GFZ und Universität Potsdam

Ausarbeitung

Drahtlose Übertragung von Sensordaten in bewaldeten Gebieten bei geringer Mobilfunkabdeckung

Andreas Meisel

Aufgabenstellung und Betreuung:
Prof. Dr. Joachim Wächter
Daniel Beiter
Markus Morgner

12. Mai 2017

Inhaltsverzeichnis

1. Einleitung

TERENO (TERrestrial ENvironmental Observatories) beschreibt nach [Ter] eine Messinfrastruktur, um die globalen Folgen menschlicher Eingriffe in die Umwelt auf regionaler Ebene abzubilden. Die auf diese Weise gewonnenen Daten sollen genutzt werden, um der Menschheit zu zeigen, wie sie auf bestehende Veränderungen reagieren kann.

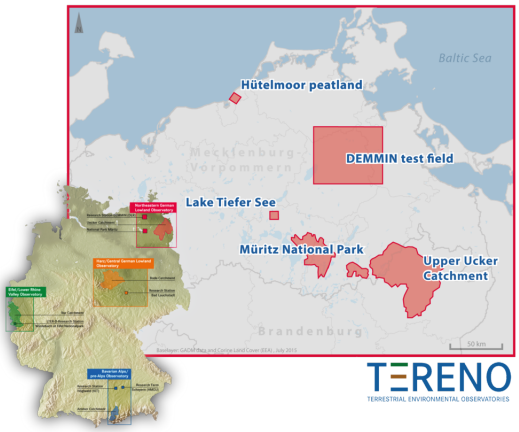

Abbildung 1.1.: TERENO-Messgebiete, entnommen aus [GFZ16b]

Die jeweiligen Messgebiete liegen in den Zuständigkeitsbereichen unterschiedlicher Zentren der Helmholtz-Gemeinschaft. So ist das *Observatorium Nordostdeutsches Tiefland* dem *Geologischen Forschungszentrum* (GFZ) in Potsdam zugeordnet. Als Aufgabenstellung sei der Müritz-Nationalpark gegeben. Dort befinden sich in einem bewaldeten Gebiet zwischen den Seen Hinnensee und Güsterpohl vier Messtationen mit einem maximalen Abstand von 200m zueinander. Es ist geplant, von diesen Stationen auf energieeffiziente und kostengünstige Weise Daten zu einem weiteren Zentralknoten zu senden. Von diesem Knoten aus kann per Mobilfunk zur Datenweiterleitung eine Verbindung zum Internet hergestellt werden. Das insgesamte, maximale Datenaufkommen pro Tag betrage geschätzt 50kBytes. Kabel werden aus sowohl aus ökologischen als auch aus Kostengründen nicht verlegt.

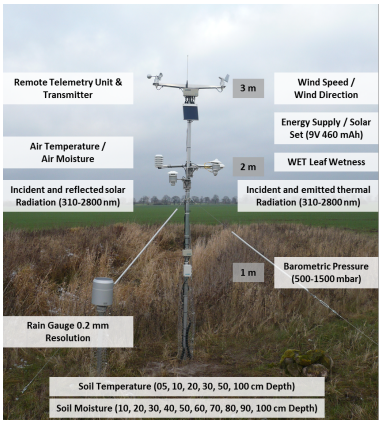

Abbildung 1.2.: Beispiel-Messstation, entnommen aus [GFZ16a]

2. Grundlagen

Die Grundlagen zur Datenübertragung von kleinen Datenmengen über größere Entfernungen und in bewaldeten Gebieten wurden in der Bachelor-Arbeit von Marcus Wolfshöfer [Wol16] beschrieben. Ausgehend vom *Internet of Things* (IoT) beschreibt er Geräte, die weniger Rechenleistung als ein herkömmlicher Computer haben, mit Batterie betrieben werden und kostengünstig in Anschaffung und Betrieb sind. Verbunden sind diese Geräte über ein Low Power Wide Area Network (LPWAN), einem Funknetzwerk. Er zitiert hierzu [Egl15] und [Pha15]. Als eingesetztes Protokoll für Funktechniken von mehr als 100m ist wegen der technologischen Verbreitung durch die Firmen *Semtech* und davon ausgehend *HopeRF* das *Long-Range*(LoRa)-Protokoll hervorzuheben ([Wol16], S.14). Erwähnt wurde auch das *SIGFOX*-Protokoll, ein sehr schmalbandiges Protokoll, das jedoch nur einen Durchsatz von 100 Bits pro Sekunde erlaubt und wegen der geringen Datendate ungeeignet für die gegebene Aufgabenstellung ist.

Bei der Auswahl der geeigneten Frequenzen zur drahtlosen Datenübertragung gilt nach [Wol16], S.7, die Friis-Übertragungsgleichung für die Ausbreitung im freien Raum:

$$P_{RX} = P_{TX} \cdot G_{RX} \cdot G_{TX} \cdot \left(\frac{\lambda}{4\pi R}\right)^2$$

mit:
P_{RX} empfangene Leistung in W
P_{TX} ausgesendete Leistung in W
G_{RX} Gewinn der Empfangsantenne, dimensionslos
G_{TX} Gewinn der Sendeantenne, dimensionslos
λ Wellenlänge in m
R Distanz zwischen Sender und Empfänger in m

Diese Gleichung drückt aus, dass bei gleicher Sendeleistung, gleichem Abstand zwischen Sender und Empfänger und bei Antennen mit gleichem Antennengewinn mehr Leistung beim Empfänger ankommt, je niedriger die Übertragungsfrequenz ist. Dem gegenüber steht die Baugröße der Antennen, die nach [Wol16], S.5, mit sinkender Frequenz zunehmen muss. Grundlage hierfür ist ein Halbwellendipol, der etwa die Hälfte der Wellenlänge bemisst. Diese berechnet sich nach folgender Formel:

$$\lambda = \frac{c}{f}$$

mit:
λ Wellenlänge in m
c Ausbreitungsgeschwindigkeit im Medium in $\frac{m}{s}$, in Luft: c=299792458$\frac{m}{s}$
f Frequenz in Hz

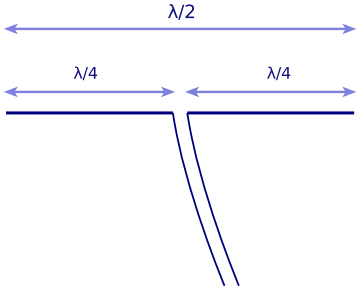

Abbildung 2.1.: Halbwellendipol, modifizierte Abbildung aus [dip]

Da sich nach Marcus Wolfshöfer im Bereich von einem Meter bis zu wenigen Zentimetern kostengünstige und effiziente Antennen leicht herstellen lassen, empfiehlt er auf S.6 den Frequenzbereich VHF (30-300MHz) und UHF(300-3000MHz). Bei der Betrachtung der Ausbreitung von Radiowellen im Wald werden im von ihm zitierten [Yu 09] in einem Testgebiet in Singapur mit 240 MHz kleinere Ausbreitungsverluste erzielt als mit 700 MHz. Seine eigenen Messungen im überwiegend ländlichen Bereich zeigen die Reichweitenvorteile von 169MHz gegenüber 868MHz. Ein anderer Aspekt ist die Polarisation: Lineare vertikale Polarisation bedeutet bei einem Dipol, dass er senkrecht zur Erdoberfläche steht, während er bei linearer horizontaler Polarisation parallel zum Erdboden liegt.

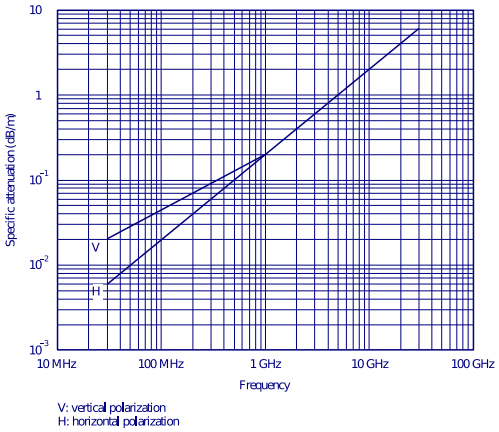

Abbildung 2.2.: Spezifische Dämpfung im Wald, entnommen aus [Int16a], S.3

Abbildung 2.2 zeigt die spezifische Dämpfung abgeleitet aus verschiedenen Messungen in Waldgebieten von 30 MHz bis 30 GHz. Demnach ist bei Frequenzen unterhalb einem GHz die Dämpfung im Wald bei horizontaler Polarisation geringer als bei vertikaler.

2.1. Rechtliches

In [Wol16], S.8-10 und sich auf [Bun14] beziehend, sind für ein LPWAN die Frequenzbereiche für *Short Range Devices* (SRD) um 169MHz, 433MHz und 868MHz geeignet. Bei 169,4MHz-169,475MHz und 869,400-869,650 MHz darf mit bis zu 500mW E.R.P. gesendet werden. E.R.P. steht für Effective Radiated Power oder auch effektive Strahlungsleistung und bedeutet das Produkt aus effektiver Sendeleistung und dem Antennengewinn gegenüber einem Halbwellendipol bei gegebener Richtung ([Int16b]). Neben der maximalen Strahlungsleistung ist in den Bereichen um 169MHz und 868MHz auf den maximalen Arbeitszyklus bzw. auf die maximale relative Sendezeit bezogen auf eine Stunde zu achten. Sollten die Vorschriften für SRD für ein Projekt zu restriktiv sein, können Frequenzen entsprechend der *Verwaltungsvorschriften für Frequenzzuteilungen im nichtöffentlichen mobilen Landfunk (VVnömL)* ([Bun16]) genutzt werden. Je nach Anwendungsfall sind auch höhere Leistungen als 500mW E.R.P erlaubt. Hierfür ist ein Antrag bei einer lokalen Außenstelle der Bundesnetzagentur zu stellen und es sind die dabei entstehenden Kosten zu berücksichtigen ([Bun16], Kapitel 4, S.14 und [Bun]).

3. Versuchsaufbau

3.1. Hardware

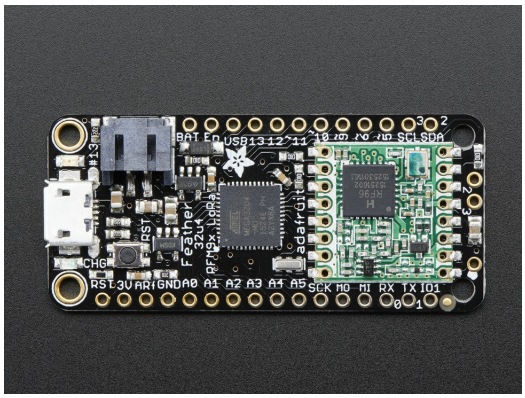

Abbildung 3.1.: Adafruit Feather für 868 oder 915MHz, Bild entnommen aus [Ada17]

Beim Aufbau eines Testsystems innerhalb der erlaubten Frequenzen für SRD wäre aus Effizienzgründen der Frequenzbereich um 169MHz zu bevorzugen. Allerdings sollen die eingesetzten Geräte kostengünstig in der Anschaffung, leicht zu beschaffen und in Betrieb zu nehmen sein. Es gibt zwar ein Sende- und Empfangsmodul der Firma Telit [Tel] für 169MHz zu kaufen. Dieses ist jedoch wegen Verwendung des Modbus-Protokolls zu spezialisiert. Erwähnt sei auch ein Sende- und Empfangsmodul der Firma HopeRF *RFM98PW-169S* für 169 MHz [Hop]. Dieses Modul ist auf eine ebenfalls von dieser Firma erhältliche Platine zu löten und an ein Arduino-Modul anzuschließen. Der hierfür höhere Aufwand würde jedoch die Grenzen dieses Projekts sprengen. Übrig bleiben 433MHz und 868MHz. Für 433MHz sind Funkmodule leicht zu beschaffen, allerdings ist die in Deutschland maximal erlaubte Leistung für SRD auf 10mW E.R.P. beschränkt, was die Testmöglichkeiten stark einschränkt. Bei 868MHz haben die Antennen im Gegensatz zu den anderen genannten SRD-Bereichen die kleinste Baugröße und die schlechteren physikalischen Ausbreitungsbedingungen können durch Richtfunkantennen oder eine höhere Sendeleistung kompensiert werden. Dieses Projekt dient jedoch nur der allgemeinen Abschätzung.

Die Wahl zum Aufbau eines Testsystems fiel auf den *Adafruit Feather 32u4 RFM95 LoRa Radio - 868 or 915 MHz*. Hier sind Funkmodul und Arduino auf einer Platine integriert. Wenn der elektrische Strom abgeschaltet wird, behält dieses Modul sein Programm. Dieses System lässt sich zudem mit einfachen Mitteln in Betrieb nehmen und testen. Neben zwei solcher Module wurden noch folgende Komponenten gekauft:

- 2 uFL-SMT-Antennenstecker (Connector)
- 2 RP-SMA-nach-uFL-Adapterkabel
- 2 RP-SMA-Stabantennen für 868 MHz mit Knickgelenk, 4,5 dBi, Länge: 23cm
- 2 USB-2.0-A-Stecker-auf-Micro-USB-B-Stecker-Verbindungskabel
- 2 LiPo-Akkus, 2000mAh, 3,7V mit Kabel und 2-mm-JST-Stecker
- 2 Sidekicks mit zahlreichem Zubehör wie Steckbrettern für weitere Versuche

3.2. Setup

Auf [Ada17] unter Antenna Options befindet sich eine Anleitung, wie der uFL-Antennenstecker anzulöten ist. Ausgehend vom entsprechenden Datenblatt ([Dat]) muss *NOTCH* beim *Feather* nach außen zeigen, da sonst die innere Leitung des Steckers keine Verbindung hat.
Auf https://www.arduino.cc/en/main/software kann die jüngste *Arduino IDE* heruntergeladen werden. In [Ada17] unter Arduino IDE Setup und nachfolgenden Links ist beschrieben, wie die Boards, die Windows-Treiber und die *RadioHead*-Bibliothek zu installieren sind. Unter *Tools* oder *Werkzeuge* in der *Arduino IDE* muss unter *Boards* das *Adafruit Feather 32u4* ausgewählt sein. Nun kann ein Programm implementiert und per USB hochgeladen werden. Mit dem seriellen Monitor unter *Tools* oder *Werkzeuge* kann die Ausgabe der seriellen Schnittstelle überwacht werden.

3.3. Programmierung

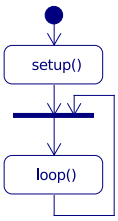

Abbildung 3.2.: Aktivitätsdiagramm eines Ardiunos

Abbildung 3.2 zeigt den Ablauf eines Arduino-Programms. Zuerst wird er nach dem Einschalten mit *setup()* der Arduino initialisiert und anschließend befindet er sich mit dem in *loop()* ausgeführten Code in einer Endlosschleife. Für das Testsystem werden zwei Arduinos benötigt. Ein Arduino dient als Client, kommuniziert per Funk mit dem Server und gibt über die serielle Schnittstelle, welche per USB mit einem Laptop verbunden ist, Daten aus. Der dazugehörige Code befindet sich in Anhang A und B. Mit Hilfe der Klasse *RHReliableDatagram* aus der *RadioHead*-Bibliothek wird ein verbindungsorientiertes Protokoll verwendet, mit dem auch Adressierung möglich ist. So ist dem Client die Adresse 1 und dem Server die Adresse 2 zugeordnet.

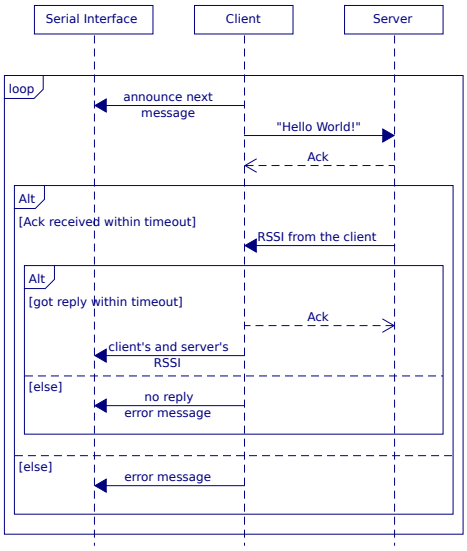

Abbildung 3.3.: Sequenzdiagramm

In Abbildung 3.3 kündigt der Client über das *Serial Interface* eine neue Nachrichtenübertragung an und schickt ein ”Hello World” an den Server. Dieser bestätigt mit *Ack*, liest aus der empfangenen Nachricht die Signalstärke

RSSI (Received Signal Strength Indication) aus und sendet sie in Form einer neuen Nachricht zurück an den Client. Der Client gibt diese Nachricht über die serielle Schnittstelle aus und zusätzlich das aus dem Server-Signal ermittelte RSSI. Sollte der Empfang gestört sein, gibt der Client Fehlermeldungen aus.

3.4. Tests

Abbildung 3.4.: Messstationen, verwendetes Bildmaterial: Google Earth und ©2009 GeoBasis-DE/BKG

Abbildung 3.4 zeigt das Messgebiet. Im Norden befindet sich der See Güsterpohl und im Osten der Hinnensee. Der eingezeichnete Knoten *K5* stellt den Zentralknoten mit Mobilfunkanschluss dar, zu dem die anderen Stationen ihre Daten senden. Folgende Koordinaten sind gegeben:

- K5: Zone: 33 U, Rechtswert: 379165.00 m E, Höchstwert: 5910260.00 m N
- SM_PB3: Zone: 33 U, Rechtswert: 379291.00 m E, Höchstwert: 5910276.00 m N, ungefährer Abstand zu K5: 127m (Luftlinie)
- SM_PB4: Zone: 33 U, Rechtswert: 379255.00 m E, Höchstwert: 5910282.00 m N, ungefährer Abstand zu K5: 92m (Luftlinie)
- SM_PB5: Zone: 33 U, Rechtswert: 379233.00 m E, Höchstwert: 5910180.00 m N, ungefährer Abstand zu K5: 106m (Luftlinie)
- SM_B: Zone: 33 U, Rechtswert: 379123.00 m E, Höchstwert: 5910182.00 m N, ungefährer Abstand zu K5: 98m (Luftlinie)

(a) Server (b) Client

Abbildung 3.5.: Versuchsaufbau, aufgenommen von Markus Morgner am 04.04.2017

In der Versuchsanordnung nach Abbildung 3.5 ist am Server-Arduino zur Stromversorgung ein Akku angeschlossen. Die Daten werden über die Antenne übertragen, die mit einem Kabelbinder auf einer Holzstütze des 2. Rohrs der Regenbestandsrinne K5 befestigt ist. Der Antennenfuß befand sich in einer Höhe von 1,80m über dem Erdboden. Der Client ist an einem Laptop angeschlossen und seine Antenne ist mit einem Kabelbinder an einem Stock befestigt. Auf diese Weise sollte beim Testen ein ausreichend großer Abstand zwischen Mensch und Antenne hergestellt sein, um das Signal möglichst wenig zu beeinflussen. Mit dem Client und am Laptop über das Terminal-Programm wurde an den Außenknoten der geplanten Sterntopologie das RSSI gemessen und dabei die Antenne so ausgerichtet, um optimale Ergebnisse zu erhalten. Im Gegensatz zur Abbildung wurde die Server-Antenne um 45° nach unten abgewinkelt, da sonst das Signal zu schwach gewesen wäre. Für Messungen an den Station SM_PB3 und SM_PB4 hatte diese Antenne ausgehend von Norden einen Azimuth von 65° und bei den anderen beiden Stationen einen Azimuth von 198°. K5 befindet sich auf einer mehrere Meter großen Anhöhe und zu den Stationen SM_PB3, SM_PB4 und SM_PB5 gäbe es auch ohne Vegetation keinen Sichtkontakt.

Station	Winkel zur Horizontalen	Höhe des Antennenfußes über dem Erdboden	Azimuth	RSSI	Wert bei setTXpower
SM_PB4	25°	1,30m	98°	-61...-64	23
SM_PB3	10°	1,35m	96°	-67...-71	23
SM_B	45°	1,80m	45°	-57...-60	23
SM_PB5	13°	1,69m	119°	-69...-74	23
SM_PB4	45°	1,68m	72°	-68...-74	14
SM_PB3	65°	1,90m	114°	-72...-76	14
SM_B	60°	1,80m	190°	-61...-64	14
SM_PB5	45°	1,80m	150°	-72...-80	14
SM_PB4	25°	1,68m	66°	-75...-80	10
SM_PB3	10°	1,32m	93°	-80...-84	10
SM_B	35°	1,85m	190°	-68...-74	10
SM_PB5	30°	1,65m	170°	-76...-80	10

Tabelle 3.1.: Versuchsaufbau

Der Leistungspegel in Bezug auf 1 Milliwatt (mW) in Dezibel Milliwatt (dBm) kann durch Veränderung des ersten Parameters der Methode

```
rf95.setTxPower(10, false);
```

gesetzt werden (siehe Anhang A und B). Die auf diese Weise modifizierten Programme wurden über den Laptop auf den Client und Server hochgeladen und anschließend die Ausgaben des Clients am Laptop überwacht. Die entsprechende Sendeleistung in Milliwatt ergibt sich nach folgender Formel [Pro]:

$$P_{mW} = 10^{\left(\frac{P_{dBm}}{10}\right)} \cdot 1mW$$

Demnach ergeben sich für 20dBm 100mW, für 14dBm ungefähr 25mW und für 10dBm 10 mW. Davon abweichend entspricht ein Wert von 23 in setTxPower der maximalen Sendeleistung von 100mW. Bei Werten ≤ 20 sollen nach Mike McCauley, dem Autor der *RadioHead*-Library, die eingesetzten Werte dem tatsächlichen dBm-Wert entsprechen. Im Quellcode ist hierzu in der Klasse *RH_RF95* als Kommentar eine Widersprüchlichkeit zwischen Dokumentation der HopeRF-Chips und Messungen der Sendeleistung vermerkt. Wie diese Messungen zustande kamen, ist nicht dokumentiert. *Adafruit* verwies hierzu an die Mail-Liste für die *RadioHead*-Bibliothek, da diese Firma nur die Hardware bereitstellt. Eine entsprechende Anfrage ergab, dass genaue Werte z.B. mit einem Spektrumanalysator (englisch: RF spectrum analyzer) selbst gemessen werden müssten. Von *HopeRF*, das die Funkmodule für *Adafruit* produziert, kam eine Erklärung nach einer E-Mail-Anfrage, wie der Leistungspegel zu berechnen sei. So, wie die *RadioHead*-Bibiliothek implementiert ist und unter der Annahme, die Leistungswerte von *HopeRF* seien korrekt, wäre bei einem Wert von 14 in *setTxPower* mit 11dBm oder ungefähr 12,6mW und bei einem Wert von 10 mit 7dBm oder ungefähr 5mW gesendet worden.
Laut Herstellerangaben des *Adafruit Feather* kann RSSI einen Wert von ungefähr -15 für ein sehr starkes bis -100 für ein sehr schwaches Signal annehmen. Wenn beide Arduinos und ihre parallel zueinander ausgerichteten Antennen dicht beieinander lagen, ergab RSSI bei allen drei Sendestärken einen besten Wert bei -22 und bei einem RSSI von -103 sind noch Pakete angekommen.

3.5. Auswertung und Ausblick

Trotz gewisser Unklarheiten in Bezug auf die tatsächliche Sendeleistung konnte gezeigt werden, dass im gegebenen Testgebiet eine energieeffiziente Datenübertragung möglich ist. Dass die Daten trotz der Anhöhe ankamen, lässt sich mit dem Phänomen der Beugung erklären. Die Server-Antenne musste abgewinkelt werden, da der angegebene Antennengewinn von 4,5dBi eine Bündelung in der Vertikalen und einen kleineren vertikalen Abstrahlwinkel bedeutet, auch wenn in der Horizontalen ein Rundstrahlverhalten vorliegt. Sollte beim Zentralknoten nur eine Antenne nicht ausreichen, hätte man mit dem Zukauf eines *Adafruit FeatherWings* einen weiteren Antennenanschluss, so dass bei zwei Antennen entweder über die eine oder die andere gesendet werden kann:

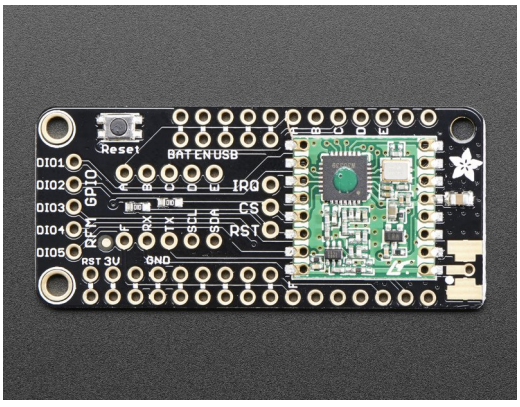

Abbildung 3.6.: Adafruit FeatherWing, entnommen aus [Ada16]

Des Weiteren sind Bandbreitenmessungen der Nettobandbreite nötig, da in der Klasse *RHReliableDatagram* Pakete wiederholt gesendet werden, wenn innerhalb gewisser Timeouts keine Quittung (Ack) ankommt.
Zur Erhöhung der Energieeffizienz kann mit Richtfunkantennen [Swe05] gearbeitet werden:

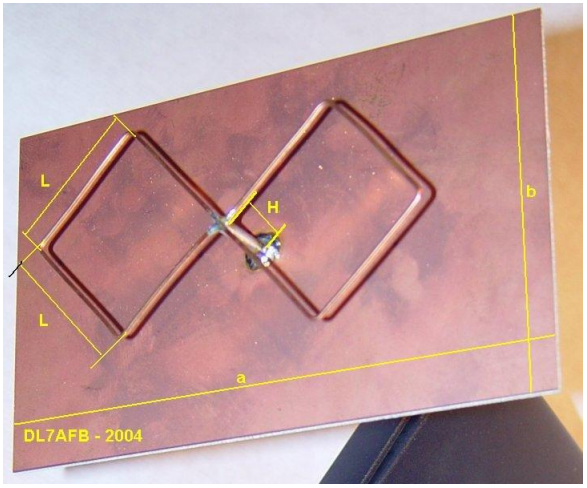

Abbildung 3.7.: Eigenbau-Biquad-Antenne mit Reflektor, entnommen aus [Bod09]

Dadurch kann die Sendeleistung weiter reduziert werden. Aus Kostengründen kann hier über einen möglichen Eigenbau nachgedacht werden. Zudem kann durch die Veränderung der Parameter wie *Spreading Factor* und *Bandwith* der LoRa-Modulation die Bandbreite zu Lasten der Leistungsübertragungsbilanz (englisch: Link budget) erhöht werden:

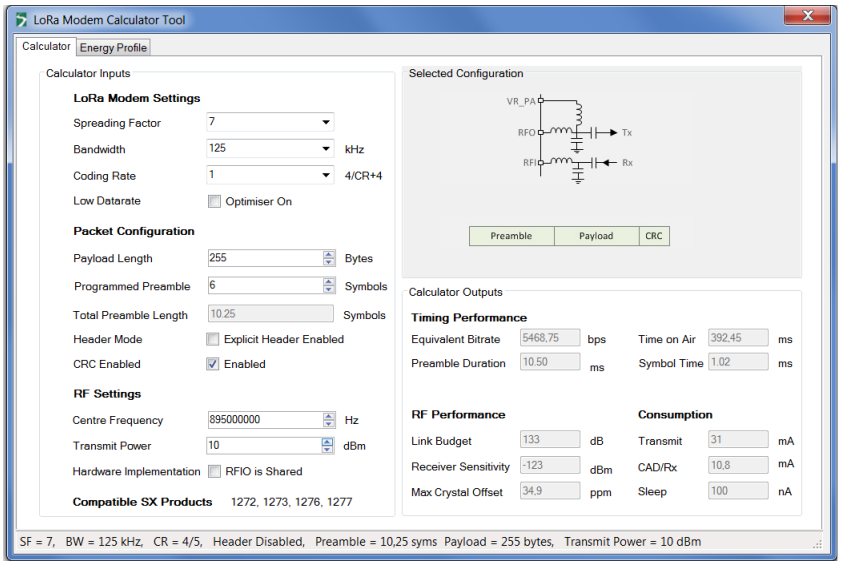

Abbildung 3.8.: Lora Modem Calculator Tool von Semtech, Download: [too]

Somit bräuchte man für die Datenübertragung weniger Zeit und bei gleicher oder reduzierter Sendeleistung würde somit der Akku geschont werden. Als weitere Optimierung bei niedrigen Sendeleistungen ausgehend vom aktuellen Wissensstand kann wegen besserer physikalischer Ausbreitungseigenschaften der Bereich um 433MHz in Betracht gezogen werden.

Quellverzeichnis

[Ada16] Adafruit. *Radio FeatherWing*, August 2016. URL: https://learn.adafruit.com/radio-featherwing. Abgerufen am 07.05.2017.

[Ada17] Adafruit. *Adafruit Feather 32u4 with LoRa Radio Module*, Mai 2017. URL: https://learn.adafruit.com/adafruit-feather-32u4-radio-with-lora-radio-module. Abgerufen am 23.04.2017.

[Bod09] Bodo Woyde, DL7AFB. *Externe Antennen verbessern Reichweite bei DECT und WLAN*, Oktober 2009. URL: http://dl7afb.darc.de/projects/DECT-WIFI-Antennas.htm. Abgerufen am 01.05.2017.

[Bun] Bundesministerium der Justitz und für Verbraucherschutz. *Frequenzgebührenverordnung (FGebV) Anlage (zu § 1 Abs. 1)*. URL: http://www.gesetze-im-internet.de/fgebv/anlage.html. Abgerufen am 07.05.2017.

[Bun14] Bundesnetzagentur. *Allgemeinzuteilung von Frequenzen zur Nutzung durch Funkanwendungen mit geringer Reichweite für nicht näher spezifizierte Anwendungen; Non-specific Short Range Devices (SRD)*. Vfg 69/2014, 2014. URL: https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Allgemeinzuteilungen/2014_69_SRD_pdf.pdf?__blob=publicationFile&v=1. Abgerufen am 23.04.2017.

[Bun16] Bundesnetzagentur. *Verwaltungsvorschriften für Frequenzzuteilungen im nichtöffentlichen mobilen Landfunk (VVnömL)*, November 2016. URL: https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Verwaltungsvorschriften/VVn%C3%B6mL.pdf?__blob=publicationFile&v=5. Abgerufen am 23.04.2016.

[Dat] Datenblatt des uFL-Antennensteckers. URL: https://cdn-shop.adafruit.com/datasheets/734120114_sd.pdf. Abgerufen am 23.04.2017.

[dip] *FM Dipole Antenna Design*. URL: https://www.electronics-notes.com/articles/antennas-propagation/dipole-antenna/fm-dipole-antenna.php. Abgerufen am 22.04.2017.

[Egl15] Peter R. Egli. *Overview of emerging technologies for low power wide area networks in internet of things and M2M scenarios*, 2015. URL: http://www.indigoo.com/dox/itdp/12_MobileWireless/LPWAN.pdf. Abgerufen am 17.04.2017.

[GFZ16a] GFZ. *Messinfrastruktur*, April 2016. URL: http://www.gfz-potsdam.de/sektion/fernerkundung/projekte/tereno/messinfrastruktur/. Abgerufen am 17.04.2017.

[GFZ16b] GFZ. *Observatorium Nordostdeutsches Tiefland*, September 2016. URL: http://www.gfz-potsdam.de/de/wissenschaftliche-infrastruktur/observatorien/regionale-observatorien/tereno/tereno-nord-ost/. Abgerufen am 30.04.2017.

[GFZ16c] GFZ. *TERENO - Zielstellungen*, September 2016. URL: http://www.gfz-potsdam.de/sektion/fernerkundung/projekte/tereno/. Abgerufen am 15.04.2017.

[Hop] HopeRF. RFM98PW-Modul. URL: http://www.hoperf.com/rf_transceiver/Enhanced_Power/RFM98PW.html. Abgerufen am 23.04.2017.

[Int16a] International Telecommunication Union (ITU). *Attenuation in vegetation*. P Series Radiowave propagation, September 2016. URL: https://www.itu.int/rec/R-REC-P.833-9-201609-I/en. Abgerufen am 22.04.2017.

[Int16b] International Telecommunication Union (ITU). *Radio Regulations*,Volume 1, Chapter 1, Section VI, no. 1.162, November 2016. URL: http://www.itu.int/pub/R-REG-RR-2016. Abgerufen am 23.04.2016.

[Ist99] István Z. Kovács, Patrick C. F. Eggers, Kim Olesen. *Radio channel characterisation for forest environments in the VHF and UHF frequency bands.* IEEE 50th Vehicular Technology Conference, Amsterdam, The Netherlands, pp. 1387-1391, September 1999. URL: https://www.researchgate.net/profile/K_Olesen/publication/3819757_Radio_channel_characterisation_for_forest_environments_in_the_VHF_and_UHF_frequency_bands/links/53eb04c60cf2fb1b9b6ad1bf.pdf, doi:10.1109/VETECF.1999.801490. Abgerufen am 22.04.2017.

[Pha15] Prof. Congduc Pham. *Low-power, Long-range WAN for IoT: A technology overview*, Januar 2015. URL: http://cpham.perso.univ-pau.fr/Paper/Talk-Rescom-16-LPWAN-review.pdf. Abgerufen am 17.04.2017.

[Pro] Prof. Dr. rer. nat. Achim Enders. *Grundlagenlabor - Elektromagnetische Verträglichkeit.* URL: https://www.tu-braunschweig.de/Medien-DB/emv/grundlagenlabor.pdf. Abgerufen am 09.05.2017.

[Sus] Susanne Schlötzer, Tobias Renk. *Grundlagen der Antennentheorie.* URL: http://dk0te.dhbw-ravensburg.de/studienarbeit-4nec2/downloads/antennentheorie.pdf. Abgerufen am 22.04.2017.

[Swe05] Björn Swendrak. *Entwurf und Simulation einer WLAN-Gruppenantenne*, Januar 2005. URL: http://www.swendrak.de/docs/wlan_gruppenantenne.pdf. Abgerufen am 30.04.2017.

[Tel] Telit. Seite des ME70-169-Moduls. URL: http://www.telit.com/sr-rf/me70-169/. Abgerufen am 23.04.2017.

[Ter] Startseite von TERENO. URL: http://teodoor.icg.kfa-juelich.de/overview-de. Abgerufen am 17.04.2017.

[too] Lora modem calculator tool von semtech. URL: http://www.semtech.com/apps/filedown/down.php?file=SX1272LoRaCalculatorSetup1%271.zip. Abgerufen am 30.4.2017.

[Wol16] Marcus Wolfshöfer. *Aufbau und Optimierung eines LoRa-IoT-Systemes mittels messtechnischer Analysen und Simulationen*, Dezember 2016. URL: http://www.ennet-gmbh.de/wp-content/uploads/2017/03/Thesis.pdf. Abgerufen am 17.04.2017.

[Yu 09] Yu Song Meng, Yee Hui Lee, Boon Chong Ng. *Empirical Near Ground Path Loss Modeling in a Forest at VHF and UHF Bands*, Mai 2009. URL: http://www.ntu.edu.sg/home/eyhlee/Prof%20Lee/2009%20TAP-proofread.pdf, doi:10.1109/TAP.2009.2016703. Abgerufen am 22.04.2017.

A. Code des Clients

```
#include <RHReliableDatagram.h>
#include <SPI.h>
#include <RH_RF95.h>

/* for feather32u4 */
#define RFM95_CS 8
#define RFM95_RST 4
#define RFM95_INT 7

// Change to 434.0 or other frequency, must match RX's freq!
#define RF95_FREQ 869.5

// Singleton instance of the radio driver
RH_RF95 rf95(RFM95_CS, RFM95_INT);

// Blinky on receipt
#define LED 13

#define CLIENT_ADDRESS 1
#define SERVER_ADDRESS 2

RHReliableDatagram manager(rf95, CLIENT_ADDRESS);

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);

  while (!Serial);
  Serial.begin(9600);
  delay(100);

  Serial.println("Feather LoRa RX Test!");

  // manual reset
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);
  // Defaults after init are 434.0MHz, 13dBm, Bw = 125 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC on

  // The default transmitter power is 13dBm, using PA_BOOST.
  // If you are using RFM95/96/97/98 modules which uses the PA_BOOST transmitter pin, then
  // you can set transmitter powers from 5 to 23 dBm:
  if (!manager.init()){
    Serial.println("init failed");
    while(1);
  }

  Serial.println("LoRa radio init OK!");

  // Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM
```

```
  if (!rf95.setFrequency(RF95_FREQ)) {
    Serial.println("setFrequency failed");
    while (1);
  }
  Serial.print("Set Freq to: "); Serial.println(RF95_FREQ);


  rf95.setTxPower(10, false);
}

uint8_t data[] = "Hello World!";
// Dont put this on the stack:
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];

void loop() {
   Serial.println("Sending to serial_reliable_datagram_server");

  // Send a message to manager_server
  if (manager.sendtoWait(data, sizeof(data), SERVER_ADDRESS))
  {
     digitalWrite(LED, HIGH);
    // Now wait for a reply from the server
    uint8_t len = sizeof(buf);
    uint8_t from;
    if (manager.recvfromAckTimeout(buf, &len, 2000, &from))
    {
      Serial.print("got reply from : 0x");
      Serial.print(from, HEX);
      Serial.print(": ");
      Serial.println((char*)buf);
      Serial.print("RSSI from the server: ");
      Serial.println(rf95.lastRssi(), DEC);
    }
    else
    {
      Serial.println("No reply, is serial_reliable_datagram_server running?");
    }
     digitalWrite(LED, LOW);
  }
  else
    Serial.println("sendtoWait failed");
  delay(500);
}
```

B. Code des Servers

```
#include <RHReliableDatagram.h>
#include <SPI.h>
#include <RH_RF95.h>

/* for feather32u4 */
#define RFM95_CS 8
#define RFM95_RST 4
#define RFM95_INT 7

// Change to 434.0 or other frequency, must match RX's freq!
#define RF95_FREQ 869.5

// Singleton instance of the radio driver
RH_RF95 rf95(RFM95_CS, RFM95_INT);

// Blinky on receipt
#define LED 13

#define CLIENT_ADDRESS 1
#define SERVER_ADDRESS 2

RHReliableDatagram manager(rf95, SERVER_ADDRESS);

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(RFM95_RST, OUTPUT);
  digitalWrite(RFM95_RST, HIGH);
  delay(100);

  // manual reset
  digitalWrite(RFM95_RST, LOW);
  delay(10);
  digitalWrite(RFM95_RST, HIGH);
  delay(10);

  // Defaults after init are 434.0MHz, 13dBm, Bw = 125 kHz, Cr = 4/5, Sf = 128chips/symbol, CRC on

  // The default transmitter power is 13dBm, using PA_BOOST.
  // If you are using RFM95/96/97/98 modules which uses the PA_BOOST transmitter pin, then
  // you can set transmitter powers from 5 to 23 dBm:
  if (!manager.init()){
    digitalWrite(LED, HIGH);
    while(1);
  }
  // Defaults after init are 434.0MHz, modulation GFSK_Rb250Fd250, +13dbM
  if (!rf95.setFrequency(RF95_FREQ)) {
     digitalWrite(LED, HIGH);
    while (1);
  }
  digitalWrite(LED, LOW);
  rf95.setTxPower(10, false);
}
```

```
// Dont put this on the stack:
uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
void loop()
{
  if (manager.available())
  {
    // Wait for a message addressed to us from the client
    uint8_t len = sizeof(buf);
    uint8_t from;
    if (manager.recvfromAck(buf, &len, &from))
    {
      digitalWrite(LED, HIGH);
      uint8_t data[26] = "RSSI from the client:     ";
      itoa(rf95.lastRssi(), data+22, 10);
      // Send a reply back to the originator client
      manager.sendtoWait(data, sizeof(data), from);
      digitalWrite(LED, LOW);
    }

  }
}
```