



Eine Plattform zur methodischen Vermessung und Modellierung der Funkausbreitung in einem Experimentierszenario

Bachelorarbeit

von

Markus Kerper

aus

Düsseldorf

vorgelegt am 31.10.2006

Lehrstuhl für Rechnernetze und Kommunikationssysteme

Prof. Dr. Martin Mauve

Heinrich-Heine-Universität Düsseldorf

Oktober 2006

Betreuer:

Dipl. Wirtsch.-Inf. Wolfgang Kiess

Dipl. Wirtsch.-Inf. Christian Lochert

Danksagung

Mein ganz besonderer Dank gilt meinen beiden Betreuern Wolfgang Kiess und Christian Lochert, die immer wieder Tipps, Ideen und Anregungen für mich hatten.

Weiterhin möchte ich Stephan Zalewski für die Installation des Linux Systems und der benötigten Software auf den Endgeräten bedanken.

Großer Dank gilt auch allen Teilnehmern der Experimente, die nicht nur die Messungen durchführten, sondern mich auch mit Ideen und Vorschlägen zur Verbesserung versorgten.

Inhaltsverzeichnis

| | |
|--|------------|
| Abbildungsverzeichnis | vii |
| Tabellenverzeichnis | ix |
| 1 Einleitung | 1 |
| 1.1 Motivation | 1 |
| 1.2 Problemstellung | 2 |
| 1.3 Struktur der Arbeit | 2 |
| 2 Grundlagen | 3 |
| 2.1 GIMP ToolKit | 3 |
| 2.2 Zaurus mit GPE | 3 |
| 3 Toolkit | 5 |
| 3.1 Landkarte | 5 |
| 3.2 Mapconfig | 6 |
| 3.2.1 Raster | 6 |
| 3.2.2 Messpunkte | 7 |
| 3.2.3 Stützstellen | 8 |
| 3.2.4 Platzierung der Geräte | 8 |
| 3.3 Mapkit | 12 |
| 3.3.1 Positionierungsdienst | 13 |
| 3.3.2 Messung | 13 |
| 3.3.3 Steuerung des Experimentes | 14 |
| 3.4 Bibliothek | 16 |
| 3.4.1 Algorithmus | 16 |
| 3.4.2 Schnittstellenbeschreibung | 18 |
| 3.4.3 Analyzer | 19 |

| | |
|--|-----------|
| 4 Vermessung und Ergebnisse | 21 |
| 4.1 Konfiguration | 21 |
| 4.2 Durchführung | 22 |
| 4.3 Ergebnisse | 23 |
| 4.4 Validierung der interpolierten Messwerte | 25 |
| 5 Zusammenfassung und Ausblick | 29 |
| 5.1 Zusammenfassung | 29 |
| 5.2 Ausblick | 30 |
| Literaturverzeichnis | 31 |

Abbildungsverzeichnis

| | | |
|-----|---|----|
| 3.1 | Beispiel für Platzierung der Messpunkte | 7 |
| 3.2 | Ablauf einer Vermessung | 15 |
| 3.3 | Beispiel für die Punkt Interpolation mit $n = 2$ | 17 |
| 3.4 | Wahl der Messpunkte bei der Punkt Interpolation mit $n = 4$ | 18 |
| 4.1 | Linkqualitäten von einem Messpunkt aus | 24 |
| 4.2 | Interpolierte Linkqualität eines Punktes | 25 |
| 4.3 | Analyse der Interpolationsverfahren | 26 |
| 4.4 | Fehlerhafte berechnete Linkqualitäten | 27 |

Tabellenverzeichnis

| | | |
|-----|---|----|
| 2.1 | Sharp Zaurus SL-6000 Hardware | 4 |
| 3.1 | Positionierung der Messgeräte mit einer Stützstelle | 10 |
| 3.2 | Positionierung der Messgeräte mit zwei Stützstellen | 11 |
| 4.1 | Details der durchgeführten Messungen | 22 |
| 4.2 | Abweichung der interpolierten Linkqualitäten anhand von 100 Messungen | 27 |

Kapitel 1

Einleitung

1.1 Motivation

Im Forschungsbereich der mobilen ad-hoc Netzwerke (MANETs)¹ finden aktuell viele Entwicklungen von Anwendungen und Protokollen statt. Diese wurden bisher hauptsächlich simulativ untersucht. Den Simulationen liegen Modelle zur Funkausbreitung zugrunde, welche stark von der Realität abstrahieren und oft nicht ausreichend sind, um Aussagen über die Funktionsweise der Anwendung in der Realität zu machen. Einige Experimentierreihen zeigen, dass auch die gängigsten Modelle aufgrund von sehr schwer oder gar nicht modellierbaren realen Einflüssen in Experimenten schnell an ihre Grenzen kommen. [LNT02] [LYN⁺05]

Daher müssen reale Experimente oder Emulationen durchgeführt werden, um eine genauere Aussage über das Verhalten der Anwendung oder des Protokolls in der Realität zu machen. Aufgrund der realen nicht abschätzbaren Effekte der Funkausbreitung sind Kenntnisse über diese in dem Testgebiet sowohl für die Emulation, als auch für die Durchführung eines realen Experiments erforderlich. Daher ist es nötig eine neue Methode zu entwickeln, mit der es möglich ist die Funkausbreitung zur Vorbereitung der Tests zu vermessen und zu modellieren.

¹MANETs sind mobile sich selbstorganisierende Netzwerke ohne Infrastruktur, in denen jedes Endgerät Routeraufgaben übernimmt.

1.2 Problemstellung

In dieser Arbeit soll eine Plattform zur methodischen Vermessung und Modellierung der Funkausbreitung in einem Experimentierszenario entwickelt werden. Mit Hilfe dieser Plattform ist es möglich jedes beliebige Experimentiergebiet zu vermessen und zu modellieren. Das Ergebnis ist eine Bibliothek, die anhand von zwei Koordinaten die Linkqualität zwischen diesen Punkten berechnet. Diese Daten können dann zur Vorbereitung von realen Experimenten und Emulationen über eine bereitgestellte Schnittstelle genutzt werden.

Diese Bibliothek wird von Werkzeugen erstellt, welche in kurzer Zeit und mit wenig Aufwand zu einem vermessenen und berechneten Experimentiergebiet führen. Da es unmöglich ist jedes Koordinatenpaar zu vermessen, werden nur ausgewählte Paare vermessen und die dazwischenliegenden durch Interpolation berechnet.

Die Steuerung und die Koordination des Experimentes sollten weitestgehend automatisiert ablaufen, da die Abstimmung von mehreren sich bewegenden Geräten und Personen schwer ist.

1.3 Struktur der Arbeit

In Kapitel 2 werden die Grundlagen für die Entwicklung der Werkzeuge vorgestellt.

Kapitel 3 erläutert daraufhin den Umgang mit den Werkzeugen und deren Algorithmen. Es werden die Tools *Mapconfig* und *Mapkit* vorgestellt, welche für die Experimentvorbereitung und die Vermessung zuständig sind. Anschließend werden die Algorithmen und die Schnittstelle der *Bibliothek*, mit dem darauf aufbauenden *Analyzer*, vorgestellt.

Kapitel 4 beschreibt Konfiguration, Durchführung und Ergebnisse der Messungen. Anschließend findet eine Validierung der berechneten Ergebnisse statt.

Kapitel 5 beinhaltet eine Zusammenfassung der Arbeit und beschreibt danach mögliche Weiterentwicklungen der Tools, sowie die Anwendungsgebiete der Plattform.

Kapitel 2

Grundlagen

2.1 GIMP ToolKit

Das Gimp Toolkit (GTK+) [gtk06] ist eine Bibliothek, welche die Entwicklung einer graphischen Benutzeroberfläche (GUI) möglich macht. Sie stellt sogenannte Widgets, mit denen sich Textboxen, Labels, Buttons etc. erstellen lassen, bereit und ermöglicht so die Kommunikation mit dem User. GTK+ wird unter der GNU Lesser General Public License (LGPL) veröffentlicht, welche sowohl eine Nutzung der Bibliotheken in freier als auch in kommerzieller Software erlaubt. GTK+ ist in C entwickelt worden und stellt Schnittstellen zu Programmiersprachen wie C++, Java, Perl oder Python bereit. Eigentlich sollte GTK+ ausschließlich für die graphische Oberfläche des Grafikprogramms GIMP entwickelt werden. Mittlerweile wird GTK+ allerdings von vielen Anwendungen, darunter auch das GNOME Projekt [gno06], verwendet.

Mit Hilfe dieser Bibliothek ist es uns möglich eine grafische Oberfläche in C auf Basis des X-Window-Systems für die von uns verwendeten Endgeräte zu schreiben.

2.2 Zaurus mit GPE

Als Testsysteme kamen Sharp Zaurus SL-6000 [zau04] zum Einsatz. Diese Geräte verfügen über einen Intel XScale PXA255 Prozessor mit 400MHz und einer integrierten

WLAN Karte. Die wichtigsten technischen Eigenschaften sind in Tabelle 2.1 zu finden:

| | |
|------------|-----------------|
| CPU | 400 MHz |
| BILDSCHIRM | 480 x 640 PIXEL |
| RAM | 64 MB |
| ROM | 64 MB |
| NETZWERK | WLAN 802.11b |

Tabelle 2.1: Sharp Zaurus SL-6000 Hardware

Als Benutzeroberfläche wird das GPE Palm Environment [gpe06] verwendet. GPE stellt eine speziell für Palmtops/Handhelds entwickelte Oberfläche für Geräte bereit, die auf GNU/Linux oder anderen UNIX Systemen aufbaut. In unserem Fall wird GPE auf Basis von dem OpenEmbedded System OpenZaurus [oz206] eingesetzt. GPE ist in C geschrieben und benutzt das X-Window-System und GTK+.

Kapitel 3

Toolkit

In diesem Kapitel wird das entwickelte Toolkit für die Vermessung und Modellierung vorgestellt. Als Grundlage für die Werkzeuge wird eine Landkarte des Messgebiets benötigt. Auf der Basis dieser Landkarte ist es dann möglich mit *Mapconfig* den Ablauf der Vermessung und die Positionen der Endgeräte für die Messung zu bestimmen. Während der Messung wird auf den Endgeräten *Mapkit* eingesetzt. *Mapkit* führt die Positionierung der Endgeräte auf dem Messgebiet mit Hilfe der Landkarte und die Messung durch. Die *Bibliothek* wertet die gesammelten Daten aus und stellt eine Schnittstelle zur Abfrage dieser Daten bereit. Über diese Schnittstelle ist der *Analyzer* in der Lage die Messergebnisse graphisch darzustellen.

3.1 Landkarte

Als Grundlage für alle Werkzeuge wird eine Landkarte oder Luftbild von dem zu vermessenden Gebiet im JPEG-Format benötigt. Anhand dieses Bildes, im folgenden als Karte bezeichnet, muss es durch Betrachten möglich sein die eigene Position zu bestimmen. Die Karte wird zur Positionierung der Messgeräte und zur Visualisierung der Ergebnisse eingesetzt. Eine Karte mit hoher Auflösung ist hier von Vorteil. Zu beachten ist, dass die Karte für die Verwendung in allen drei Werkzeugen dieselbe Höhe und Breite haben muss und gegebenenfalls auf leistungsschwache Endgeräte angepasst wird. In unserem Fall kommt ein Luftbild der Universität Düsseldorf, angefertigt vom Fachbereich

Geographie, zum Einsatz. Alternativ ist Kartenmaterial in ausreichender Auflösung von vielen Städten bei Google Earth [gea06] zu finden.

3.2 Mapconfig

Die Aufgabe von Mapconfig ist die Erstellung der Szenarien für die Vermessung, wobei ein Szenario eine Positionierung aller Messgeräte beschreibt. Da es nicht möglich ist jedes Koordinatenpaar der Landkarte zu vermessen, wird ein Raster über die Karte gelegt, wobei jedes Rasterkreuz einen potenziellen *Messpunkt* darstellt. Die Wahl der Rastergröße und die Auswahl der aktiven Messpunkte sind die Schritte eins und zwei in dem Erstellungsprozess für die Szenarien mit Mapconfig.

Für die Steuerung der Messung wird eine Kommunikation zwischen allen Geräten benötigt. Diese Kommunikation soll über eine multi-hop Verbindung ermöglicht werden, die mit Hilfe von *Stützstellen*, welche eine feste Position haben, sichergestellt wird. Dadurch entsteht ein hierarchisches System, in dem wir unsere Geräte in den einzelnen Szenarien positionieren müssen. Die Positionierung findet anhand von zwei Algorithmen statt, welche am Ende des Kapitels ausführlich beschrieben werden.

3.2.1 Raster

Die Messung der Linkqualität zwischen zwei beliebigen Punkten in unserem Messgebiet ist zwar möglich, die Messung zwischen allen Punkten aber zeitlich unmöglich. Aus diesem Grund legen wir ein Raster über die Karte, welches eine Vermessung des ganzen Gebietes an ausgewählten Punkten ermöglicht. Jedes Rasterkreuz stellt dabei einen potenziellen Messpunkt dar. Die Wahl des Rasters in der Software hängt stark vom Maßstab der verwendeten Karte und der gewünschten Genauigkeit der Vermessung ab. Je größer die Genauigkeit, desto kleiner muss das Raster sein. Allerdings wächst die Anzahl der Messrunden mit der Auflösung des Rasters dramatisch. Es besteht also ein Trade-Off zwischen der Genauigkeit einerseits und dem Aufwand für die Messungen andererseits. In unseren Messungen hat sich eine Rastergröße von 15-20 Metern bewährt.

3.2.2 Messpunkte

Nachdem die Karte geladen und das Raster eingestellt ist, müssen die gewünschten Messpunkte aktiviert werden. Bei der Wahl der Messpunkte ist zu beachten, dass sie auf keinem Gebäude und an keinem unerreichbaren Ort liegen, da an jeden gewählten Punkt während der Messung ein Gerät bewegt werden muss.

Ein Beispiel für ein markiertes Gebiet ist in Abbildung 3.1 zu sehen. Sie zeigt einen Screenshot von Mapconfig mit der von uns verwendeten Karte der Universität Düsseldorf. Die aktiven Messpunkte auf dem gewählten Raster sind weiß markiert, die nicht gewählten Messpunkte sind rot dargestellt.

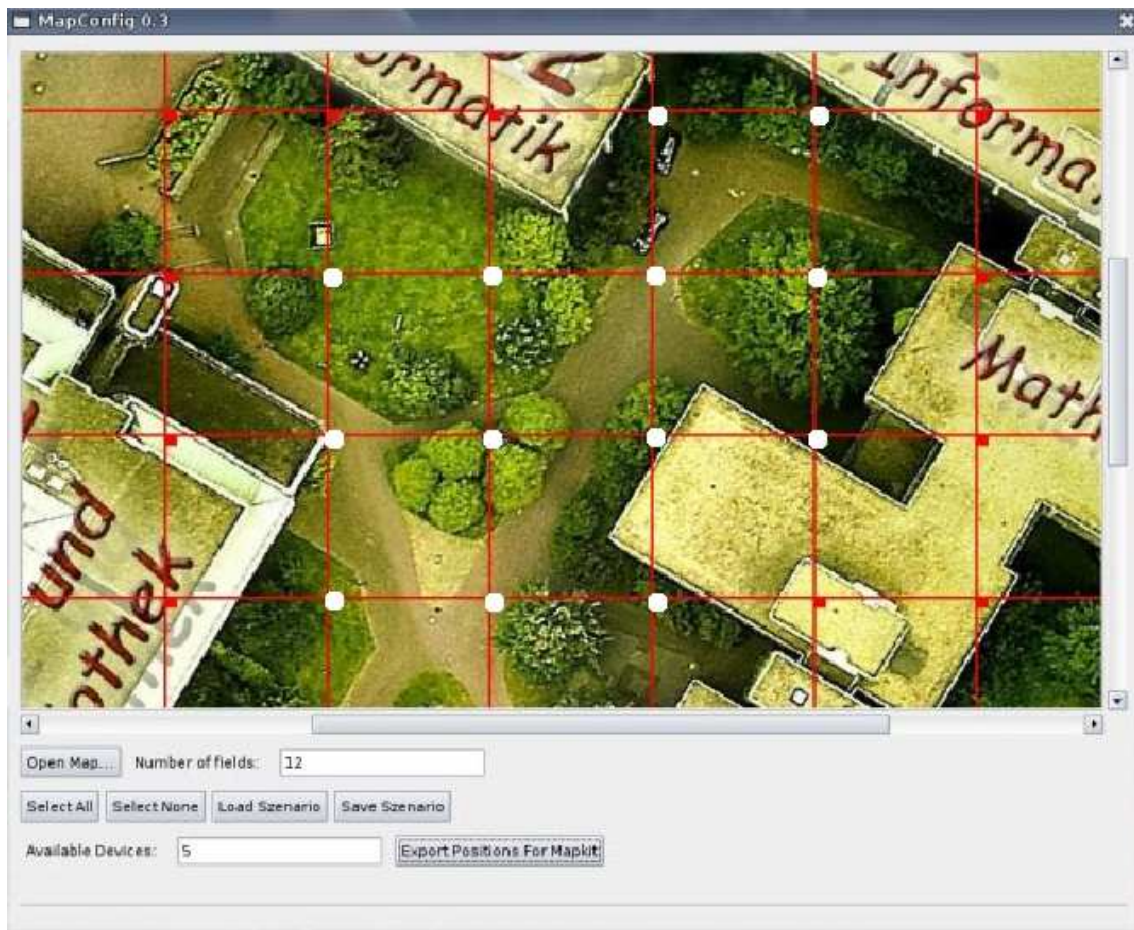


Abbildung 3.1: Beispiel für Platzierung der Messpunkte

3.2.3 Stützstellen

Da die Mapkit Software die Vermessung, nachdem sie einmal angestoßen wurde, selbst steuert, ist eine Kommunikation zwischen den Geräten nötig. Diese soll die Aktionen der Geräte zeitlich abstimmen, so dass nicht gleichzeitig von mehreren Knoten Messpakete versendet werden. Um eine solche Kommunikation zu erreichen muss eine multi-hop Verbindung zwischen zwei Geräten bestehen. Wenn dies der Fall ist, haben wir ein unpartitioniertes Netzwerk. Wir bauen ein solches Netzwerk indem wir Stützstellen in dem Gebiet platzieren. Stützstellen sind ganz normale Messgeräte, welche ihre Position nicht verändern. Sie werden so platziert, dass jede Stützstelle mit allen Anderen multi-hop kommunizieren kann. Außerdem muss jeder gewählte Messpunkt ebenfalls mit einer Stützstelle kommunizieren können. Dadurch entsteht ein multi-hop Netzwerk, in dem jedes Messgerät mit jedem anderen kommunizieren kann.

Die Platzierung der Stützstellen ist meistens durch einen Blick auf die Karte möglich. Um ganz sicher zu gehen empfiehlt sich allerdings eine Vormessung. Diese könnte zum Beispiel mit dem Programm `ping` durchgeführt werden. Wir platzieren dazu Geräte an allen Stützstellen und bewegen uns mit einem anderen Gerät von Messpunkt zu Messpunkt. Falls von jedem Messpunkt mindestens eine Stützstelle zu erreichen ist und auch die Kommunikation der Stützstellen untereinander klappt haben wir eine gute Wahl getroffen.

3.2.4 Platzierung der Geräte

Nachdem wir die Messpunkte und Stützstellen auf unserem Raster festgelegt haben können wir Mapconfig automatisch die Platzierung der Geräte in unseren einzelnen Szenarien vornehmen lassen. Da in den meisten Fällen weniger Messgeräte als Messpunkte vorhanden sind, müssen die Geräte in mehreren aufeinander folgenden Runden platziert werden. Zu diesem Zweck werden Algorithmen entwickelt, deren Ziel eine geschickte Positionierung ist, so dass jeder Messpunkt einmal mit jedem anderen Messpunkt vermessen wird. Dabei sollte es so wenig wie möglich doppelte Messungen geben und die Positionen der Helfer, welche die Geräte an die richtige Position tragen, sollte sich nach Möglichkeit nicht oft ändern, um einen schnellen Ablauf der Messung zu ermöglichen und die Fehlerwahrscheinlichkeit gering zu halten. Der Algorithmus, den

Mapconfig dafür verwendet, lässt sich in zwei Fälle unterteilen: Den Fall für die Platzierung mit einer Stützstelle und den Fall für die Platzierung mit mehreren Stützstellen.

Platzierung mit einer Stützstelle

Ziel des Algorithmus ist, jeden markierten Messpunkt mit jedem anderen Messpunkt einmal zu vermessen und dabei so wenige Messungen wie möglich doppelt durchzuführen. Zu diesem Zweck werden die Geräte in zwei Gruppen eingeteilt. Die Messpunkte können als Bitfolge interpretiert werden, wobei die erste Gruppe links am Beginn unserer Bitfolge und die zweite Gruppe direkt dahinter platziert wird. Für dieses Bitmuster wird dann eine Messrunde durchgeführt. Anschließend wird die zweite Gruppe um die Anzahl der Geräte in dieser Gruppe nach rechts verschoben und eine nächste Messrunde durchgeführt. Die zweite Gruppe wird immer weiter bis an das Ende der Bitfolge verschoben. Wenn alle Positionen durch die zweite Gruppe einmal belegt waren, wird auch die erste Gruppe um die Anzahl der Geräte in ihr nach rechts verschoben. Die zweite Gruppe wird wieder rechts davon platziert und durchläuft dann alle Bits. So wird vorgegangen bis die erste Gruppe ganz rechts platziert werden muss und kein Platz mehr für die zweite Gruppe ist. Für diesen Fall muss auch keine Messung mehr durchgeführt werden und der Algorithmus wird beendet. Anhand des Pseudocodes und eines Beispiels wird der Algorithmus nochmal erklärt.

```
void PlatzierungMitEinerStützstelle() {
    PositioniereStützstelle();
    AnzahlGeräte--;
    g1=(AnzahlGeräte+1)/2;
    g2=(AnzahlGeräte)/2;

    for(i=0;i<AnzahlPositionen/g1-1;i++){
        //Gruppe 1 jedes i weiter nach rechts schieben
        Positioniere_g1(i);
        //Setze Geräte aus Gruppe 2 nacheinander auf
        //alle Positionen rechts von Gruppe 1
        for(k=i+1;k<AnzahlPositionen/g2;k++){
```

```

    Positioniere_g2(k);
    ErstelleMessung();
}
}
}

```

An einem Beispiel mit acht Messpunkten, einer Stützstelle und fünf für die Messung zur Verfügung stehenden Geräten ist der Algorithmus gut zu verstehen. Hierbei muss ein Gerät immer auf der Position der Stützstelle stehen, während die anderen vier Geräte frei beweglich sind. Wir definieren unsere Messpunkte als $x_1, x_2, \dots, x_8 \in X$. Dabei interessiert die absolute Position auf der Landkarte für die Anwendung des Algorithmus nicht. Wir teilen die Geräte in zwei Gruppen (G_1, G_2) , wobei $g_1, g_2 \in G_1$ und $g_3, g_4 \in G_2$. Daraufhin werden die Geräte gruppenweise auf die acht Positionen verteilt. Zu Beginn werden g_1 und g_2 auf die Positionen x_1 und x_2 gesetzt. Dies geschieht durch die äußere for-Schleife in dem Pseudocode. Die freien Positionen werden nacheinander in der zweiten for-Schleife von g_3 und g_4 besetzt. Das bedeutet im ersten Szenario sind g_3 und g_4 auf x_3 und x_4 . Im zweiten Szenario befinden sich g_1 und g_2 unverändert auf ihren Positionen, g_3 und g_4 rücken dann auf die Positionen x_5 und x_6 . Wenn alle Positionen von den Geräten aus G_2 abgedeckt sind, rückt auch G_1 um die Anzahl der Geräte in G_1 auf und G_2 deckt wiederum alle Positionen nacheinander ab. Das komplette Beispiel ist in Tabelle 3.1 zu finden.

| SZENARIO | x_1 | x_2 | x_3 | x_4 | x_5 | x_6 | x_7 | x_8 |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1 | g_1 | g_2 | g_3 | g_4 | | | | |
| 2 | g_1 | g_2 | | | g_3 | g_4 | | |
| 3 | g_1 | g_2 | | | | | g_3 | g_4 |
| 4 | | | g_1 | g_2 | g_3 | g_4 | | |
| 5 | | | g_1 | g_2 | | | g_3 | g_4 |
| 6 | | | | | g_1 | g_2 | g_3 | g_4 |

Tabelle 3.1: Positionierung der Messgeräte mit einer Stützstelle

Platzierung mit mehreren Stützstellen

Für den Fall, dass wir n Stützstellen für die Kommunikationssicherung in unserem Netzwerk benötigen werden unsere Geräte ebenfalls in zwei Gruppen G_1 und G_2 geteilt und

nacheinander immer zwei Stützstellen zugeordnet. Unsere Stützstellen definieren wir als S_1, S_2, \dots, S_n und fassen alle Messpunkte in n möglichst gleich große Gruppen zusammen. Für das Beispiel mit acht Messpunkten und $n = 2$ fassen wir also x_1, x_2, x_3, x_4 zur Gruppe X_1 und x_5, x_6, x_7, x_8 zur Gruppe X_2 zusammen. Anschließend ordnen wir unsere Gruppen X_1, X_2 den Stützstellen S_1, S_2 zu. Damit die Gruppen so geschickt gewählt werden, dass die Messpunkte auch auf der Landkarte um die Stützstelle platziert werden, achtet die Software bei der Einteilung der Gruppen auf die absoluten Koordinaten von x_1, \dots, x_8 . Wir haben unser Messgebiet nun in zwei gleich große Teilgebiete unterteilt. Diese Teilgebiete müssen nun im ersten Schritt untereinander vermessen werden. Im zweiten Schritt muss jedes Teilgebiet intern vermessen werden. Da ein Teilgebiet genau aus einer Stützstelle besteht, wird der Algorithmus für die Platzierung mit einer Schnittstelle für den zweiten Schritt angewendet. Für die Vermessung der Teilgebiete untereinander kommt ein ähnlicher Algorithmus zum Einsatz (s. Tabelle 3.2). Hier wird die Verschiebung der Gerätegruppen auf den Messpunkten der zugeordneten Stützstelle vorgenommen. Nach der Vermessung der Stützstellen untereinander kommt es zur Vermessung der einzelnen Stützstellen. Zu diesem Zweck werden alle Gruppierungen der Geräte aufgelöst und zwei neue Gruppen, wie oben beschrieben gebildet und die Vermessung für alle S_1, \dots, S_n durchgeführt. In unserem Beispiel wird die Anwendung des obigen Algorithmus in den Szenarien fünf und sechs dargestellt.

| SZENARIO | S_1 | | | | S_2 | | | |
|----------|-------|-------|-------|-------|-------|-------|-------|-------|
| | x_1 | x_2 | x_3 | x_4 | y_1 | y_2 | y_3 | y_4 |
| 1 | g_1 | g_2 | | | g_3 | g_4 | | |
| 2 | g_1 | g_2 | | | | | g_3 | g_4 |
| 3 | | | g_1 | g_2 | g_3 | g_4 | | |
| 4 | | | g_1 | g_2 | | | g_3 | g_4 |
| 5 | g_1 | g_2 | g_3 | g_4 | | | | |
| 6 | | | | | g_1 | g_2 | g_3 | g_4 |

Tabelle 3.2: Positionierung der Messgeräte g_1, \dots, g_4 mit zwei Stützstellen S_1, S_2

Optimierung

Nach der Vorstellung der Algorithmen können wir uns nun Anwendungsfälle überlegen, welche für diese optimal sind. Nach diesen optimalen Fällen können wir dann die Anzahl der Messpunkte und Geräte festlegen. Wenn wir oben genanntes Beispiel mit einer Stütz-

stelle und $8 + 1$ Messpunkten betrachten, wir aber weiterhin 4 Geräte zur Messung nutzen, fällt auf, dass die Verschiebung der Gerätegruppen nicht mehr richtig funktioniert. Der Messpunkt x_9 wird also von Gerät g_3 besetzt, und g_4 hat keine neue Position mehr zur Verfügung. Dieser Fall tritt für jede Positionierung der Gruppe G_1 auf. Das heißt es kommen vier Szenarien für nur einen Messpunkt dazu. Ausschließen kann man dies, indem die Anzahl der Messgeräte und der Messpunkte der Formel

$$K \bmod \left(\frac{G-S}{2}\right) = 0$$

angepasst wird, wobei K für die Anzahl der Messpunkte, G für die Anzahl der Geräte und S für die Anzahl der Stützstellen steht. Diese Formel gibt uns die Möglichkeit zu überprüfen, ob die Wahl für K , G und S für die von Mapconfig verwendeten Algorithmen optimal ist. Nach Anpassung der Vermessung an diese Formel lässt sich die Anzahl der Szenarien nach

$$\sum_{i=1}^{\frac{2K}{G-S}-1} \left(\frac{2K}{G-S} - i\right)$$

bestimmen¹. Hier beschreibt die Formel die Aufsummierung aller Messungen, die für eine Positionierung der Gruppe G_1 gemacht werden müssen. In unserem ersten Beispiel mit einer Stützstelle ist $K = 8$, $G = 5$ und $S = 1$. Dafür liefert die Formel $3 + 2 + 1 = 6$ Szenarien. Auch die Überprüfung auf Optimierung liefert $8 \bmod \left(\frac{5-1}{2}\right) = 0$.

3.3 Mapkit

Die Mapkit Software ist für die eigentliche Durchführung der Messung zuständig. Mapkit unterstützt die Teilnehmer des Experimentes bei der Positionierung des Messgerätes auf der Landkarte und steuert das Experiment nach dem Start automatisch, wodurch die Geräte von den Teilnehmern nur noch an ihre richtige Position gebracht werden müssen um die Messung durchzuführen.

¹Für eine nicht optimierte Wahl der Messpunkte, Geräte und Stützstellen trifft diese Formel nicht zu.

3.3.1 Positionierungsdienst

Um das in Mapconfig gewählte Gebiet zu vermessen, müssen die Messgeräte in jedem Szenario an eine Position auf der Landkarte gebracht werden. Deswegen benötigen wir einen Positionierungsdienst, der die Geräte zu den Messpunkten führt und es erlaubt die Position zu bestätigen.

Das Global Positioning System (GPS) erfüllt diese beiden Aufgaben. GPS ist ein satellitengestütztes System, welches z.B. in Autos oder vom Militär zur Navigation und Positionierung eingesetzt wird. Leider hat GPS den Nachteil, dass eine Sichtverbindung zu mehreren Satelliten bestehen muss. Da die Vermessung durchgeführt wird um Einflüsse auf die Funkübertragung von Gebäuden, Bäumen etc. zu analysieren, muss der Positionierungsdienst auch in solchen Umgebungen einwandfrei funktionieren. In der Nähe solcher Hindernisse wird allerdings auch die Funkübertragung der GPS-Satelliten gestört, wodurch eine Positionierung erschwert wird. Außerdem ist für die Satellitenpositionierung zusätzliche, z. T. teure Hardware erforderlich. Aufgrund dieser beiden Nachteile ist GPS für unsere Vermessung ungeeignet.

Eine bessere und genauere Lösung für unseren Anwendungsfall bietet die manuelle Positionierung. Dazu wird lediglich eine Landkarte des Messgebietes benötigt. Dem Benutzer wird bei dieser Art der Positionierung die Landkarte mit der Zielposition für das aktuelle Szenario gezeigt. Diese ist rot markiert und kann durch den Benutzer mit einem Klick auf die Karte bestätigt werden. Daraufhin bestätigt das Gerät die Position des Teilnehmers durch das Versenden einer Nachricht. Da die Auflösung der über Google Earth [gea06] erhältlichen Karten sehr gut ist, kann eine Positionierung sehr genau überall auf der Karte durchgeführt werden. Außerdem benötigt diese Methode keine zusätzliche Hardware und eine eigene Landkarte ist schnell in das System zu integrieren.

3.3.2 Messung

Ziel der Messungen ist es, die Linkqualität anhand der Paketverlustrate zu bestimmen. Zu diesem Zweck verschickt die Mapkit Software eine vorher bestimmte Anzahl von Paketen mit einer auswählbaren Paketgröße an die Broadcastadresse.

Um die eingehenden Pakete bei der Analyse unterscheiden zu können, versieht jeder Sender seine Pakete mit einer eindeutigen Sequenznummer, welche durch die Gerätenummer und die aktuelle Messrunde eindeutig bestimmt wird. Jeder Empfänger speichert nun alle empfangenen Sequenznummern, so dass die Paketverlustrate zwischen dem Sender und Empfänger nach dem Messlauf bestimmt werden kann.

3.3.3 Steuerung des Experimentes

Während der Messungen müssen die Geräte ständig neu positioniert werden und zeitlich abgestimmt Messpakete senden. Falls Messpakete schon gesendet werden, wenn andere Geräte noch nicht positioniert sind oder falls mehrere Geräte gleichzeitig Messpakete senden ist die Messung unbrauchbar. Eine manuelle Steuerung mit Funkgeräten oder ähnlichem ist sehr schwer, da mehrere Aktionen von den Benutzern koordiniert werden müssen. Außerdem verzögert eine manuelle Steuerung die Messung unnötig und die Fehlerwahrscheinlichkeit durch Benutzereingaben oder Kommunikationsfehler erhöht sich. Daher wird eine automatische Steuerung der Messung nötig.

Die automatische Steuerung läuft nach dem Dominoprinzip ab. Jedes Gerät reagiert ausschließlich auf die Kontrollnachricht des Vorgängers. Da die Geräte sich mehrfach positionieren und Messpakete senden müssen läuft die Steuerung rundenbasiert ab. Innerhalb einer Runde muss jedes Gerät zwei Aufgaben erfüllen und diese durch Fluten einer Kontrollnachricht bestätigen. Zuerst muss es sich positionieren, und nach der Positionierung aller Geräte muss es die Messpakete senden. Zur Bestätigung dieser Aufgaben werden zwei Kontrolltypen eingeführt. Ein dritter Kontrolltyp wird benötigt um eine nächste Runde zu beginnen.

Typ 0 Bestätigt die Positionierung.

Typ 1 Bestätigt das Ende des Sendevorgangs.

Typ 2 Fordert alle Geräte zur neuen Positionierung auf.

Zusammen mit diesen Nachrichten versendet jedes Gerät seine eigene Gerätenummer, welche vorher in der Konfigurationsdatei von Mapkit angegeben werden muss. Beim

Empfang einer solchen Kontrollnachricht reagiert jedes Gerät ausschließlich auf Nachrichten seines Vorgängers. Ist also die Gerätenummer der empfangen Nachricht gleich der Gerätenummer des Vorgängers, überprüft das Gerät den eigenen Status und handelt entsprechend dem Kontrolltyp der Nachricht. Anschließend versendet das Gerät eine Kontrollnachricht mit demselben Typ und der eigenen Gerätenummer. Eine Ausnahme hierbei bildet das Gerät mit der höchsten Nummer. Diesem ist es möglich den Nachrichtentyp zu verändern und statt der eigenen Gerätenummer eine Null in die Kontrollnachricht zu schreiben. Diese Nachricht versteht dann das Gerät mit der Nummer eins als Aufforderung zu handeln und eine neue Runde beginnt.

In Abbildung 3.2 wird die Funktionsweise der Steuerung mit drei Geräten deutlich. Die Geräte sind von eins bis drei durchnummeriert. Zur Kontrolle dieses Ablaufes wird ein *Master* eingesetzt. Dieser startet die Messung und ihm ist es möglich im Fehlerfall verlorene Kontrollnachrichten erneut zu senden, so dass die Messung fortgesetzt werden kann.

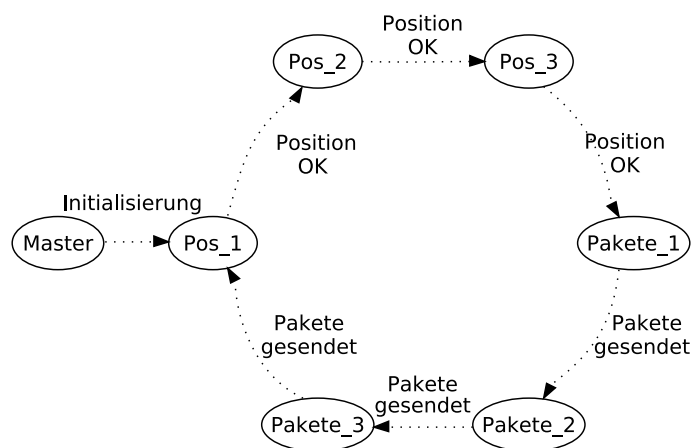


Abbildung 3.2: Ablauf eines Szenarios am Beispiel von drei Geräten

Nach der Initialisierung durch den Master muss das erste Gerät in *Pos_1* seine Position bestätigen. Anschließend bestätigen auch Gerät zwei und drei ihre Positionen. Daraufhin ändert Gerät drei (*Pos_3*) aufgrund der höchsten Gerätenummer den Kontrolltyp. Dadurch wird Gerät eins (*Pakete_1*) aufgefordert die Messpakete zu senden. Nachdem die Geräte zwei und drei die Messpakete gesendet haben, erhält Gerät eins die Aufforderung die nächste Position zu bestätigen. Den Hinweis eine neue Position einzunehmen, welcher nicht im Diagramm dargestellt ist, erhalten alle Geräte durch Fluten von Gerät drei. Dadurch findet die Positionierung parallel statt, die Bestätigung der Positionen wird aber nacheinander ausgeführt.

3.4 Bibliothek

3.4.1 Algorithmus

Aufgrund der Verwendung eines Rasters bei den Messungen ist es nicht direkt möglich die Linkqualität für zwei Koordinaten aus den Messdaten abzulesen. Bei einer Anfrage für die Linkqualität von Startpunkt S zu Zielpunkt Z können also vier verschiedene Fälle auftreten.

1. Beide Punkte sind ein Messpunkt
2. S ist ein Messpunkt, Z ist kein Messpunkt
3. S ist kein Messpunkt, Z ist ein Messpunkt
4. Weder S noch Z liegt auf einem Messpunkt

Für den ersten Fall können wir das Ergebnis sofort aus den Messdaten ablesen. Für die anderen drei Fälle müssen wir die Linkqualität aus den vorhandenen Messwerten berechnen, wozu im Folgenden ein Interpolationsverfahren entwickelt wird.

Punkt Interpolation

Die Grundidee der Punkt Interpolation ist das Betrachten der Linkqualität an den n nächstliegenden Punkten, wobei wir günstige Werte für n später ermitteln werden. Gehen wir davon aus, dass unser Startpunkt S ein Messpunkt ist, welcher genau auf einem Rasterkreuz liegt und der Zielpunkt Z irgendwo zwischen dem Raster liegt. Dann hat Z im Fall ohne Hindernisse vier direkte Nachbarn, zu denen wir Messergebnisse besitzen. Wir wählen nun die n Punkte mit dem kleinsten Abstand zu Z und werten die Signalqualität von S zu diesen n Punkten aus. Diese n Signalqualitäten müssen nun gemittelt werden. Eine einfache Mittelwertberechnung ist hier nicht günstig, da Z meistens nicht genau zwischen den ausgewählten Punkten liegt und somit eine Gewichtung des Abstandes der Messpunkte besser ist. Wir gewichten diese Punkte also nach ihrem Abstand, wobei der

kleinste Abstand das größte Gewicht, und der größte Abstand das kleinste Gewicht erhält. Ein Punkt der sehr nah an einem Nachbarn liegt, erhält so beinahe die Linkqualität dieses Nachbarn. Für den dritten Fall wird die Berechnung genauso durchgeführt, mit dem einzigen Unterschied, dass die Nachbarn des Start- statt des Zielpunktes gesucht und ausgewertet werden müssen.

Für den vierten Fall müssen die Nachbarpunkte von S und Z gesucht und miteinander berechnet werden. Anhand der folgenden Codebeschreibung lässt sich der Algorithmus nachvollziehen.

1. Suche die n nächsten Nachbarpunkte von S .
2. Suche die n nächsten Nachbarpunkte von Z .
3. Suche für jeden gewählten Punkt um S die Linkqualität zu jedem gewählten Nachbarn von Z aus den Messdaten.
4. Berechne für jeden gewählten Punkt aus S die Linkqualität zu Z durch Gewichtung der gewählten Nachbarn von Z .
5. Gewichte nun die berechneten Werte der gewählten Punkte um S und berechne daraus die Signalqualität für S zu Z .

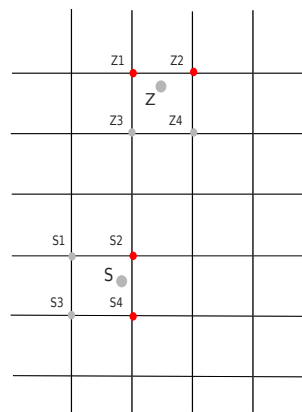


Abbildung 3.3: Beispiel für die Punkt Interpolation mit $n = 2$

In dem Beispiel aus Abbildung 3.3 für $n = 2$ werden im ersten Schritt s_2, s_4 und im zweiten z_1, z_2 gewählt. Daraufhin werden die Messwerte von $s_2 \rightarrow z_1, s_2 \rightarrow z_2, s_4 \rightarrow z_1$ und $s_4 \rightarrow z_2$ gesucht. Aus diesen Messwerten wird für die Punkt s_2, s_4 der durch

$\overline{z_1 Z}$ und $\overline{z_2 Z}$ gewichtete Mittelwert berechnet. Als Ergebnis erhalten wir die berechnete Linkqualität von $s_2 \rightarrow Z$ und $s_4 \rightarrow Z$. Im fünften Schritt gewichten wir nun auch s_2, s_4 durch den Abstand zu S und berechnen damit den Wert der Linkqualität von $S \rightarrow Z$.

Allerdings ist zu beachten, dass nicht jeder Messpunkt in der Umgebung S oder Z ein geeigneter Messpunkt ist. In Abbildung 3.4 ist der Punkt s_1 sehr weit von S entfernt, so dass die Messwerte von dieser Position bei einer Interpolation mit $n = 4$ keinen brauchbaren Beitrag zum Wert S liefern sondern diesen nur verfälschen würden.

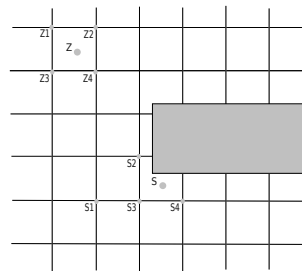


Abbildung 3.4: Wahl der Messpunkte bei der Punkt Interpolation mit $n = 4$

Somit muss für die Punkt Interpolation ein maximaler Abstand zwischen S_1, \dots, S_4 und S bestimmt werden, bei dem der Messwert noch verwendet wird. Für diesen maximalen Abstand r wurde nach Analyse der Messdaten die Breite eines Rasterfeldes R gewählt. Somit verändert sich in diesem Fall die Interpolation mit $n = 4$ zu einer Interpolation, die nur drei Punkte um S aber weiterhin vier um Z betrachtet.

Diese Abstandsregel tritt häufiger mit steigendem n ein. Allerdings kann es auch sein, dass am Rand des vermessenen Gebietes für $n = 2$ nur ein Messpunkt gewählt wird. Weiterhin liefert die Interpolation durch diese Regel für Punkte außerhalb des Gebietes automatisch eine Linkqualität von null.

3.4.2 Schnittstellenbeschreibung

Um die Bibliothek und die eigenen Messergebnisse in andere Anwendungen einzubinden sind folgende Schritte durchzuführen. Als erstes muss die Datei *analyze.h* in die Anwendung eingebunden werden. Diese Datei stellt einige Funktionen bereit, von denen die

wichtigsten kurz erklärt werden. Es ist möglich die Schnittstelle mit jeder Programmiersprache zu nutzen, die das Einbinden von C Funktionen zulässt.

```
void readData(char* path);
```

genutzt. Dieser Funktion muss der Pfad zu den Messdaten übergeben werden, welche nach der Messung von den Endgeräten geladen werden müssen. Die Funktion

```
int getQuality(int x1,int y1, int x2, int y2,int algo);
```

liefert einen *int* Wert zwischen 0 und 100, der die Linkqualität zwischen den Punkten (x_1/y_1) und (x_2/y_2) , berechnet durch den verwendeten Algorithmus, in Prozent angibt. Unseren Messergebnissen zur Folge bietet sich an $algo = 2$ zu wählen. Weiterhin bietet die Bibliothek die Möglichkeit sich die berechneten Linkqualitäten von einem Punkt aus zu jedem anderen Punkt auf der Karte zurückgeben zu lassen. Die Funktion

```
int** getQualityAround(int x, int y, int width, int height, int algo);
```

liefert ein zweidimensionales *int*-Array und benötigt als Eingabe eine Position, die Größe der Karte und den zu verwendenden Algorithmus. Außerdem bietet die Bibliothek noch Funktionen mit denen die reinen Messergebnisse ausgelesen werden können. Erklärungen dazu sind der Header-Datei zu entnehmen.

3.4.3 Analyzer

Der Analyzer ist eine grafische Oberfläche, die auf die Schnittstelle der Bibliothek aufsetzt. Mit ihm ist es möglich sowohl die rohen Messdaten als auch die Interpolierten Linkqualitäten zu betrachten. Screenshots des Analyzers sind im Kapitel 4.3 zu finden.

Kapitel 4

Vermessung und Ergebnisse

Mit Hilfe von Mapconfig, Mapkit und dem Analyzer wurden die Messungen vorbereitet, durchgeführt und ausgewertet. Die folgenden Absätze erklären den Aufbau der Messungen, beschreiben Erfahrungen der Messdurchführung, visualisieren die Messergebnisse und verifizieren die interpolierten Linkqualitäten anhand von zusätzlich durchgeführten Messungen.

4.1 Konfiguration

Um allgemein Störungen durch andere WLAN-Netze während der Messung zu vermeiden kam vor Beginn des Messlaufs Wi-Spy [wis06] zum Einsatz. Wi-Spy ist ein Spektrum-Analyse Gerät in der Form eines USB-Sticks. Mit Hilfe der mitgelieferten Software für Windows oder freier Software für andere Betriebssysteme ist es möglich den gesamten WLAN Frequenzbereich zu analysieren, so dass schnell ein freier WLAN-Kanal für die Vermessung gefunden werden kann. Mögliche Störungen durch andere Netze oder sonstige Quellen von Radiostrahlung werden so minimiert. In unserem Fall wurden die Geräte in einem ad-hoc Netzwerk auf Kanal zwei verbunden.

Zur Positionierung kam eine vom Fachbereich Geographie angefertigte Karte des Geländes der Universität Düsseldorf zum Einsatz. Nach einigen durchgeführten Messläufen mit kleinem Raster wählten wir einen Rasterabstand von ca. 18 Metern. Ein kleineres Raster hatte zur Folge, dass sehr viel mehr Messungen durchgeführt werden mussten,

wohingegen ein größeres Raster zu ungenau bei der Interpolation ist. Weitere Informationen über das vermessene Gebiet und die gewählte Karte sind in Tabelle 4.1 zu finden.

| | |
|---------------------------------|----------------------------|
| KARTE | 1500 X 1500 PIXEL |
| GRÖSSE DES VERMESSENEN GEBIETES | CA. 90 X 45 M |
| RASTERABSTAND | 125 PIXEL $\hat{=}$ 18,75M |
| MASSSTAB | 1 PIXEL $\hat{=}$ 15 CM |

Tabelle 4.1: Details der durchgeführten Messungen

Auf diesem Raster wurden nun zwölf Messpunkte, verbunden durch eine Stützstelle, gewählt. Zur Vermessung standen dabei fünf Geräte zur Verfügung. Mit dieser Konfiguration erstellte Mapconfig 15 Szenarien für die komplette Vermessung.

Nach der Installation der Mapkit Software auf den Endgeräten zusammen mit der erstellten Szenario Datei, musste zuletzt die Konfigurationsdatei jedes einzelnen Gerätes angepasst werden. Dabei kann zusätzlich zur Gerätenummer auch die Anzahl, die Größe und das Sendeintervall der zu verschickenden Messpakete eingestellt werden. Wir variierten die Anzahl von 100 bis zu 200 Paketen mit einer Größe von 256 Byte in den verschiedenen Messungen.

4.2 Durchführung

Die Messung wird mit Hilfe der automatischen Steuerung der Mapkit Software durchgeführt. Zur Unterstützung bietet sich dennoch die Verwendung von Funkgeräten an. Dadurch können Fehler schneller behoben werden und den Teilnehmern ist es möglich Fragen zur Messung oder dem Ablauf zu stellen.

Zu Beginn der Entwicklung von Mapkit wurden einige kleine Messungen durchgeführt, die in erster Linie die Positionierungsmöglichkeit und die automatische Steuerung testen sollten. Dabei wurden auch Messergebnisse gesammelt, welche aber keinesfalls komplett waren und somit auch nicht in die Auswertung mit einfließen. Nachdem die grundlegenden Algorithmen für Mapconfig entstanden waren und Mapkit in der Lage war in einer Emulation, bei der alle Geräte in einer Radioreichweite waren, fehlerfrei zu Arbeiten,

wurden Messungen in größerem Ausmaß durchgeführt. Leider gab es auch in diesen Messungen noch Systemabstürze, durch welche die gesammelten Daten nicht vollständig waren. Die Messdaten reichten allerdings aus um die Entwicklung der Bibliothek voran zu treiben.

In die endgültige Auswertung fließt eine komplette Messung mit 5 Geräten, vier Teilnehmern und einem Master über zwölf Messpunkte ein. Die Stützstelle blieb unbetreut, da die Position eines Gerätes automatisch bestätigt wird falls keine Änderung gegenüber des letzten Szenarios vorliegt. Dabei ist der Vorteil des in Mapconfig implementierten Algorithmus, dass sich zwei der vier Geräte nur selten neu positionieren mussten. Aufgrund dessen wurde die Position von Mapkit sofort bestätigt und die Messung ging schnell voran. Durch Einflüsse die vorher nicht in der Erstellung der Szenarien berücksichtigt werden konnten, kam es allerdings zwischendurch zu Paketverlusten die aber schnell durch den Master ausgeglichen werden konnten.

Die Durchführung einer solchen Messung dauert ca. 30-40 min, wodurch wir ein Richtmass von 20-30 Szenarien pro Stunde festlegen können. Eine große Rolle hierbei spielt natürlich die Zeit der Positionierung. Je weiter die Wege der Teilnehmer, desto mehr verzögert sich der Beginn des nächsten Szenarios.

4.3 Ergebnisse

Nach der Durchführung müssen die gesammelten Messdaten von den Geräten geladen und dem Analyzer zur Verfügung gestellt werden. Es gibt nun zwei Möglichkeiten die Ergebnisse zu betrachten. Entweder über die zur Verfügung gestellte Schnittstelle oder über die Oberfläche, die eine Visualisierung der Daten ermöglicht.

Im ersten Schritt betrachten wir die Messergebnisse ohne die interpolierten Messpunkte. Dazu ist es dem Analyzer möglich alle Messungen von einem ausgewählten Messpunkt darzustellen. Ein ausgewählter Messpunkte ist in Abbildung 4.1 dargestellt. Es ist deutlich zu erkennen welchen Einfluss Bäume und Gebäude auf die Funkübertragung haben. Die Bäume in der Mitte des Bildes sind bei jeder Messung ein großer Störfaktor und auch hier kommen nur 15 Prozent der verschickten Pakete durch. Der Gebäudekomplex rechts daneben schattet die Funkwellen komplett ab.



Abbildung 4.1: Linkqualitäten von einem Messpunkt aus

In Abbildung 4.2 sind alle mit Hilfe der 2-Punkt Interpolation berechneten Linkqualitäten eines Punktes dargestellt. Der Sender ist hierbei an der Position des weißen Quadrats platziert. Auch an den Interpolierten Werten sieht man eindeutig den Einfluss der Hindernisse auf die Funkausbreitung. Allerdings ist es schwierig die Qualität in der Nähe kleiner Hindernisse korrekt zu ermitteln. Die Bäume in der Mitte des Bildes schatten die Funksignale eigentlich komplett ab. Da wir die Linkqualität aber auch aus Messpunkten neben den Bäumen berechnen, erhalten wir Ergebnisse in einem Bereich von 40% bis 60%. Im Gegensatz dazu kann der Algorithmus anhand der vielen beeinflussten Messergebnisse den Gebäudekomplex sehr wohl erkennen und auch seine Abschattung gut modellieren.



Abbildung 4.2: Interpolierte Linkqualität eines Punktes

4.4 Validierung der interpolierten Messwerte

Nach der Visualisierung der Messergebnisse stellt sich nun die Frage, inwieweit die berechneten Linkqualitäten der Realität entsprechen. Um diese Frage zu beantworten wurden zusätzliche Messungen zeitnah zu der eigentlichen Vermessung durchgeführt. Jeder Teilnehmer nahm fünf zufällig gewählte Positionen in dem zu vermessenden Gebiet ein, wodurch 100 Messungen zwischen zwei Positionen durchgeführt wurden. Diese gemessenen Werte können nun mit den errechneten Werten der Bibliothek verglichen werden. Abbildung 4.3 vergleicht die Punkt Interpolationsverfahren mit drei Parametern für n und zeigt die Abweichung des gemessenen Wertes zum Berechneten. Auf der x-Achse ist die Abweichung in Prozent und auf der y-Achse die Anzahl der Messungen aufgetragen.

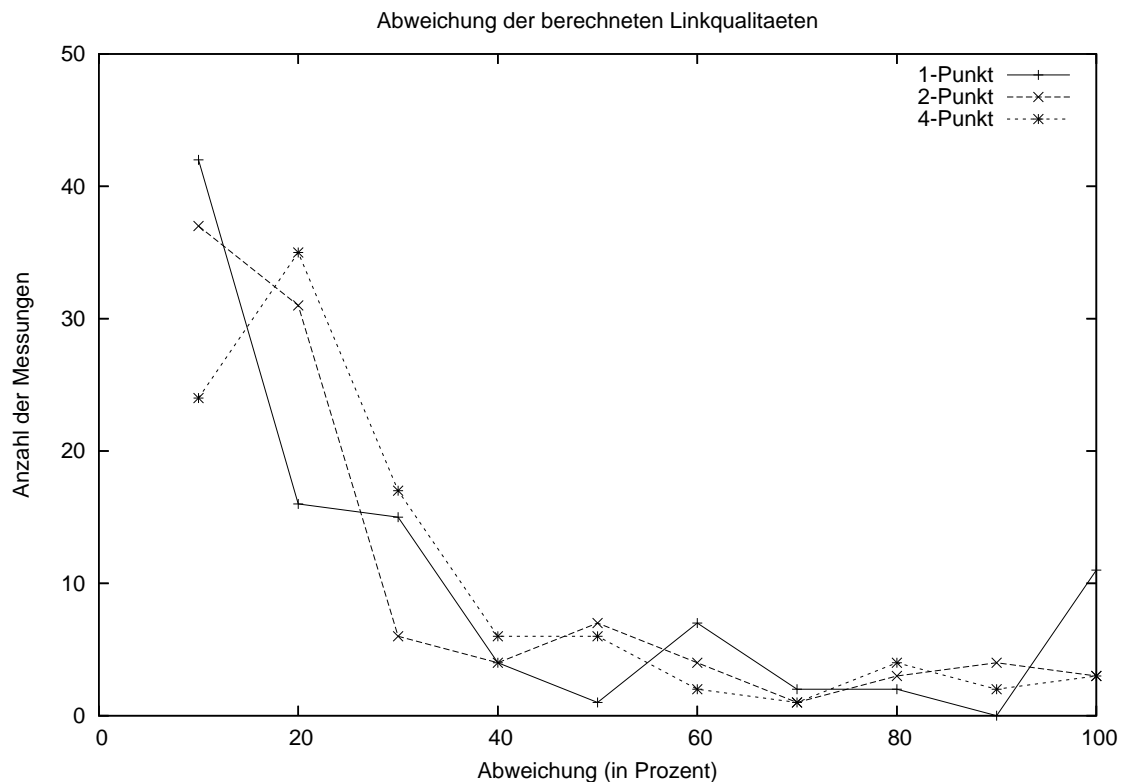


Abbildung 4.3: Analyse der Interpolationsverfahren

Die x-Achse ist in zehn Abschnitte eingeteilt, in denen die Messergebnisse aus einem Bereich zusammengefasst sind. Der x-Wert an 10 stellt alle Linkqualitäten, die eine Abweichung im Bereich von 0 bis 10 Prozent haben, dar.

An diesen Kurven ist zu erkennen, dass für $n = 1$ mehr Linkqualitäten komplett falsch berechnet werden als für $n = 2$ und $n = 4$. Das entgegengesetzte Bild ist in dem Bereich von 90 Prozent zu erkennen, wo die Interpolation mit einem Punkt die Beste ist. Ein solches Verhalten war zu erwarten, da bei den Messergebnissen häufig der Fall auftritt, dass ein Messpunkt eine sehr gute Linkqualität zu einem anderen Messpunkt besitzt, zu dessen direktem Nachbarn aber eine sehr schlechte, da dieser hinter einem Hindernis steht. In diesem Fall würde der Algorithmus für $n = 1$ entweder eine sehr gute oder eine sehr schlechte Linkqualität liefern. Entweder ist das berechnete Ergebnis also sehr genau, oder es ist sehr ungenau. In den Fällen für größeres n fließen beide Messpunkte in das Ergebnis mit ein, so dass auch Linkqualitäten im Bereich von 50 Prozent berechnet werden. Daraus folgt, dass die Abweichung selten minimal oder maximal wird. Wir haben hier also einen Trade-Off zwischen einerseits sehr genauen Ergebnissen, mit vie-

len sehr falschen Ergebnissen und etwas ungenaueren Ergebnissen, die aber selten sehr falsch sind.

Tabelle 4.2 fasst dieses Verhalten der Algorithmen nochmal zusammen. Es ist zu erkennen, dass in knapp 70 Prozent der Messungen, der Algorithmus mit $n = 2$ eine Abweichung von weniger als 20 Prozent hat. Aufgrund dieser Ergebnisse ist $n = 2$ das Optimum für die Punkt Interpolation.

| ABWEICHUNG | $n = 1$ | $n = 2$ | $n = 4$ |
|-------------|---------|---------|---------|
| $\leq 20\%$ | 58 | 68 | 59 |
| $\leq 50\%$ | 78 | 85 | 88 |
| $> 50\%$ | 22 | 15 | 12 |
| $> 80\%$ | 11 | 7 | 5 |

Tabelle 4.2: Abweichung der interpolierten Linkqualitäten anhand von 100 Messungen

Nun stellt sich die Frage, welche Links durch die Interpolation mit einer Abweichung von mehr als 50% berechnet werden. Visualisiert sind diese fehlerhaft berechneten Links in Abbildung 4.4. Hierbei ist zu erkennen, dass sich zwischen Start- und Endpunkt eines jeden falsch berechneten Links ein Hindernis befindet, welches es dem Interpolationsverfahren nicht möglich macht anhand dieser Punkte eine korrekte Linkqualität zu bestimmen.

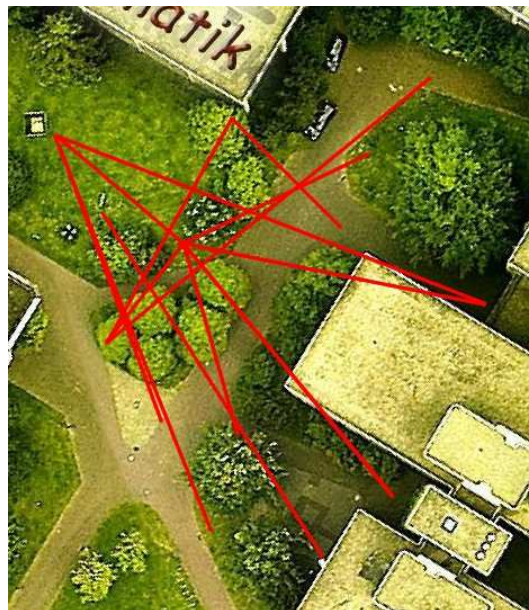


Abbildung 4.4: Fehlerhafte berechnete Linkqualitäten

Kapitel 5

Zusammenfassung und Ausblick

5.1 Zusammenfassung

Um Emulationen und reale Experimente im Bereich der mobilen ad-hoc Netzwerke besser vorbereiten zu können, wurden Werkzeuge und Algorithmen für die Vermessung eines Experimentierszenarios entwickelt. Mit Hilfe dieser Werkzeuge ist es möglich eine Vermessung zu planen, durchzuführen und auszuwerten.

Mapconfig unterstützt bei der Planung und erstellt die benötigten Konfigurationsdateien für die Endgeräte. Während der Planungsphase wird ein Raster über das Gebiet gelegt und Stützstellen zur Kommunikationssicherung platziert. Anhand der erstellten Dateien führt Mapkit dann die Positionierung der Geräte auf dem Messgebiet und die Vermessung durch. Um den Ablauf der Vermessung zu beschleunigen und zu vereinfachen wurde eine automatische Experimentsteuerung für diesen Anwendungsfall entwickelt, deren Kommunikation durch die Stützstellen sichergestellt wird.

Die gesammelten Messdaten können dann über eine bereitgestellte Schnittstelle ausgewertet werden. Dazu wurde ein Algorithmus entwickelt, der die Linkqualität zwischen den Messpunkten, die auf dem Raster liegen, interpoliert. Auf dieser Schnittstelle baut der Analyzer auf, dessen Aufgabe die Visualisierung der Messergebnisse ist.

Nach der Entwicklung der Werkzeuge wurde ein Gebiet auf dem Gelände der Universität Düsseldorf vermessen und die berechneten Linkqualitäten zwischen den Messpunkten mit denen von zusätzlichen Messungen verglichen. Das Ergebnis von einem Fehler unter 20% in 80% der Fälle erfreut uns, wirft aber auch Ideen zur Weiterentwicklung der Methode auf.

5.2 Ausblick

Es ist vorstellbar die Methode an zwei Punkten weiter zu entwickeln. Zum einen wäre es möglich an kritischen Punkten auf dem Messgebiet das Raster zu verkleinern. Kritische Punkte sind z.B. Bäume und Gebäude, bei denen man nach Analyse der Messdaten große Unterschiede zwischen benachbarten Messpunkten feststellt. In diesem Fall könnten die betroffenen Rasterfelder durch Kleinere ersetzt werden. Die Auswahl der kritischen Punkte ist mit einem gewissen Erfahrungsschatz auch vor der Durchführung der eigentlichen Messung vorstellbar, so dass ein entsprechender Algorithmus in Mapconfig integriert werden könnte.

Eine weitere Möglichkeit zur Verbesserung ist die Modellierung der Hindernisse. Hierbei würden die Koordinaten der Hindernisse vom User über eine GUI eingegeben und mit einem bestimmten Faktor, der die Abschattung der Funkwellen beschreibt, belegt werden. Die Interpolation könnte so nicht nur die Endpunkte des Links, sondern zusätzlich die Hindernisse, welche auf direktem Weg zwischen den beiden Punkten liegen, betrachten.

Literaturverzeichnis

- [gea06] *Google Earth*. Online Dokumentationen und Download unter <http://earth.google.de/>, 2006
- [gno06] *Gnome Project*. Online Dokumentationen und Download unter <http://www.gnome.org/>, 2006
- [gpe06] *GPE: The GPE Palmtop Environment*. Online Dokumentationen unter <http://gpe.handhelds.org/>, 2006
- [gtk06] *GTK+*. Online Dokumentationen unter <http://www.gtk.org/>, 2006
- [LNT02] LUNDGREN, Henrik; NORDSTRÖM, Erik; TSCHUDIN, Christian: Coping with communication gray zones in IEEE 802.11b based ad hoc networks. In: *Proceedings of the fifth ACM international workshop on Wireless mobile multimedia*. Atlanta, GA, September 2002, S. 49–55.
- [LYN⁺05] LIU, Jason; YUAN, Yougu; NICOL, David M.; GRAY, Robert S.; NEWPORT, Calvin C.; KOTZ, David; PERRONE, Luiz F.: Validation of Wireless Models Using Direct-Execution Simulation of Ad-Hoc Routing Protocols. In: *Simulation: Transactions of The Society for Modeling and Simulation International* (2005), January. <http://www.cs.dartmouth.edu/~dfk/papers/liu:jdirex.pdf>.
- [oz206] *OpenZaurus*. Online Dokumentationen unter <http://openzaurus.org/>, 2006
- [wis06] *Wi-Spy*. Online Dokumentationen und Bestellmöglichkeit unter <http://www.metageek.net/>, 2006

[zau04] *Sharp Zaurus SL-6000*. Details online unter http://www.mobiletechreview.com/sharp_zaurus_SL6000.htm, May 2004

Ehrenwörtliche Erklärung

Hiermit versichere ich, die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Düsseldorf, 24. Oktober 2006

Markus Kerper