Systems zur automatischen Abschaltung bei längerer Inaktivität könnten zur Reduzierung des Stromverbrauchs beitragen. Zudem ist es durchaus denkbar, dass auch weitere Entfernung als 300 Meter pro Hop oder eine Kette mit mehr Hops realisierbar sind.

Anhang

```
sudo apt-get update
#sudo apt-get upgrade
sudo apt-get install dnsmasq hostapd -y
sudo wget http://downloads.fars-robotics.net/wifi-drivers/install-wifi -O
     /usr/bin/install-wifi
sudo chmod +x /usr/bin/install-wifi
sudo /usr/bin/install-wifi
sudo systemctl stop dnsmasq
sudo systemctl stop hostapd
echo "Please enter the third octet of the AP network 192.168.xxx.0:"
read netid
sudo mv /etc/dhcpcd.conf /etc/dhcpcd.conf.orig
sudo cat > /etc/dhcpcd.conf <<EOF
# A sample configuration for dhcpcd.
# See dhcpcd.conf(5) for details.

# Allow users of this group to interact with dhcpcd via the control
    socket.
#controlgroup wheel

# Inform the DHCP server of our hostname for DDNS.
hostname

# Use the hardware address of the interface for the Client ID.
clientid
# or
# Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per
    RFC4361.
#duid

# Persist interface configuration when dhcpcd exits.
persistent

# Rapid commit support.
# Safe to enable by default because it requires the equivalent option set
# on the server to actually work.
option rapid_commit

```

```
# A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
# Most distributions have NTP support.
option ntp_servers
# Respect the network MTU.
# Some interface drivers reset when changing the MTU so disabled by
    default.
#option interface_mtu

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

# A hook script is provided to lookup the hostname if not set by the DHCP
# server, but it should not be run by default.
nohook lookup-hostname

interface wlan0
static ip_address=192.168.$netid.1/24
nohook wpa_supplicant
EOF
sleep 1s
sudo service dhcpcd restart
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
sudo cat > /etc/dnsmasq.conf <<EOF
interface=wlan0      # Use the require wireless interface - usually wlan0
dhcp-range=192.168.$netid.2,192.168.$netid.20,255.255.255.0,96h
dhcp-option=3,192.168.$netid.1
EOF
sleep 1s
echo "Please enter the ssid for the new AP"
read apssid
echo "Please enter the password for the new AP"
read appsk
sudo cat > /etc/hostapd/hostapd.conf <<EOF
interface=wlan0
driver=nl80211
ssid=$apssid
ieee80211d=1
country_code=DE

hw_mode=g
ieee80211n=1
ht_capab=[SHORT-GI-40][HT40+][DSSS_CCK-40]
channel=1
wmm_enabled=1

macaddr_acl=0
```

```
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=$appsk
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
EOF
sleep 1s
sudo mv /etc/default/hostapd /etc/default/hostapd.orig
sudo cat > /etc/default/hostapd <<EOF
# Defaults for hostapd initscript
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
#
DAEMON_CONF="/etc/hostapd/hostapd.conf"

# Additional daemon options to be appended to hostapd command:-
#   -d   show more debug messages (-dd for even more)
#   -K   include key data in debug messages
#   -t   include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automatically
# configured by the init.d script and must not be added to DAEMON_OPTS.
#
#DAEMON_OPTS=""

EOF
sleep 1s
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
sudo systemctl start dnsmasq
sudo mv /etc/sysctl.conf /etc/sysctl.conf.orig
sudo cat > /etc/sysctl.conf <<EOF
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

```

```
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

###################################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path
    filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
```

```
#net.ipv4.conf.all.log_martians = 1
#

EOF
sleep 1s
sudo sysctl net.ipv4.ip_forward=1
sudo iptables --table nat --append POSTROUTING --out-interface wlan1 -j
    MASQUERADE
sudo iptables --append FORWARD --in-interface wlan0 -j ACCEPT
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
sleep 1s

echo "Please enter the source ssid:"
read source_ssid

echo "Please enter the source password:"
read source_psk

sudo mv /etc/wpa_supplicant/wpa_supplicant.conf /etc/wpa_supplicant/
    wpa_supplicant.conf.orig
sudo cat > /etc/wpa_supplicant/wpa_supplicant.conf <<EOF
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="$source_ssid"
    psk="$source_psk"
}
EOF
sleep 1s
sudo mv /etc/network/interfaces /etc/network/interfaces.orig
sudo cat > /etc/network/interfaces <<EOF
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpc
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

#iface eth0 inet manual

allow-hotplug wlan0
iface wlan0 inet static
address 192.168.$netid.1
netmask 255.255.255.0

allow-hotplug wlan1
```

```
iface wlan1 inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

EOF
sleep 1s
wget https://iperf.fr/download/ubuntu/libiperf0_3.1.3-1_armhf.deb
wget https://iperf.fr/download/ubuntu/iperf3_3.1.3-1_armhf.deb
sudo dpkg -i libiperf0_3.1.3-1_armhf.deb iperf3_3.1.3-1_armhf.deb
rm libiperf0_3.1.3-1_armhf.deb iperf3_3.1.3-1_armhf.deb
sudo crontab -e
```

Installationsskript: 2,4/5 Ghz-Repeater mit 2,4 Ghz-Access-Point

```
sudo apt-get update
#sudo apt-get upgrade
sudo apt-get install dnsmasq hostapd -y
sudo wget http://downloads.fars-robotics.net/wifi-drivers/install-wifi -O
     /usr/bin/install-wifi
sudo chmod +x /usr/bin/install-wifi
sudo /usr/bin/install-wifi
sudo systemctl stop dnsmasq
sudo systemctl stop hostapd
echo "Please enter the third octet of the AP network 192.168.xxx.0:"
read netid
sudo cat > /usr/local/bin/start5ghzap.sh <<EOF
sudo rfkill unblock wifi
sleep 5s
sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
EOF
sleep 1s
sudo mv /etc/dhcpcd.conf /etc/dhcpcd.conf.orig
sudo cat > /etc/dhcpcd.conf <<EOF
# A sample configuration for dhcpcd.
# See dhcpcd.conf(5) for details.

# Allow users of this group to interact with dhcpcd via the control
    socket.
#controlgroup wheel

# Inform the DHCP server of our hostname for DDNS.
hostname

# Use the hardware address of the interface for the Client ID.
clientid
# or
# Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per
    RFC4361.
#duid

```

```
# Persist interface configuration when dhcpcd exits.
persistent

# Rapid commit support.
# Safe to enable by default because it requires the equivalent option set
# on the server to actually work.
option rapid_commit

# A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
# Most distributions have NTP support.
option ntp_servers
# Respect the network MTU.
# Some interface drivers reset when changing the MTU so disabled by
    default.
#option interface_mtu

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate Stable Private IPv6 Addresses instead of hardware based ones
slaac private

# A hook script is provided to lookup the hostname if not set by the DHCP
# server, but it should not be run by default.
nohook lookup-hostname

interface wlan1
static ip_address=192.168.$netid.1/24
nohook wpa_supplicant
EOF
sleep 1s
sudo service dhcpcd restart
sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
sudo cat > /etc/dnsmasq.conf <<EOF
interface=wlan1      # Use the require wireless interface - usually wlan0
dhcp-range=192.168.$netid.2,192.168.$netid.20,255.255.255.0,96h
dhcp-option=3,192.168.$netid.1
EOF
sleep 1s
echo "Please enter the ssid for the new AP"
read apssid
echo "Please enter the password for the new AP"
read appsk
sudo cat > /etc/hostapd/hostapd.conf <<EOF
interface=wlan1
driver=nl80211
ssid=$apssid
ieee80211d=1
country_code=DE
```

```
84
85 hw_mode=a
86 ieee80211n=1
87 ieee80211ac=1
88 channel=36
89 require_vht=1
90 wmm_enabled=1
91
92 macaddr_acl=0
93 auth_algs=1
94 #ignore_broadcast_ssid=0
95 wpa=2
96 wpa_passphrase=$appsk
97 wpa_key_mgmt=WPA-PSK
98 #wpa_pairwise=TKIP
99 rsn_pairwise=CCMP
100 EOF
101 sleep 1s
102 sudo mv /etc/default/hostapd /etc/default/hostapd.orig
103 sudo cat > /etc/default/hostapd <<EOF
104 # Defaults for hostapd initscript
105 #
106 # See /usr/share/doc/hostapd/README.Debian for information about
    alternative
107 # methods of managing hostapd.
108 #
109 # Uncomment and set DAEMON_CONF to the absolute path of a hostapd
    configuration
110 # file and hostapd will be started during system boot. An example
    configuration
111 # file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
112 #
113 DAEMON_CONF="/etc/hostapd/hostapd.conf"
114
115 # Additional daemon options to be appended to hostapd command:-
116 #   -d   show more debug messages (-dd for even more)
117 #   -K   include key data in debug messages
118 #   -t   include timestamps in some debug messages
119 #
120 # Note that -B (daemon mode) and -P (pidfile) options are automatically
121 # configured by the init.d script and must not be added to DAEMON_OPTS.
122 #
123 #DAEMON_OPTS=""
124
125 EOF
126 sleep 1s
127 sudo systemctl unmask hostapd
128 sudo systemctl enable hostapd
129 sudo systemctl start hostapd
130 sudo systemctl start dnsmasq
131 sudo mv /etc/sysctl.conf /etc/sysctl.conf.orig
```

```
sudo cat > /etc/sysctl.conf <<EOF
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#

#kernel.domainname = example.com

# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3

##############################################################3
# Functions previously found in netbase
#

# Uncomment the next two lines to enable Spoof protection (reverse-path
    filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1

# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1

# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net.ipv4.conf.all.secure_redirects = 1
```

```
#
# Do not send ICMP redirects (we are not a router)
#net.ipv4.conf.all.send_redirects = 0
#
# Do not accept IP source route packets (we are not a router)
#net.ipv4.conf.all.accept_source_route = 0
#net.ipv6.conf.all.accept_source_route = 0
#
# Log Martian Packets
#net.ipv4.conf.all.log_martians = 1
#

EOF
sleep 1s
sudo sysctl net.ipv4.ip_forward=1
sudo iptables --table nat --append POSTROUTING --out-interface wlan0 -j
    MASQUERADE
sudo iptables --append FORWARD --in-interface wlan1 -j ACCEPT
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
sleep 1s

echo "Please enter the source ssid:"
read source_ssid

echo "Please enter the source password:"
read source_psk

sudo mv /etc/wpa_supplicant/wpa_supplicant.conf /etc/wpa_supplicant/
    wpa_supplicant.conf.orig
sudo cat > /etc/wpa_supplicant/wpa_supplicant.conf <<EOF
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="$source_ssid"
    psk="$source_psk"
}
EOF
sleep 1s
sudo mv /etc/network/interfaces /etc/network/interfaces.orig
sudo cat > /etc/network/interfaces <<EOF
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpc
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback
```

```

#iface eth0 inet manual

allow-hotplug wlan1
iface wlan1 inet static
address 192.168.$netid.1
netmask 255.255.255.0

allow-hotplug wlan0
iface wlan0 inet dhcp
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

EOF
sleep 1s
wget https://iperf.fr/download/ubuntu/libiperf0_3.1.3-1_armhf.deb
wget https://iperf.fr/download/ubuntu/iperf3_3.1.3-1_armhf.deb
sudo dpkg -i libiperf0_3.1.3-1_armhf.deb iperf3_3.1.3-1_armhf.deb
rm libiperf0_3.1.3-1_armhf.deb iperf3_3.1.3-1_armhf.deb
sudo crontab -e
```

Installationsskript: 2,4/5 Ghz-Repeater mit 5 Ghz-Access-Point

```
echo "%2 at range = %3m with %4 hop(s)"
echo "Starting test(%1)"
echo "TCP Test"
echo "Single connection"
call iperf3.exe -c %1 -V -O 2 -t 20 --logfile %2_%3m_%4hop.txt
echo "2x connection"
call iperf3.exe -c %1 -V -O 2 -P 2 -t 20 --logfile %2_%3m_%4hop_2x.txt
echo "5x connection"
call iperf3.exe -c %1 -V -O 2 -P 5 -t 20 --logfile %2_%3m_%4hop_5x.txt

echo "UDP Test"
echo "Single connection"
call iperf3.exe -c %1 -V -t 20 -u -b 10G --logfile %2_%3m_%4hop_udp.txt
echo "2X connection"
call iperf3.exe -c %1 -V -t 20 -P 2 -u -b 10G --logfile %2_%3m_%4
    hop_udp_2x.txt
echo "5x connection"
call iperf3.exe -c %1 -V -t 20 -P 5 -u -b 10G --logfile %2_%3m_%4
    hop_udp_5x.txt
```

iperf3 Skript für Messungen

Literatur

[1]M. Allman und V. Paxson - ICSI; E. Blanton - Purdue University. *RFC5681: TCP Congestion Control*. Internet Engineering Task Force, Network Working Group, 2009 (zitiert auf Seite 37).

[2]Edimax. *EW-7612UAn V2 Datasheet*. 2019. URL: https://www.edimax.com/edimax/mw/cufiles/files/download/edimaxDE/transfer/products/EW-7612UAnV2/EW-7612UAn_V2_Datasheet.zip (besucht am 3. Juni 2019) (zitiert auf Seite 4).

[3]Elektronik Kompendium. *Mobilfunktechnik (Grundlagen)*. 2019. URL: https://www.elektronik-kompendium.de/sites/kom/0406221.htm (besucht am 3. Juni 2019) (zitiert auf Seite 2).

[4]Freifunk Franken. *Raspberry Pi ins B.A.T.M.A.N.* 2019. URL: https://wiki.freifunk-franken.de/w/Raspberry_Pi_ins_B.A.T.M.A.N. (besucht am 3. Juni 2019) (zitiert auf Seite 25).

[5]Tim Szigeti; Christina Hattingh. *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs*. 2004. Kap. Quality of Service Design Overview (zitiert auf den Seiten 13, 42, 48, 49).

[6]http://pantofflhelden.com. *Raspberry Pi – Wifi repeater im Eigenbau*. 2019. URL: http://pantofflhelden.com/2013/02/raspberry-pi-wifi-repeater-im-eigenbau/ (besucht am 3. Juni 2019) (zitiert auf Seite 25).

[7]https://jpinjpblog.wordpress.com. *Setting up B.A.T.M.A.N. mesh on RPi3*. 2019. URL: https://jpinjpblog.wordpress.com/2017/12/06/setting-up-b-a-t-m-a-n-mesh-on-rpi3/ (besucht am 3. Juni 2019) (zitiert auf Seite 25).

[8]iPerf. *iPerf - Download iPerf3 and original iPerf pre-compiled binaries*. 2019. URL: https://iperf.fr/iperf-download.php (besucht am 3. Juni 2019) (zitiert auf Seite 13).

[9]iperf. *What is iPerf / iPerf3 ?* 2019. URL: https://iperf.fr/ (besucht am 3. Juni 2019) (zitiert auf Seite 5).

[10]C. Demichelis - Telecomitalia Lab; P. Chimento - Ericsson IPI. *RFC3393: IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)*. Internet Engineering Task Force, Network Working Group, 2002 (zitiert auf Seite 13).

[11]Telecommunication Standardization Sector of ITU. *Y.1541 Network performance objectives for IP-based services*. International Telecommunication Union, 2011 (zitiert auf den Seiten 13, 42, 48, 49).

[12]Kortex. *Kortex Xtend Lite V1.0.0*. 2019. URL: https://www.tindie.com/products/kortex_am/kortex-xtend-lite-v100/ (besucht am 3. Juni 2019) (zitiert auf Seite 2).

[13]TP-Link. *TL-WN722N V3 Datasheet*. 2019. URL: https://static.tp-link.com/2018/201810/20181022/TL-WN722N(EU&US)_3.0_datasheet.pdf (besucht am 3. Juni 2019) (zitiert auf Seite 4).

[14]TP-Link. *TP-Link M7450*. 2019. URL: https://www.tp-link.com/de/home-networking/mifi/m7450/ (besucht am 3. Juni 2019) (zitiert auf Seite 2).

[15]MrEngman. *(UPDATE) Drivers for TL-WN725N V2 - 3.6.11+ -> 4.xx.xx+*. 2019. URL: https://www.raspberrypi.org/forums/viewtopic.php?f=28&t=62371&sid=97f79dbe9f8ac40727b1c4ba236c9454 (besucht am 3. Juni 2019) (zitiert auf Seite 8).

[16]MrEngman. *Fars Robotics Website*. 2019. URL: http://downloads.fars-robotics.net/ (besucht am 3. Juni 2019) (zitiert auf Seite 8).

[17]Raspberry Pi Foundation. *Download Raspbian for Raspberry Pi*. 2019. URL: https://www.raspberrypi.org/downloads/raspbian/ (besucht am 3. Juni 2019) (zitiert auf Seite 3).

[18]Raspberry Pi Foundation. *Raspberry Pi Documentation FAQs*. 2019. URL: https://www.raspberrypi.org/documentation/faqs/ (besucht am 3. Juni 2019) (zitiert auf Seite 3).

[19]Raspbian. *Welcome to Raspbian*. 2019. URL: https://www.raspbian.org/ (besucht am 3. Juni 2019) (zitiert auf Seite 3).

[20]Simon Wunderlich; Marek Lindner; Andrew Lunn. *batctl - html man page (v2019.1-3-g1ca604d)*. URL: https://downloads.open-mesh.org/batman/manpages/batctl.8.html (zitiert auf Seite 4).

[21]u/EveningStarNM. *How To Configure batman-adv on the Raspberry Pi 3*. 2019. URL: https://www.reddit.com/r/darknetplan/comments/68s6jp/how_to_configure_batmanadv_on_the_raspberry_pi_3/ (besucht am 3. Juni 2019) (zitiert auf Seite 25).

[22]Wikipedia. *Raspberry Pi*. 2019. URL: https://de.wikipedia.org/wiki/Raspberry_Pi (besucht am 3. Juni 2019) (zitiert auf Seite 3).

[23]www.open-mesh.org. *B.A.T.M.A.N. advanced*. 2019. URL: https://www.open-mesh.org/projects/batman-adv/wiki/Wiki (besucht am 3. Juni 2019) (zitiert auf Seite 4).

[24]www.open-mesh.org. *B.A.T.M.A.N. Advanced Documentation Overview*. 2019. URL: https://www.open-mesh.org/projects/batman-adv/wiki/Doc-overview (besucht am 3. Juni 2019) (zitiert auf Seite 4).

[25]www.open-mesh.org. *B.A.T.M.A.N. Concept*. 2019. URL: https://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept (besucht am 3. Juni 2019) (zitiert auf Seite 4).

[26]www.open-mesh.org. *Fragmentation-technical - batman-adv - Open Mesh*. 2019. URL: https://www.open-mesh.org/projects/batman-adv/wiki/Fragmentation-technical (besucht am 3. Juni 2019) (zitiert auf Seite 32).

Abbildungsverzeichnis

Tabellenverzeichnis