

LOCATION BASED SERVICE (LBS) IM WLAN

eingereicht von:

Matthias Wessely
tm0310038078

DIPLOMARBEIT

zur Erlangung des akademischen Grades
Diplom Ingenieur (FH)
(Dipl. Ing. (FH))

Fachhochschule St. Pölten

Studienrichtung: Telekommunikation und Medien

Begutachter:

FH-Prof. Dipl. Ing. Johann Haag
Dipl.-Ing. (FH) Daniel Ruby

St. Pölten, im April 2008

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.
- ich dieses Diplomarbeitsthema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der von den Begutachtern beurteilten Arbeit überein.

St. Pölten, 12. April 2008

Matthias Wessely

Kurzfassung

Bei dieser Arbeit wird die Möglichkeit eines LBS mittels WLAN untersucht. Ziel daher ist es ein Testsystem zu entwickeln das unabhängig vom Klienten und ohne notwendige Installationen auf dem Zielsystem funktioniert.

Zu Beginn der Arbeit wird ein kurzer Einblick in die Grundlagen von WLAN und LBS gegeben. Dabei werden die verschiedenen Arten der Positionsbestimmung beschrieben und drei fertige LBS Systeme vorgestellt. Des weiteren werden die physikalischen Grundlagen elektromagnetischen Welle sowie ein Modell zur Wellenausbreitung untersucht.

Als nächstes folgt ein Einblick in neue Technologien. Hierbei handelt es sich um den Standard 802.11k sowie IPv6 im WLAN. In diesem Standard, der zur Zeit nur als Draft vorliegt, werden Methoden zur Verwaltung des Funkkanals definiert.

Der größte Teil der Arbeit beschäftigt sich mit der Erstellung eines LBS Systems im Festsaal der Fachhochschule St.Pölten. Hierzu werden zuerst verschiedene Algorithmen theoretisch beschrieben, ein kurzer Exkurs stellt dabei den RSSI Wert vor. Dieser Wert wird von WLAN Implementierungen benutzt um die Feldstärke eines Klienten anzuzeigen. Die Arbeit endet mit der Beschreibung der erstellten Teststellung im Festsaal. Das System erreicht eine Genauigkeit von 1,5 Meter.

Abstract

This work is the possibility of using wireless LBS. The aim therefore is a test system to be developed independently of the client and without the need for installation on the target system.

The work starts with is a brief insight into the fundamentals of WLAN and LBS. Here are the different types of position determination described, and three finished LBS systems introduced. In addition, the physical fundamentals electromagnetic wave and a model for wave propagation the field strength of a client.

Then follows an insight into new technologies. These are the standard 802.11k and IPv6 in WLAN. The 802.11k standard defines methods for the management of the radio channel which can be used to determine one's position.

The bulk of the work deals with the creation of an LBS system in the ballroom of the university of applied sciences St.Pölten. These are first described theoretically different algorithms, a short digression provides information about the RSSI value. The value will be used to appropriate the field strength of a client. The work ends with the description of the test created in the ballroom. The system achieved an accuracy of 1.5 meters.

Danksagung

Als erstes möchte ich FH-Prof. Dipl. Ing. Johann Haag danken welcher mir die Möglichkeit gegeben hat diese Diplomarbeit unter seiner Führung zu schreiben.

Weiter möchte ich mich bei Dipl.-Ing. (FH) Rainer Poisel bedanken, welcher mir mit vielen Ideen geholfen hat.

Zum Schluss möchte ich mich beim Team der Bibliothek der Fachhochschule St.Pölten bedanken, deren Bücher mir bei der Erstellung dieser Arbeit eine sehr große Hilfe waren.

“Wer nicht auf seine Weise denkt, denkt überhaupt nicht.”

–Oscar Wilde, engl. Schriftsteller, 1854-1900

St. Pölten, Österreich
12. April 2008

Matthias Wessely

Inhaltsverzeichnis

Ehrenwörtliche Erklärung	ii
Kurzfassung	iii
Abstract	iv
Danksagung	v
Abbildungsverzeichnis	x
Tabellenverzeichnis	xi
1 Einleitung	1
1.1 Problemstellung	1
2 Grundlagen	2
2.1 WLAN	2
2.1.1 Weitere Wireless LAN Funklösungen	3
2.1.2 Geschichte	5
2.1.3 Gesundheit	5
2.2 Location Based Service	6
2.2.1 Komponenten von LBS	7
2.2.2 Einsatzgebiete	8
2.2.3 Arten der Standortbestimmung	8
2.2.3.1 Techniken der Ortungsbestimmung	8
2.2.3.2 Eigenschaften bei der Gewinnung der Positionsinformation	10
2.2.4 Lösungen verschiedener Unternehmen	11
2.2.4.1 MS Radar	11
2.2.4.2 Intel Place Lab	13
2.2.4.3 Magic Map	16

2.3	Physikalische Grundlagen	18
2.4	Wellenausbreitung	19
2.4.1	Freiraumausbreitung	19
2.4.2	Empirisches Feldstärkemodell	21
2.5	Sicherheit	24
3	Neue Technologien im WLAN	25
3.1	802.11k	25
3.1.1	Beacon	27
3.1.2	Frame	27
3.1.3	Channel Load	27
3.1.4	Noise Histogram	27
3.1.5	STA Statistics	27
3.1.6	Location Configuration Information	28
3.1.7	Neighbor Report	28
3.1.8	Link Measurement	28
3.1.9	Transmit Stream /Category Measurement	28
3.1.10	Measurement Pause	28
3.1.11	Measurement Pilot	28
3.2	WLAN und IPv6	28
3.2.1	Bandbreitenversuch	29
4	Theoretische Grundlagen von LBS	31
4.1	Ausgangslage	31
4.1.1	Differenzmethode	32
4.1.2	Trilateration	33
4.1.3	Normalverteilung und Trilateration	35
4.1.4	Durchschnittsposition	36
4.1.5	Abstandsberechnung	37
4.1.6	Schnittpunkt zweier Kreise	38
4.2	Exkurs: Der RSSI Wert	40
4.2.1	FHSS Technologie	42
4.2.2	DSSS Technologie	42
4.2.3	RSSI und Verschlüsselung	43
4.3	Kalibrieren	44
4.3.1	Radiomap	44
4.3.2	Kalibrierungspunkt	45
4.3.3	Selbstkalibrierung der Umwelteinflüsse	45

4.3.4	Abstandsberechnung	46
4.4	Optimieren	46
5	LBS - Die Komponenten	48
5.1	LBS Systemarchitektur	48
5.2	LBS Server	49
5.2.1	Grundsystem	49
5.2.2	Datenbank	49
5.2.3	Perl	50
5.3	Access Points installieren	50
5.3.1	WLAN Treiber	50
5.3.2	Kismet	51
5.4	Erstellen der Radiomap - Offlinephase	51
5.5	Lesen der Feldstärke	53
5.6	Identifizierung des Klienten	54
5.7	Transport der Daten	56
6	Bestimmung der Position	58
6.1	Teststellung Festsaal	58
6.2	Aufbau	58
6.3	Referenzpunkte aufnehmen	59
6.4	Methode 1: Differenzmethode mit Mittelwerten	59
6.4.1	Vorgehensweise	63
6.4.2	Ergebnisse	63
6.5	Methode 2: Trilateration mit Mittelwerten	65
6.5.1	Vorgehensweise	65
6.5.2	Ergebnisse	66
6.6	Methode 3: Trilateration mit Normalverteilung	67
6.6.1	Vorgehensweise	69
6.6.2	Ergebnisse	70
6.7	Methode 4: Durchschnittsposition mit Normalverteilung	70
6.7.1	Vorgehensweise	70
6.7.2	Ergebnisse	71
6.8	Methode 5: Trilateration mit Abstandsberechnung	71
6.8.1	Vorgehensweise	72
6.8.2	Ergebnisse	72
6.9	Methode 6: Schnittpunkt zweier Kreise mit Mittelwerten	73
6.9.1	Vorgehensweise	74

6.9.2 Ergebnisse	74
6.10 Visualisierung	75
6.11 Ergebnisse	76
7 Ausblick	77
A Quellcode	78
A.1 Sniffer	78
A.2 Methode 1: Differenzmethode mit Mittelwerten	80
A.3 Methode 2: Trilateration mit Mittelwerten	81
A.4 Methode 3: Trilateration mit Normalverteilung	84
A.5 Methode 4: Durchschnittsposition mit Normalverteilung	88
A.6 Methode 5: Trilateration mit Abstandsberechnung	91
A.7 Methode 6: Schnittpunkt zweier Kreise mit Mittelwerten	93
A.8 GnuPlot	96
B Installationsanleitung	97
B.1 Grundsystem	97
B.2 Datenbank	97
B.3 Perl	98
B.4 Madwifi	98
B.5 Kismet	98
B.6 Lesen der Feldstärke	98
B.7 DHCP	99
B.8 PIX	99
B.9 Kismet Server starten	99
B.10 GnuPlot	100
Abkürzungsverzeichnis	101
Literaturverzeichnis	105

Abbildungsverzeichnis

2.1	LBS als Schnittmenge von Technologien [Sch07, vgl. s. 3]	6
2.2	LBS Komponenten [SS06, vgl. s. 4]	7
2.3	MS Radar Architektur	12
2.4	PlaceLab Architektur [AL05, vgl. s. 6]	14
2.5	Magic Map Klient [Ber04]	17
2.6	Elektromagnetische Welle [Zür05]	18
2.7	Freiraumausbreitung [Neu05, vgl. s. 8]	19
2.8	Regressionsmodell	23
3.1	TCP/IP-Protokollstapel	29
3.2	IPv6 Versuch	29
4.1	Differenzmethode Offlinephase	32
4.2	Trilateration [Wik08]	35
4.3	Durchschnittsposition	37
4.4	Abstandsmodell	39
4.5	Schnittpunkt zweier Kreise [Kow08]	39
4.6	WLAN Layer Eins und Zwei [Soc03, vgl. s. 28]	40
4.7	WLAN Funkstrecke [Neu06, vgl. s. 12]	42
4.8	DSSS Frameformat [Rec06, vgl. s. 99]	43
4.9	MAC Frameformat [Soc03, vgl. s. 34]	44
4.10	Selbstkalibrierung	45
5.1	Systemarchitektur	48
5.2	DHCP Relay	55
5.3	IPSEC Tunnel	56
6.1	Festsaal Aufbau	59
6.2	Festsaal linke Seite	60
6.3	Festsaal rechte Seite	60

6.4	AP1, AP4 und Webserver	60
6.5	AP3 und LBS Server	60
6.6	Referenzpunkte	61
6.7	Differenzmethode Positionsbestimmung	62
6.8	Abstände vom AP1 zu den Referenzpunkten findet man in Listing 6.7	66
6.9	Schnittpunkt zweier Kreise	75
6.10	GnuPlot ausgabe	76
B.1	Debian Installation - Softwareauswahl	97

Tabellenverzeichnis

2.1	Ausgangswerte des Rechenbeispiels	21
2.2	Regressions Messwerte	22
2.3	Koeffizienten des Regressionsbeispiels	23
3.1	IPv6 vs IPv4	30
4.1	Differenzmethode Offlinephase	32
4.2	Differenzmethode Klient	33
4.3	Differenzmethode Onlinephase	33
4.4	Normalverteilung Tabellenlayout	35
4.5	AP1 Abstände zu Referenzpunkten	36
4.6	Feldstärke in verschiedenen Entfernungen	38
4.7	Koeffizienten für das Abstandsverfahren	38
4.8	Gegenüberstellung der Koeffizienten	46
4.9	Gegenüberstellung der Feldstärken	46
5.1	Offlinephase Tabellenlayout	51
5.2	Data Tabellenlayout	52
5.3	Onlinephase Sniffer Tabellenlayout	54
6.1	Legende zu Abbildung 6.1	58
6.2	Beispiel - Differenzmethode Klient	63
6.3	Beispiel - Differenzmethode	64
6.4	Messpunkt 5 - Sniffer Tabellenlayout	64
6.5	Beispiel - Trilateration mit Mittelwerten - Aufenthalt	66
6.6	Beispiel - Trilateration mit Mittelwerten - Abstände	66
6.7	Normalverteilung Offlinephase	69
6.8	Beispiel - Trilateration mit Normalverteilung - Aufenthalt	70
6.9	Beispiel - Trilateration mit Normalverteilung - Abstände	70
6.10	Beispiel - Durchschnittsposition - Ergebnisse	71

6.11	Berechnete Abstandswerte	73
6.12	Beispiel - Abstandsberechnung - Zwischenergebnis	73
6.13	Beispiel - Schnittpunkt zweier Kreise - Abstände	74
6.14	Vergleich der Abweichungen	76

Kapitel 1

Einleitung

In dieser Diplomarbeit soll ein System zur Positionsbestimmung eines WLAN fähigen Klienten erarbeitet werden. Dabei werden verschiedene bereits bestehende Algorithmen der Positionsbestimmung miteinander verglichen und für die Verwendung im Outdoor Bereich adaptiert. Die Positionsbestimmung erfolgt dabei mittels Wireless LAN nach dem Standard IEEE 802.11. Im Mittelpunkt der Arbeit steht ein Gesamtsystem zum Auslesen der Feldstärke beziehungsweise der RSSI Werte der empfangenen Pakete.

1.1 Problemstellung

Auf dem Markt befinden sich verschieden Lösungen die jedoch alle mit einer Software am Klient arbeiten.

Diese Installation schränkt natürlich die Möglichkeiten für touristische Lösungen extrem ein. Bei dieser Arbeit soll ein System zum Auslesen der Feldstärke, unabhängig und ohne Installation beim Klienten, entwickelt werden. Die Anwender sind vor allem Touristen mit unterschiedlichste Endgeräte, wie z.B.:

- Laptop
- PDA
- Mobiltelefon
- PSP

Die Lokalisierung soll im bereits bestehenden frei zugänglichen WLAN der Stadt St.Pölten (Projekt an der Fachhochschule im Jahre 2007) zum Einsatz kommen.

Kapitel 2

Grundlagen

In diesem Kapitel wird ein kurzen Einblick in die Grundlagen von WLAN und LBS, den bereits existierenden fertigen Lösungen, der Geschichte und dem gesundheitlichen Aspekt von WLAN geben.

2.1 WLAN

Allgemein versteht man unter einem Wireless LAN ein drahtloses Funknetz. Jedoch wird Wireless LAN oft fälschlicherweise mit dem Standard IEEE 802.11 gleich gesetzt. Dieser Standard stellt nur eine mögliche Lösung eines WLANs dar, weiter werden im Abschnitt 2.1.1 beschrieben. Der Standard IEEE 802.11 beruht auf Standard IEEE 802.3 (Ethernet), hierbei werden aus Sicht des OSI Referenzmodells Anpassungen bei den Layern eins und zwei vorgenommen. Die höheren Schichten, Layer 3 (IP) beziehungsweise Layer 4 (TCP/UDP) bleiben unverändert. [Rec06, vgl. s. 5]

Wireless LAN arbeitet nach dem Standard 802.11 im Frequenzbereich von 2,4Ghz und kann sowohl Indoor als auch Outdoor verwendet werden. Die typischen Reichweiten von WLAN betragen dabei 10-300m, mit speziellen Antennen wenige Kilometer. [Rec06, vgl. s. 54]

Der WLAN Streckenrekord liegt zurzeit aber bei erstaunlichen 279km und wurde in Venezuela aufstellt. Hierbei kamen zwei handelsübliche APs der Firma Linksys mit dem Betriebssystem OpenWrt zum Einsatz. Als Antennen wurden zwei Parabolantenne aus der Satellitentechnik, mit einem Durchmesser von 2,74m und einem Antennengewinn von jeweils 32 dBi, verwendet. [Tri06, vgl. s. 1 - 2]

Der Grundstandard IEEE 802.11 wurde am 26 Juni 1997 verabschiedet. Zu diesem Grundstandard existieren eine Vielzahl an Erweiterungen und Arbeitsgruppen. Einige dieser Erweiterungen werden hier kurz aufgezählt. [Rec06, vgl. s. 5 - 9]

- 802.11b: Erweiterung der Datenrate auf 11Mbit/s

- 802.11a: Erweiterung auf das 5GHz Band und einer Datenrate von 54Mbit/s
- 802.11g: Erweiterung der Datenrate auf 54Mbit/s im 2,4GHz Band
- 802.11h: Anpassung des 5GHz Bandes aus regulatorischen Gründen
- 802.11d: System zum Informationsaustausch zwischen Klient und AP zur Anpassung an länderspezifische Gegebenheiten.
- 802.11e: Unterstützung von QoS
- 802.11f: Interoperabilität zwischen Basisstationen
- 802.11i: Erweiterung im Bereich Sicherheit und Authentifizierung
- 802.11k: Erweiterter Informationsaustausch zwischen Klient und AP, siehe Abschnitt 3.1

Neben diesen Erweiterungen existieren noch eine Vielzahl an Arbeitsgruppen die hier nicht weiter angeführt werden, da sie für die Erstellung eines LBS Systems keine relevante Rolle spielen.

2.1.1 Weitere Wireless LAN Funklösungen

Neben den 802.11 Standards gibt es noch weitere Funklösungen die einen drahtlosen Datenaustausch zur Verfügung stellen. Dabei handelt es sich um HiperLAN, HomeRF, Bluetooth, ZigBee und WiMax. [Rec06, vgl. s. 10] Im folgenden Abschnitt werden diese Wireless LAN Funklösungen kurz vorgestellt.

HiperLAN¹ wurde von ETSI, in der ersten Version 1996, im Standard EN-300652 spezifiziert und bot dabei eine maximale Datenrate von 23,529MBit/s. Das System arbeitet im 5GHz Bereich und verwendet als Modulationsverfahren GMSK, dieses Verfahren wird ebenfalls bei GSM eingesetzt. Im Jahre 2000 wurde nach einer zweijährigen Entwicklungsphase die zweite Version von HiperLAN im ETSI Dokument TR 101 683 spezifiziert. Dabei wurde die Datenrate auf 54Mbit/s erhöht und ein QoS System implementiert. Neben HiperLAN gibt es noch einige Erweiterungen zur "Hiper-Familie". Diese sind HiperACCESS und HyperLINK. [Rec06, vgl. s. 10 - 11]

HomeRF² ist eine speziell für Privatanwender entwickelte Technologie, die sich durch einfache und preiswerte Geräte auszeichnet. HomeRF bietet in der Version 1.2 eine

¹<http://de.wikipedia.org/wiki/HIPERLAN>

²<http://de.wikipedia.org/wiki/HomeRF>

maximale Datenrate von 1,6Mbit/s, diese wurde in der Version 2.0 auf 10Mbit/s gesteigert. Heutzutage ist HomeRF von WLAN 802.11 verdrängt worden. [Rec06, vgl. s. 12]

Bei Bluetooth³ handelt es sich um eine Funktechnologie die nur im Nahbereich, bis maximal 10m, arbeitet. Die Grundidee von Bluetooth war die Kabelverbindung von in der Nähe stehenden Geräten zu ersetzen. Hardware für Bluetooth ist sehr kompakt und weist eine geringe Leistungsaufnahme auf. Bluetooth wurde als Studie der Firma Ericsson, im Jahr 1994, ins Leben gerufen. Die eigentliche Entwicklung startete dann 1998. Seit dem wurden verschiedene Versionen von Bluetooth entwickelt, die neueste Version 2.1 wurde 2007 verabschiedet. Diese erreicht eine Datenrate von bis zu 3Mbit/s. Bluetooth arbeitet so wie 802.11 WLAN im 2,4GHz Frequenzbereich und benutzt als Übertragungsverfahren das FHSS Verfahren. Die Sendeleistung unterteilt sich in drei Klassen.

- Klasse 1: 20dBm
- Klasse 2: 4dBm
- Klasse 3: 0dBm

Heutzutage arbeiten Bluetooth Geräte meist mit der Klasse 3 mit der eine Datenübertragung bis zu 10m erzielt werden kann. [Rec06, vgl. s. 12 - 15]

ZigBee⁴ wurde für Steuerungs - und Überwachungssysteme sowie für die Vernetzung von intelligenten Sensoren entwickelt. Standardisiert wurde es von der IEEE in der Spezifikation 802.15.4. Es zeichnet sich durch geringe Kosten, hohe Zuverlässigkeit, geringe Leistungsaufnahme und hohe Sicherheit aus. In Europa wird das 2,4GHz Band genutzt. Bei ZigBee ist dieses Band in 16 Kanäle unterteilt. Dabei wird eine Datenrate von 250kBit/s erzielt. Die Reichweite eines ZigBee Systems liegt bei maximal 75m. [Rec06, vgl. s. 15 -16]

WiMax⁵ ist in der Grundversion im IEEE 802.16a Standard spezifiziert. Für die Umsetzung sind drei Stufen geplant. In der ersten Ausbaustufe sollen Gebäude mit schnellen Internetanbindungen versorgt werden. In der Ausbaustufe zwei soll der Einsatz von Indoor Antennen ermöglicht werden. Die dritte und letzte Ausbaustufe soll die Anbindung von mobilen Geräten wie Mobiltelefone und Notebooks ermöglichen. WiMax

³<http://de.wikipedia.org/wiki/Bluetooth>

⁴<http://de.wikipedia.org/wiki/ZigBee>

⁵<http://de.wikipedia.org/wiki/WiMax>

ist somit eine Weitverkehrstechnologie für die Abdeckung der „letzten Meile“⁶. WiMax soll Datenraten von bis zu 70MBit/s erreichen und eine Entfernung von bis zu 70km überbrücken können. Diese Werte gelten jedoch nur für den optimalen Fall von stationären Geräten die eine Sichtverbindung besitzen. Bei Mobiltelefonen und Laptops ohne einer Sichtverbindung verringert sich die Datenrate auf 20MBit/s und die Entfernung auf 600m. [Rec06, vgl. s. 16]

2.1.2 Geschichte

WLAN wie wir es heute kennen basiert auf dem ALOHA System. Das ALOHA System wurde auf der Universität von Hawaii entwickelt. Dies Ur Wireless LAN wurde verwendet um die Standorte der Universität, die sich auf mehrere Inseln verteilen, miteinander zu verbinden, dadurch konnte auf die relativ teure Nutzung von Telefonleitungen verzichtet werden. Erste kommerzielle Produkte kamen Anfang der 90er Jahre, von der Firma Lucent, auf den Markt. Diese Produkte hatten jedoch einige Nachteile, die erzielbare Datenrate und Reichweite waren sehr eingeschränkt und relativ teuer. Diese Probleme verhinderten eine Verbreitung von WLAN in den 90ern. Diesen Problemen nahm sich das IEEE an und verabschiedete 1997 den Standard IEEE 802.11. Zu einem der ersten Anbieter von kommerziellen Funklösungen nach IEEE 802.11 gehörte Ende der 90er die Firma Apple. [Rec06, p. 4]

2.1.3 Gesundheit

Immer wenn Daten beziehungsweise Informationen mittels Funk übertragen werden kommt das Thema Gesundheit ins Gespräch. Es stellt sich die Frage wie sich elektromagnetische Wellen auf den Menschen auswirken. Bekannt ist, dass elektromagnetische Wellen Wärme erzeugen. Hierbei gibt es drei maßgebliche Faktoren, diese Faktoren sind die Frequenz, die Leistung sowie die Entfernung zum Sender. Da die Leistung mit dem Quadrat der Entfernung abnimmt beträgt, bei einer Vergrößerung des Abstandes von 1cm auf 1m, die Leistung nur mehr 1/100 der ursprünglichen Leistung. Allgemein lässt sich festhalten das die Leistungen bei WLAN deutlich geringer ist als die von Mobilfunkeinrichtungen. Ein handelsübliches Mobilfunktelefon sendet mit einer Leistung von bis zu 2 Watt, ein AP ist auf 100mW, was einem Zwanzigstel der Leistung eines Mobiltelefons entspricht, beschränkt. Des weiteren kommt hinzu das ein Mobiltelefon meist direkt am Ohr betrieben wird im Vergleich zu einem Laptop mit PCMCIA WLAN Karte die einen durchschnittlichen Abstand von 60cm zum Kopf aufweist.

⁶Die letzte Meile stellt den Teil einer Telefon- oder Internetverbindung dar, die vom ISP zum Endkunden führt.

Im Jahre 2001 wurde vom Nova Institut für Ökologie und Innovation ein Gutachten erstellt das die Belastung durch die elektromagnetische Strahlung eines 2,4GHz WLANs untersuchte. Dieses Gutachten stellte fest das, bei einem mittleren Abstand von 60cm zum Notebook, die Leistungsflussdichte $3,15mW/m^2$ betrug und sich auf $158mW/m^2$ erhöht wenn man sich nur mehr 20cm vom Notebook entfernt befindet. Ein AP hat in einer Entfernung von 2,5m eine Leistungsflussdichte von lediglich $0.66mW/m^2$. Diese Werte liegen weit unter den Grenzwert der in Österreich $10W/m^2$ beträgt und durch die ÖVE/ÖNORM 8850 geregelt wird. [Rec06, p. 31 - 32]

2.2 Location Based Service

Systeme die Standortbasierte Dienste anbieten werden schon seit den 1980er Jahren entwickelt. Zur Positionsbestimmung kann GPS genauso wie z.B.: Plakate oder Verkehrszeichen herangezogen werden. In der Literatur gibt es mehrere Definitionen von LBS eine davon ist folgende.

LBS sind Informationsdienste, die von mobilen Geräten über das mobile Netzwerk abgerufen werden können und die Position des mobilen Gerätes zur Informationsgewinnung verwenden. [Sch07, s.3]

Eine weitere Definition beschreibt LBS als eine Schnittmenge von drei Technologien. Diese sind das Internet, geografische Informationssysteme mit räumlichen Datenbanken sowie Informations- und Kommunikationstechnologien wie mobile Telekommunikationssysteme und tragbare Geräte, siehe Abbildung 2.1

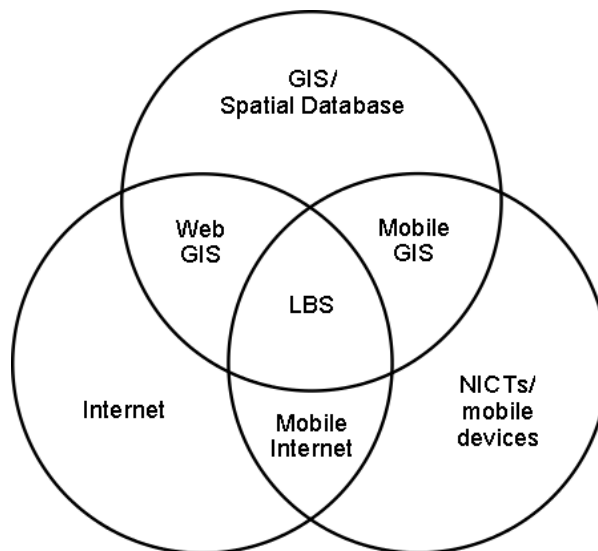


Abbildung 2.1: LBS als Schnittmenge von Technologien [Sch07, vgl. s. 3]

Eine entscheidende Grundlage bilden also geografische Informationssysteme die mit einer räumlichen Datenbank operieren. Entwickelt wurden solche Systeme ursprünglich für professionelle geografische Datenapplikationen. Viele dieser Systeme benötigen sehr große Rechenkapazitäten. Hier liegt der Unterschied zu LBS Systemen, da diese prinzipiell mit Einschränkungen wie beschränkter Rechenleistung, kleinen Displays bzw. Batterie Laufzeiten zu kämpfen haben. [Sch07, vgl. s. 4]

2.2.1 Komponenten von LBS

LBS Systeme erfordern verschieden Infrastrukturelemente die einen reibungslosen Ablauf ermöglichen sollten und dafür gut aufeinander abgestimmt sein müssen. Die Klienten verwenden ein **mobiles Gerät** um die benötigten Informationen vom Server zu erhalten. Das Gerät muss in der Lage sein, die Ergebnisse auf verschiedene Art und Weise darstellen zu können, z.B.: Sprache, Bilder, Texte. Für die Kommunikation zwischen Klient Endgerät und Dienstanbieter ist das **Kommunikationsnetzwerk** zuständig. Es übermittelt die Anfragen von mobilen Geräten zum Service Provider und die Ergebnisse zurück zum Klient.

Weiters benötigten LBS-Systeme eine **Komponente zur Positionsbestimmung** des Benutzers. Diese Positionsbestimmung kann entweder durch das Netzwerk oder durch das mobile Endgerät des Klienten erfolgen. Der **Dienst- und Anwendungsprovider** stellt in der Regel eine Vielzahl verschiedener Dienste zur Verfügung und ist für die Bearbeitung der Anfragen des Klienten zuständig. Dies kann die Positionsbestimmung oder das Finden von Routen sein. **Daten- und Inhaltsprovider** stellen diese Information zu Verfügung. [Sch07, vgl. s. 6-7]

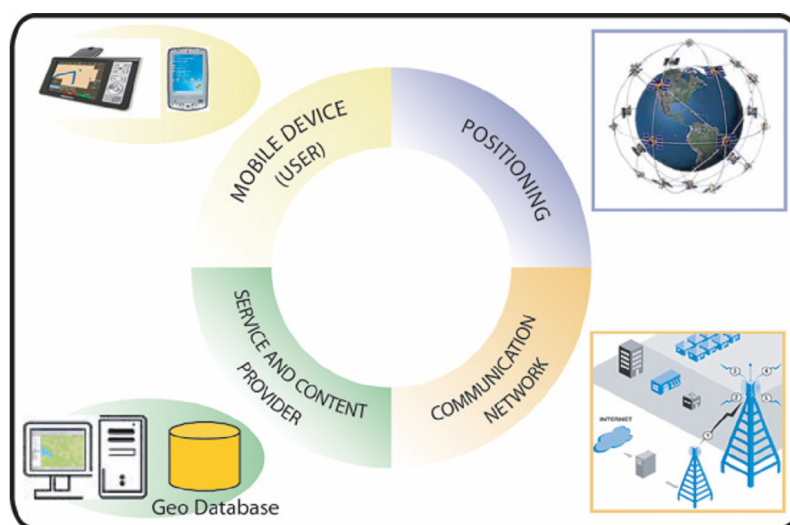


Abbildung 2.2: LBS Komponenten [SS06, vgl. s. 4]

2.2.2 Einsatzgebiete

Prinzipiell kann zwischen zwei Arten von Bereitstellung von Diensten unterschieden werden, beide Arten bieten dabei eine Vielzahl von Anwendungsmöglichkeiten. Die erste Art der Bereitstellung sind Pull-Services. Diese müssen vom Nutzer direkt angefordert werden. Ein Beispiel dazu ist das Aufrufen einer Webseite. Die andere Art sind Push-Services, diese liefern Informationen die nicht oder nur indirekt vom Benutzer angefragt wurden. Solche Dienste werden meist durch ein bestimmtes Ereignis ausgelöst. Einige Beispiele zählt folgende Liste auf. [Sch07, vgl. s. 12 - 13]

- Museumsführer
- Stadtinformationssystem
- Routenplaner
- Produkt Tracking
- Personenfinder
- Mobile Spiele / Geocaching

2.2.3 Arten der Standortbestimmung

Positionierungssysteme lassen sich Anhand der Technik zur Ortsbestimmung sowie Anhand der Eigenschaften bei der Gewinnung der Positionsinformationen einteilen. [Jeh07, p. 2ff]

2.2.3.1 Techniken der Ortungsbestimmung

- Nachbarschaftsbeziehung
Hierbei wird die physikalische Nähe zu einem oder mehreren Referenzpunkten benutzt um die Position zu bestimmen.
- Triangulation
Hierbei werden mindestens 2, für Ortungen im Raum 3, fixe Punkte benötigt. Durch die Messung der Winkel und Entfernung von den Fixpunkten zum Objekt kann dessen Position bestimmt werden.
- Trilateration
Die Trilateration benötigt in der Ebene mindestens 3, im Raum mindestens 4, bekannte Fixpunkte. Kennt man die Entfernung zu jedem Fixpunkt ist man in der Lage den Schnittpunkt der drei Kreise die sich aus den Abständen zum Objekt ergeben zu errechnen.

- Fingerprinting

Hierbei wird der Bereich in dem die Positionsbestimmung zum Einsatz kommen soll in einer Offlinephase vermessen. Die gewonnen Daten (Signalstärke und/oder Signal-Rausch-Abstand) werden in einer Datenbank abgelegt. Will man nun die Position eines Klienten bestimmen muss man nur die aktuellen Daten mit denen in der der Datenbank vergleichen.

- Szenenanalyse

Die Szenenanalyse bestimmt die Position eines Klienten anhand äußerer Umgebungsmerkmale. Diese Merkmale sind z.B.: Signalstärken von Referenzpunkten, Kameraaufnahmen oder Ultraschallmessungen.

- Time of Arrival

Bei der ToA Methode werden die Signallaufzeiten zur Berechnung der Entfernung verwendet. Die Laufzeiten werden hierbei direkt in Abstände umgerechnet. Das große Problem dieser Methode ist das die Uhren der Sender und Empfänger synchron laufen müssen.

- Time Differenc of Arrival

TDoA arbeitet ähnlich wie ToA jedoch werden hier nur die Unterschiede der Empfangszeiten zur Berechnung der Position herangezogen. Somit ist das Problem gelöst das die Uhren der Sender und Empfänger synchron laufen müssen, die Uhren der Empfänger müssen zueinander jedoch immer noch Synchron sein.

- Phase des Trägersignals

Hierbei wird die Phase des internen Referenzsignals mit der Trägerphase des empfangenen Signals verglichen. Die Phasenunterschiede geben Aufschluss über die Entfernung des Klienten.

- Received Signal Strength Indicator

Die RSSI Methode macht sich die Signalstärke zu nutzen. Dadurch das die Signalstärke mit dem Quadrat des Abstandes abnimmt kann direkt auf die Entfernung eines Klienten geschlossen werden. Ein großes Problem dieser Methode sind immer auftretenden Störungen wie z.B.: Menschen, Gebäude oder Umwelteinflüsse.

- Angel of Arrival

Bei AoA werden die Einfallswinkel des Signals an den Basisstationen zur Positionsbestimmung verwendet.

- Cell of Origin

Die CoA Methode ermittelt nur die Zelle in der sich ein Klient befindet. Dieses Verfahren hängt stark von der Zellengröße ab und ist somit wie am Beispiel GSM zu sehen sehr ungenau. Die Genauigkeit liegt hierbei in einem Bereich von 100m bis 20km.

2.2.3.2 Eigenschaften bei der Gewinnung der Positionsinformation

- Geometrisch vs. symbolische Position

Mit geometrischer Position sind Längen-, Breiten-, Höhengrade gemeint. Symbolische Positionen sind z.B.: 1.Stock eines Gebäudes oder ein bestimmter Platz in einer Stadt

- Absolute vs. relative Position

Bei einer absoluten Positionsbestimmung ist man nicht wie bei der relativen Bestimmung von einem vorhergehendem Messpunkt bzw. einem fixen Bezugspunkt abhängig.

- Selbstortende vs. vernortende Systeme

Bei einem selbstortenden System bestimmt der Klient seine Position selbst, bei einem fernortenden übernimmt diese Aufgabe ein Server.

- Genauigkeit und Konsistenz

Systeme müssen die Position möglichst exakt z.B.: wenige Meter angeben können und dürfen dabei nicht ausfallen.

- Umfang

Mit Umfang ist die räumliche Ausbreitung des Systems sowie die Anzahl der zu verwalteten Klienten gemeint.

- Fähigkeit zur Wiedererkennung

Ein System kann auch anhand der Wiedererkennung eines Klienten klassifiziert werden.

- Kosten

Neben der Wiedererkennung spielen auch die Kosten eine große Rolle bei der Klassifizierung, wobei neben den ersten Installationskosten die Instandhaltungskosten einen wesentlichen Punkt darstellen.

- Beschränkungen

Unter Beschränkung ist das Einsatzgebiet eines Positionierungssystems zu ver-

stehen, so ist es z.B.: GPS nicht möglich, aufgrund von Dämpfungen, Ortungen Indoor vorzunehmen.

2.2.4 Lösungen verschiedener Unternehmen

Zur Zeit gibt es bereits einige Lösungen die LBS im WLAN anbieten. Einige dieser Lösungen werden hier aufzählen.

- Microsoft Radar
- MIT Cricket
- Aero Scout
- Ekahau
- Intel Place Lab
- Magic Map
- Horus
- Ein System von Artur Szankowski, siehe [[Sza07](#)]

Alle diese Lösungen basieren auf dem Prinzip der Verwendung einer Klientsoftware. Bei dem vorhanden System kann dieses Prinzip jedoch nicht eingesetzt werden da die Klienten meist Touristen sind die keine zusätzliche Software installieren beziehungsweise auch nicht administrieren sollen. Da es sich um viele verschiedene Endgeräte wie Laptops, Handys oder PDAs handelt wäre eine Programmierung für jedes erdenkliche Gerät mit einem sehr hohen Aufwand verbunden.

2.2.4.1 MS Radar

MS Radar wurde von Paramvir Bahl and Venkata N. Padmanabhan bei Microsoft Research im Jahr 2000 entwickelt. Es ist eines der ersten Indoor Positionierungs- und Trackingsystem das mit 802.11 WLAN arbeitet. Es erlaubt WLAN fähigen Endgeräten ihre Position anhand von Signalstärkemessungen von APs zu bestimmen. Das System benutzt dabei vorhandene WLAN Infrastruktur und benötigt keine zusätzlichen Komponenten. Das System berechnet die Position eines Klienten anhand zweier Verfahren. Das erste Verfahren ist ein empirisches Verfahren auf der Basis von Fingerprints die in einer Offlinephase angelegt werden. In der Onlinephase werden diese Fingerprints verwendet um Vergleich anzustellen. Das zweite Verfahren beruht auf mathematischen Methoden die ebenfalls in einer Offlinephase Referenzpunkte bestimmen und als Radiomap speichern. [[PB00](#), vgl. s. 1] und [[W.K06](#), vgl. s. 249 - 250]

Abbildung 2.3 zeigt die Grundarchitektur von MS Radar. Auf den APs arbeitet ein von Microsoft Mitarbeitern selbst entwickelter WLAN Treiber namens HostAP.

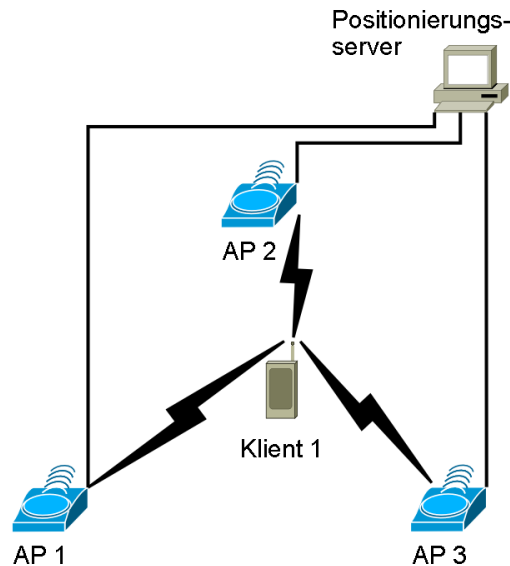


Abbildung 2.3: MS Radar Architektur

Das empirische sowie das mathematische Verfahren benutzen die „nearest neighbor in signal space“ Technik. Bei dieser Technik wird unter Zuhilfenahme der euklidischen Distanz, die geringste Abweichung von empfangenen Werten und Fingerabdrücken aus der Radiomap, die Position des Clients bestimmt. [Jeh07, vgl. s 7]

Die empirische Methode erhält Positionsinformationen durch die Auswahl des Fingerprints der der aktuellen Signalstärke am nächsten kommt oder durch die Berücksichtigung mehrere Fingerprints. Das Ziel ist die Minimierung der euklidischen Distanz die durch die Formel 2.1 beschrieben wird. [Jeh07, vgl. s 7-8]

$$d(E, M) = \sqrt{\sum_i^n (E_i - M_{i=1})^2} \quad (2.1)$$

E_i ist der Wert der die Signalstärke des i-ten APs des empfangenen Signals. M_i gibt den Signalstärkewert des i-ten APs innerhalb der Datenbank an. Der Wert n steht für die Gesamtzahl an APs.

Um die Genauigkeit bei sich bewegenden Objekten zu verbessern haben die Forscher einen Viterbi-artigen Algorithmus ⁷ entwickelt der davon ausgeht das sich Objekte nicht sprunghaft bewegen, er stellt quasi eine Art Gedächtnis dar. Des weitem wurden

⁷<http://de.wikipedia.org/wiki/Viterbi-Algorithmus>

mehrere Radiomaps erstellt um auf unterschiedliche Gegebenheiten, wie die Anzahl der Personen innerhalb des Gebäudes, einzugehen. [Jeh07, vgl. s 8-9] und [PB00, vgl. s. 4-6]

Im Gegensatz zum empirischen Modell werden bei der mathematischen Methode keine Referenzpunkte aufgenommen, es wird stattdessen versucht mit mathematischen Modellen die Signalausbreitung zu berechnen. Ein großes Problem dieser Methode stellen die Eigenschaften von elektromagnetischen Wellen innerhalb von Gebäuden dar. Diese unterliegen der Reflektion, der Streuung, der Beugung sowie einer Dämpfung. Ein zweites Problem stellt die Mehrwegeausbreitung bzw. der Mehrwegeempfang des Signals dar. Das System von Microsoft verwendet einen Kompromiss aus Genauigkeit und Einfachheit. Zum Einsatz kommt das Floor Attenuation Factor Modell von Scott Y. Seidel und Theodore S. Rappaport. Dabei wird das Modell insofern abgewandelt das die Etagendämpfungen aus der Formel genommen werden und stattdessen Dämpfungen durch Hindernisse, wie Wände, berücksichtigt werden. Die Formeln, zum nunmehr, Wall Attenuation Factor (WAF) Modell zeigen die Formeln 2.2 und 2.3.

$$P(d)[dBm] = P(d_0)[dBm] - 10 * n * \log \frac{d}{d_0} - nW * WAF \mid nW < C \quad (2.2)$$

$$P(d)[dBm] = P(d_0)[dBm] - 10 * n * \log \frac{d}{d_0} - C * WAF \mid nW \geq C \quad (2.3)$$

Die Variablen in den Formeln 2.2 und 2.3 haben folgende Bedeutung. n steht für den Steigungsgrad des Pfadverlustes mit der Distanz. $P(d_0)$ gibt die Signalstärke einer Referenzdistanz d_0 an. Die Distanz zwischen Sender und Empfänger gibt die Variable d an. Das C steht für die maximale Anzahl an Hindernissen ab denen sich der Dämpfungsfaktor ändert. nW gibt die Anzahl der Hindernisse zwischen Sender und Empfänger an. WAF stellt den Wall Attenuation Factor dar.

Die Werte n und WAF sind empirische Messwerte die Abhängigkeiten von Gebäudegrundriss und dessen Material darstellen. Der Wert für $P(d_0)$ kann empirische ermittelt werden aus der Spezifikation des WLAN Adapters entnommen werden. [Jeh07, vgl. s. 9]

2.2.4.2 Intel Place Lab

Neben Microsoft Research hat auch Intel Research ein deterministisches Verfahren zur Positionsbestimmung erarbeitet, es nennt sich Intel Place Lab. Das Einsatzgebiet ist sowohl Indoor als auch Outdoor. Das Hauptaugenmerk lag bei der Entwicklung auf

ständiger und allgegenwärtiger Verfügbarkeit und nicht auf Genauigkeit. Ebenso wie bei Radar wird die Position mithilfe der Signalstärke und Radiomaps bestimmt. Es erfolgt ebenfalls eine Aufteilung in Offline und Onlinephase. Der große Unterschied liegt in der Trainingsphase beziehungsweise Offlinephase. Place Lab benutzt im Gegensatz zu Radar Daten aus Wardrives. Unter Wardriving versteht man das aufspüren von APs mittels Auto, Laptop und GPS Empfänger. Dabei wird die Position, der Netzwerkname (BSSID), die Feldstärke sowie die MAC Adresse jedes APs aufgenommen. Zur Erstellung der Radiomap werden diese Daten herangezogen. In der Onlinephase werden die Daten aus der Radiomap mit den aktuellen Daten verglichen und so die Position bestimmt. [Jeh07, vgl. s. 11] und [W.K06, vgl. s. 242 - 245]

Abbildung 2.4 zeigt eine grobe Übersicht der Architektur von Intels Place Lab. Hierbei ist der Wardriver⁸ beziehungsweise Warwalker zu sehen der die Radio Beacons der 802.11 Infrastruktur aufnimmt und in der Beacon Datenbank speichert. Sind von einer Stadt genug Beacons in der Datenbank aufgenommen lädt sich der Klient diese herunter und ermittelt damit seine Position.

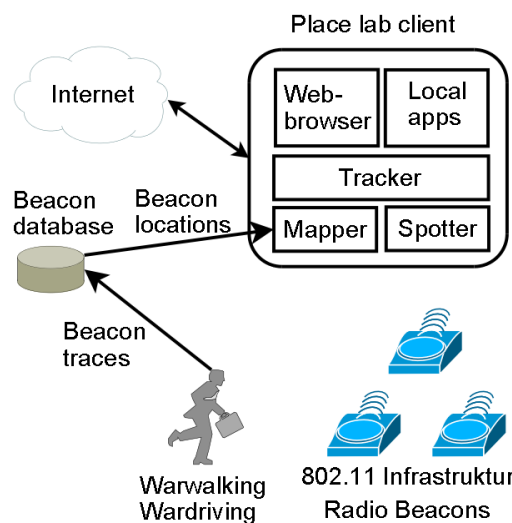


Abbildung 2.4: PlaceLab Architektur [AL05, vgl. s. 6]

Place Lab setzt bei der Positionsbestimmung die Algorithmen Centroid, Fingerprinting und Particle Filter ein. Die ersten beiden Algorithmen sind deterministisch der letzte verfolgt einen probabilistischen Ansatz.

- Centroid

Der erste Algorithmus ist zugleich der Einfachste dieser drei. Bei Centroid wird

⁸<http://de.wikipedia.org/wiki/Wardriving>

jeder AP mit seiner Position, die in Form von GPS Daten vorliegt, aufgenommen. Neben der Position wird auch noch das arithmetische Mittel der Feldstärke aller Empfangenen Pakete gespeichert. Die Position des Klienten ist der Mittelpunkt aller zu einem Zeitpunkt sichtbaren APs. Wie man sich vorstellen kann ist die sehr ungenau. Zur Verbesserung des Verfahrens kann eine Gewichtung der APs, anhand der Signalstärke, vorgenommen werden. Neben der Signalstärke können noch anderen Faktoren für die Gewichtung verwendet werden. [Glo05, vgl. s. 4]

- Fingerprint

Das Fingerprint Verfahren von Intel ist ähnlich dem von Microsoft. Es werden an verschiedenen Orten Referenzmessungen vorgenommen und gespeichert. Somit ist für jeden Referenzpunkt die Feldstärke und MAC Adresse aller APs in Sichtweite bekannt, diese Daten bilden einen Fingerabdruck. Als Fingerabdrücke können auch die Daten aus dem Centroid Verfahren verwendet werden. Die Berechnung der Position stellt sich nun wie folgt dar. Der Klient sammelt alle Pakete von umliegenden APs und entnimmt ihnen die MAC Adresse sowie die Feldstärke. Danach durchsucht der Klient die Datenbank der Fingerabdrücke nach eben diesen APs und liest die Feldstärke aus. Für jeden Eintrag wird, wie bei MS Radar, die euklidische Distanz, siehe Formel 2.1, berechnet. Unter Berücksichtigung von k-nearest neighbor(s) in signal space und der euklidischen Distanz wird die Position bestimmt. Als k hat sich ein Wert von vier als am besten funktionierend gezeigt. Als Verbesserung des Verfahrens können Einträge verwendet werden, die entweder einen Sender weniger oder einen zusätzlich aufweisen. Das ist wichtig, da bei einem defekten oder deaktivierten Sender mehrere Einträge unbrauchbar werden. [Glo05, vgl. s. 4 - 5]

- Particle Filter

Beim Particle Filter kommt eine abgewandelte Form eines Bayes Filter⁹ zum Einsatz. Ein Bayes Filter hat die Aufgabe probabilistisch den Zustand eines dynamischen Systems zu bestimmen unter dem Aspekt das die erhaltenen Daten Rauschen aufweisen. Im Falle von Place Lab ist der Zustand mit dem Ort gleichzusetzen. Ein Bayes Filter stellt die Unsicherheit des Zustandes zu jedem Zeitpunkt mit der Wahrscheinlichkeitsfunktion $Bel(x_t)$ dar. $Bel(x_t)$ wird definiert als $p(x_t|z_{1:t})$, hierbei stellt $z_{1:t}$ die Menge der Sensordaten dar. Es wird also die Wahrscheinlichkeit gesucht mit der sich der Klient zum Zeitpunkt t am Ort x aufhält, unter der Voraussetzung das vorher Sensordaten $z_{1:t}$ gesammelt wurden.

⁹<http://seattle.intel-research.net/people/jhightower/pubs/fox2003bayesian/fox2003bayesian.pdf>

Um im laufenden Betrieb nicht an der Datenflut zu ersticken wird das Prinzip von Markov ¹⁰ angewendet. Praktisch bedeutet das, dass nur aktuelle Messungen und der Aufenthaltsort zum Zeitpunkt $t-1$ in die Berechnung eingehen. Der Algorithmus besteht nun aus 2 Phasen. Als erstes wird eine Vorhersage getroffen die aussagt wie man sich seit dem letzten Zeitschritt bewegt hat.

$$Bel^-(x_t) \leftarrow \int p(x_t|x_{t-1})Bel(x_{t-1})dx_{t-1} \quad (2.4)$$

$p(x_t|x_{t-1})$ stellt die Systemdynamik dar, diese muss manuell angepasst werden. Dieses Systemdynamik sagt aus wie sich ein Klient am wahrscheinlichsten bewegt. Die zweite Phase nennt sich Korrektur.

$$Bel(x_t) \leftarrow \alpha_t p(t_t|x_t)Bel^-(x_t) \quad (2.5)$$

$p(t_t|x_t)$ ist hier das Wahrnehmungsmodell, es beschreibt die Wahrscheinlichkeit die Beobachtung z_t am Ort x_t zu machen. Der Wert α_t ist eine Normalisierungskonstante. Der Algorithmus führt wiederholt beide Phasen aus, sodass der Klient für jeden Ort die Wahrscheinlichkeit kennt sich dort aufzuhalten. [Glo05, vgl. s. 5]

2.2.4.3 Magic Map

Magic Map verwendet wie die beiden Vorhergehenden Positionierungssysteme die Feldstärke zur Positionsbestimmung. Dabei werden aber nicht nur die APs in Reichweite verwendet sondern auch andere Klienten. Es ist jedoch geplant neben der Feldstärke auch das SNR zur Verbesserung der Ergebnisse heranzuziehen. Dabei verwendet Magic Map nicht einen Wert zur Positionsbestimmung sondern einen über fünf Messungen gemittelten Wert. [PI04, vgl. s. 1]

Die Entwickler an der Humboldt Universität zu Berlin haben dabei das sogenannte JUNG Framework erstellt. Dieses Framework übernimmt die Aufgabe der Positionsbestimmung. Dabei arbeitet dieses System mit einer grafischen Oberfläche bei der durch einfaches anklicken Knoten eingefügt werden können. Diese Knoten benötigen keine absolute Positionierung, es muss nur die Kantenlänge zwischen den einzelnen Knoten eingestellt werden. Das Sprint Layout übernimmt die Positionierung der Knoten, wobei es darauf achtet das die Kantenlängen möglichst wenig von ihren eingestellten Längen variieren. [Sza07, vgl. s. 42]

¹⁰<http://de.wikipedia.org/wiki/Markow-Kette>

Abbildung 2.5 zeigt das Frontend eines Magic Map Klient. Zu sehen sind dabei die APs als rote Quadrate, Klienten als grüne Quadrate und Referenzpunkt als blaue Quadrate mit einem Anker in der Mitte.

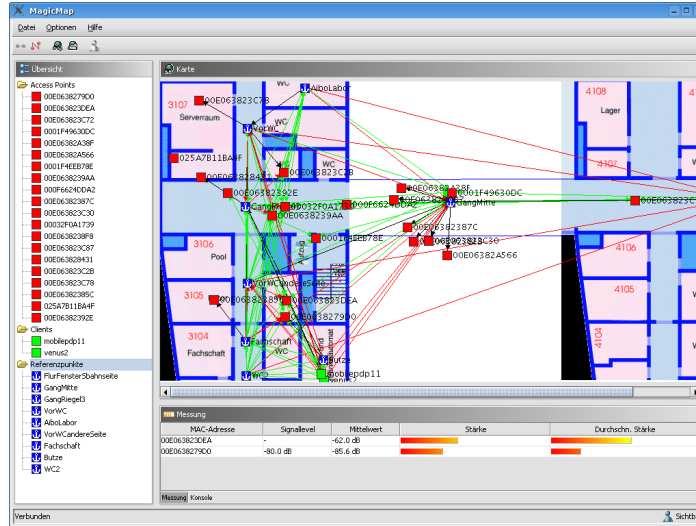


Abbildung 2.5: Magic Map Klient [Ber04]

Bei Magic Map werden die farbigen Quadrate als Knoten bezeichnet. Die Klient Knoten können die Signalstärke der APs abrufen und somit ihre Position bestimmen. APs müssen mit Klienten beziehungsweise Referenzpunkten über Kanten bestimmter Länge verbunden werden. An den Referenzpunkten messen die Klienten die Feldstärke aller umliegenden APs und speichern sie als einen Fingerabdruck in einer Datenbank ab. [Sza07, vgl. s. 43]

Mathematische Grundlagen Jeder Klient misst die Signalstärke aller APs in Reichweite und bildet über mehrere Messungen einen Mittelwert um stochastische Einflüsse auszugleichen. Zur Mittelwertbildung hat sich ein Wert von fünf Messungen als am praktikabelsten herausgestellt. Die Ausbreitung des Funksignals wird als Kugelförmig angenommen, dabei beschreibt der Radius r dieser Kugel den Abstand von AP und Klient. Die Annahme einer kugelförmigen Ausbreitung stimmt dabei nicht ganz da Antennen in der Horizontalen mehr abstrahlen als in der Vertikalen. Die Wahrscheinlichkeit p das sich ein Klient in der Entfernung r vom AP aufhält liefert die Formel 2.6. [Ber07, vgl.]

$$p = \frac{1}{\sqrt{2 * \pi * \sigma^2}} * e^{-\frac{(r-\mu)^2}{2 * \sigma^2}} \quad (2.6)$$

Dabei wird für die Varianz σ^2 ein Wert aus der Entfernung bestimmt. Da sich die Feldstärke in AP Nähe rasch ändert sollte der Wert für die Varianz hier klein sein im

Gegensatz zu weiten Entfernungen wo sich die Feldstärke nur mehr langsam ändert sollte die Varianz größer gewählt werden. Der Erwartungswert μ wird durch eine geeignete Entfernung aus der Signalstärke bestimmt. Wie das geschieht wird unter Punkt 2.4.2 beschrieben. Der Wert für r in der Formel 2.6 wird durch die folgende Formel ausgedrückt. [Ber07, vgl.]

$$r = k * \sin(v) * 10^{-\frac{s}{10}} \quad (2.7)$$

Da die Koeffizienten der Formel 2.7 ein hohes Maß an mathematischem Wissen voraussetzen sei hier auf die Seite ¹¹ verwiesen die die Koeffizienten der Formel 2.7 genau beschreibt.

2.3 Physikalische Grundlagen

Im Zusammenhang mit WLAN kommt oft der Begriff elektromagnetische Welle vor. Hierbei handelt es sich um die räumliche und zeitliche Ausbreitung einer Anregung in einem Medium. Diese Welle kann durch eine Funktion die Orts und Zeitabhängig ist beschrieben werden. [Alk04, p. 7]

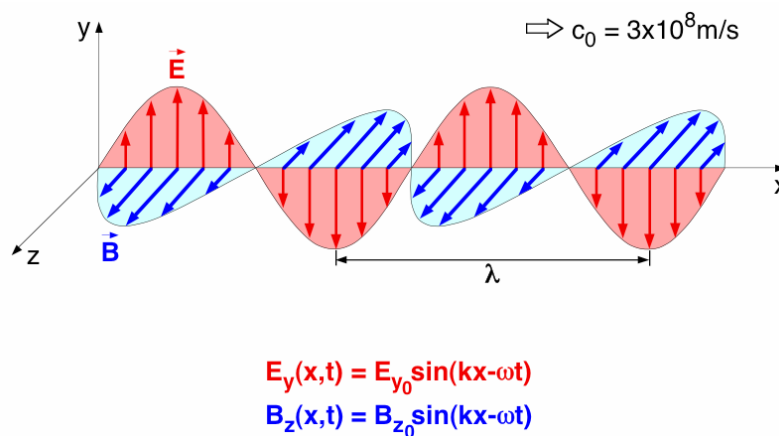


Abbildung 2.6: Elektromagnetische Welle [Zür05]

Wellen werden durch ihre Amplitude, Phase und Frequenz beschrieben. Die Frequenz (f) hängt mit der Wellenlänge (λ) über folgende Formel zusammen:

$$\lambda = \frac{c}{f} \quad (2.8)$$

Wobei c die Ausbreitungsgeschwindigkeit ist die näherungsweise mit c_0 , der Ausbreitungsgeschwindigkeit von Licht im Vakuum, gleich gesetzt werden kann. E und B

¹¹http://wiki.informatik.hu-berlin.de/nomads/index.php/MagicMap_Verfahren

geben das Elektrische bzw. Magnetische Feld an.

Informationen können in die Phase, Amplitude oder auch Frequenz moduliert werden. Die Energie einer Welle kann von einer Empfangsstation gemessen werden. Neben der eigentlichen Welle, die die Information trägt, kommen meistens noch andere wie z.B.: Mikrowellen, Mobiltelefone oder auch Reflexionen vor. Diese Störungen, die das Empfangssignal beeinflussen, heißen Rauschen und sollen möglichst gering sein. Das Verhältnis zwischen Empfangssignal und Rauschen wird im Signal-Rauschabstand ausgedrückt.

$$SNR = \frac{\text{Nutzsiganlleistung}}{\text{Rauschleistung}} \quad (2.9)$$

2.4 Wellenausbreitung

Die Elektromagnetische Wellenausbreitung wird durch die vier maxwellschen Gleichungen ¹² beschrieben. Aufgrund ihrer Komplexität werden die maxwellschen Gleichungen für den praktischen Einsatz durch Modelle ersetzt. Dabei gibt es folgende Arten von Modellen:

- Empirische Modelle
- Beugungsmodelle
- Strahlverfolgungsmodelle (Ray tracing Modelle)

Unter Punkt 2.4.2 wird ein selbst erstelltes empirisches Modell beschrieben. Allgemein lässt sich sagen das die Reichweite einer Welle mit zunehmender Frequenz abnimmt. [Neu05, vgl. s. 4 - 6]

2.4.1 Freiraumausbreitung

Die Freiraumausbreitung beschreibt den idealen Fall bei dem sich zwischen Sender und Empfänger keine Hindernisse befinden, siehe Abbildung 2.7

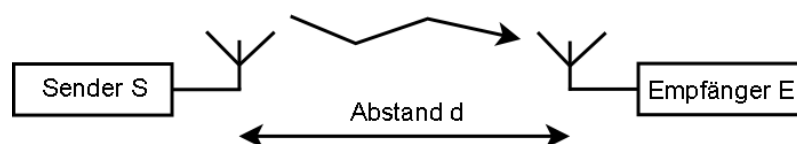


Abbildung 2.7: Freiraumausbreitung [Neu05, vgl. s. 8]

¹²siehe [PAT04, s. 970]

Der Pfadverlust stellt die Differenz zwischen ausgesendeter und empfangener Leistung dar. Bei WLAN wurde zu der zulässige Pfadverlust bei 2Mbit/s auf -80dBm festgelegt. [Rec06, vgl. s. 110]

Die Hersteller von WLAN Endgeräten geben meist eine maximal Reichweite, im freien Raum, von 300m an. Wie dieser Wert Zustande kommt wird hier kurz Beschrieben. Die Formel 2.10 beschreibt die Freiraumdämpfung.

$$L = \left(\frac{\lambda}{4 * \pi * d} \right)^2 \quad (2.10)$$

Die Formel 2.11 gibt die EIRP¹³ an. Diese spielt im regulatorischen Umfeld eine Rolle. Wobei P_s die Leistung am Sender ist und g_s der Gewinn der Sendeantenne.

$$P_s * g_s = EIRP[W] \quad (2.11)$$

Allgemein ist die Leistung am Empfänger P_e in Formel 2.12 definiert.

$$P_e = \frac{P_s}{4 * \pi * d^2} * A[W] \quad (2.12)$$

Dabei ist die Wirkfläche A aus der Formel 2.12 bei einer idealen isotropen Antenne.

$$A = \frac{\lambda^2}{4 * \pi} \quad (2.13)$$

Aus den Formeln 2.12 und 2.13 ergibt die die Leistung am Empfänger wie folgt.

$$P_e = g_s * g_e * \frac{P_s}{4 * \pi * d^2} * \frac{\lambda^2}{4 * \pi} = P_s * g_s * g_e * \left(\frac{\lambda}{4 * \pi * d} \right)^2 \quad (2.14)$$

Hierbei ist g_s der Gewinn der Sendeantenne und g_e der Gewinn an der Empfangsantenne.

Ein Praktisches Beispiel soll diese Formeln verdeutlichen. Für das Beispiel werden die Ergebnisse mit den Werten eines Cisco 851w Router¹⁴ verglichen. Es werden folgende Werte angenommen:

Mit diesen Werten ergibt sich die maximale Sendeleistung am Empfänger wie in Formel 2.15 beschrieben.

$$P_s = \frac{EIRP}{g_s} = \frac{100mW}{2^{10/10}} = 50mW \quad (2.15)$$

¹³EIRP gibt die Leistung an mit der ein Kugelstrahler versorgt werden muss um im Fernfeld die gleiche Feldstärke zu erzeugen wie eine Richtantenne.

¹⁴http://www.cisco.com/en/US/prod/collateral/routers/ps380/ps6195/product_data_sheet0900aecd8028a9a9_ps380_Products_Data_Sheet.html

VARIABLE	WERT
f	2,4 GHz
g_e	0 dBi
g_s	2 dBi
EIRP	100mW

Tabelle 2.1: Ausgangswerte des Rechenbeispiels

Zum besseren Verständnis kann man die 50mW in dBm umrechnen wie in Formel 2.16 zu sehen.

$$P_{s[dBm]} = 10 * \log\left(\frac{P_s}{1mW}\right) = 10 * \log\left(\frac{50mw}{1mW}\right) = 16,98dBm \quad (2.16)$$

Die ausgerechneten 50mW mit der der AP arbeiten darf entsprechen also circa 17dBm. Diese Werte passen zu den tatsächlichen Werten aus der Spezifikation in Tabelle 5 Abschnitt „Maximum Conducted Transmit Power“.

Wie bereits zu Beginn des Kapitels erwähnt ist der maximale Pfadverlust mit -80dBm festgelegt. Der dBm Wert kann mit der Formel 2.17 auf einen Leistungswert gebracht werden.

$$p = 10 * \log\left(\frac{P}{P_0}\right) \quad (2.17)$$

Setzt man die -80dBm in die Formel 2.17 ein ergibt sich die Formel 2.18. Diese muss nach x auflösen werden.

$$-80 = 10 * \log\left(\frac{x}{1 * 10^{-3}}\right) \rightarrow x = 1 * 10^{-10} \quad (2.18)$$

Setzt man das Ergebnis und die Ausgangswerte in die Formel 2.14 ein erhält man, nach auflösung nach d , den maximalen Abstand zwischen Sender und Empfänger

$$1 * 10^{-10} = 50 * 10^{-3} * 2^{10/10} * 1 * \left(\frac{3 * 10^8}{4 * \pi * d}\right)^2 \rightarrow d = 314.5m \quad (2.19)$$

Der rechnerische maximale Abstand beträgt 314.5 Meter bei freier Sichtverbindung (Freiraumausbreitung). [Neu05, vgl. s. 10 - 15]

2.4.2 Empirisches Feldstärkmodell

Da im Normalfall keine Freiraumausbreitung existiert muss ein Modell erstellt werden um die Feldstärke in Abhängigkeit der Entfernung zu beschreiben. Dieses Modell, genauer gesagt die Funktion die dieses Modell beschreibt, kann mithilfe von „Linearer Regression“¹⁵ gelöst werden. zum lösen einer linearen Regression kann das Programm

¹⁵<http://de.wikipedia.org/wiki/Regressionsanalyse>

Matlab ¹⁶ verwendet werden.

Zuerst müssen Messpunkte die die Feldstärke und den Abstand zum AP enthalten aufgenommen werden, siehe Tabelle 2.2 Hierbei handelt es sich um Werte die von ¹⁷ genommen wurden.

ABSTAND [M]	FELDSTRÄRKE [DBM]
0	-10
0.3	-20
1	-30
2	-40
4	-45
6	-50
10	-60

Tabelle 2.2: Regressions Messwerte

Bei der Regression wird versucht eine Funktion so in die Messwerte zu legen das der Abstand von Funktionswert zu Messwert bei allen Messpunkten ein Minimum annimmt. Dies geschieht durch Anpassung von Koeffizienten in der Ausgangsfunktion. Wobei es entscheidend ist die Ausgangsfunktion passend zu wählen. Hierbei ist etwas Geduld gefragt.

Für das Beispiel hat sich folgende Funktion als passend erwiesen.

$$f_{(x)} = a_0 + a_1 * x + a_2 * x^2 + a_3 * x^4 + a_4 * x * e^{-x} \quad (2.20)$$

Hilfe zu Erstellung dieser Funktion findet man in der Matlab v7.2 Hilfe unter den Punkten:

- MATLAB Functions for Polynomial Models
- Linear Model with Nonpolynomial Terms

Die Koeffizienten a_0 bis a_4 werden mit folgendem Matlab Programm berechnet.

```

1 %Werte Definieren
2 x = [0 .3 1 2 4 6 10]';
3 y = [-10 -20 -30 -40 -45 -50 -60]';
4
5 %Werte berechnen
6 T = [ones(size(x)) x x.^2 x.^3 x.*exp(-x)];
7 a = T\y
8 X = (0:0.1:10)';
9 Y = [ones(size(X)) X X.^2 X.^3 X.*exp(-X)]*a;
10
11 %Ergebniss plotten
12 plot(X,Y,'-o',x,y,'o'), grid on

```

Listing 2.1: Matlab Programm für lineare Regression

¹⁶<http://www.mathworks.de/>

¹⁷http://wiki.informatik.hu-berlin.de/nomads/index.php/MagicMap_Verfahren

In der Abbildung 2.8 ist schön zu sehen wie sich die Funktion an die Messpunkte anlegt. Es ist zu erkennen das die Abnahme zu Beginn stark ist und gegen Ende hin abnimmt. Nach dem Ausführen des Programms ist im „Command Windows“ das Array mit den Variablen a_0 bis a_4 zu sehen. Die konkreten Werte sind in der Tabelle 2.3 angeführt.

VARIABLE	WERT
a_0	-10,2323
a_1	-14,9109
a_2	2,0073
a_3	-0,1014
a_4	-21,6554

Tabelle 2.3: Koeffizienten des Regressionsbeispiels

Die Funktion die die Werte aus der Tabelle 2.2 beschreibt lautet also.

$$f(x) = -10.2323 + -14.9109 * x + 2.0073 * x^2 + -0.1014 * x^4 + -21.6554 * x * e^{-x} \quad (2.21)$$

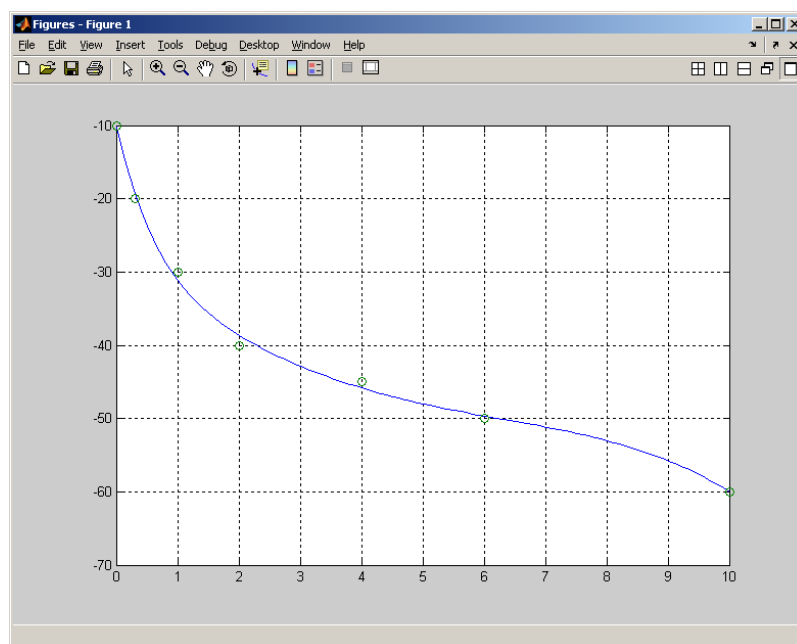


Abbildung 2.8: Regressionsmodell

2.5 Sicherheit

Aus der Sicht des Klienten ist ein Positionierungssystem die den Standort ohne dessen Zustimmung/Wissen ermitteln kann höchst bedenklich. In dieser Arbeit wird ein System vorgestellt, dass keine Personenbezogenen Daten speichert. Es wird lediglich die MAC- und IP Adresse des Klienten, für den Zeitraum der Lokalisierung, gespeichert. Dabei wird keine Verbindung zwischen dem Benutzer und der MAC Adresse hergestellt. Des weiteren werden die Daten der Positionsbestimmung in regelmäßigen Abständen gelöscht.

Kapitel 3

Neue Technologien im WLAN

Nach den Grundlagen bereits existierender Systeme möchte ich in diesem Kapitel eine kurze Übersicht der neuen Technologien 802.11k und IPv6 im WLAN geben.

3.1 802.11k

Der Standard 802.11k ist eine Weiterentwicklung des 802.11 Standards welche als Draft 2007 veröffentlicht wurde. Der Standard 802.11k definiert Informationen die über Funk- und Netzwerkaktivitäten eines WLANs zur Verfügung gestellt werden. Das Ziel des Standards ist die Datenübertragung zu optimieren. Mit den von der 802.11k Arbeitsgruppe erstellten Protokollen können WLAN Klienten wie z.B.: VoIP Mobiltelefone, direkt die notwendigen Informationen (Verfügbarkeit, Kapazität, Störungen, usw.) über benachbarte APs abfragen und bei Bedarf den zwischen den verfügbaren APs umschalten.

Durch diese Eigenschaft ist notwendig die entsprechende Signalstärke und Empfindlichkeit zu messen und zu verwalten. Dies ist im neuen Standard definiert. [Com07, vgl.]

Neue Messtechniken ermöglichen es einem AP das Funkumfeld in dem er arbeitet besser analysieren zu können. Diese Techniken bieten den APs die Möglichkeit das Funkumfeld zu beobachten und Daten zur Verbindungsqualität sowie allgemeine Informationen zum Funkumfeld aufzunehmen. Ein AP hat die Fähigkeit zu wählen ob er lokale Messungen durchführt oder eine Anfrage an einen anderen AP sendet beziehungsweise kann er auch aufgefordert werden Messungen durchzuführen und die Ergebnisse zurück zu liefern. Die Messdaten stehen dem AP für Managementaufgaben sowie Protokoll höhere Schichten, wo sie von einem breiten Bereich an Anwendungen genutzt werden können, zur Verfügung. Durch die Messungen wird es dem AP

ermöglicht sich schneller auf Änderungen im Funkumfeld, wie Störungen oder dem Ausfalls eines benachbarten APs, einstellen zu können. Der Dienst der Messtechnik beinhaltet Messverfahren für die Bereiche Leistungsfähigkeit, Zuverlässigkeit und Instandhaltbarkeit von WLANs und bietet herstellerübergreifende Messungen ebenso wie die Weitergabe der Ergebnisse an Protokoll in höheren Schichten des Kommunikationsstacks. [Soc07, p. 3]

Das Hinzufügen von Standard-Messverfahren und das Weiterleiten dieser Ergebnisse an höhere Layer bringt die Verpflichtung möglichst genauer Ergebnisse mit, da Anwendungen nur mit genauen Ergebnissen ihre Dienste in gewohnter Qualität anbieten können. Zu diesen Anwendungen gehören unter anderem VoIP, Video over IP, Standortbezogene Dienste sowie Dienste die sich in einem funktechnisch nicht optimalen Gebiet befinden (wie z.b.: Mehrfamilienhäuser, Flugzeuge, Fabriken usw.). Die WLAN Funkmesstechnik versucht auf die meisten Probleme eines frei zugänglichen Frequenzspektrums einzugehen.

Die Mobilität der Endgeräte macht für Anwendungen wie VoIP oder Videostreaming eine Reihe von Messverfahren wie Kanalbelegung und Nachbarschaftsanfragen notwendig um Informationen für die Vorbereitung von Handover-Aufgaben zur Verfügung zu stellen. Diese Informationen haben einen drastischen Anstieg der Geschwindigkeit beim Zellwechsel zur Folge und können von den APs benutzt werden um intelligente Wege zur Ausnutzung des verfügbaren Spektrums sowie der Bandbreite zu finden.

Die Anfrage/Antwort Aufgaben untergliedern sich wie folgt:

- Beacon
- Frame
- Channel Load
- Noise Histogram
- STA Statistics
- Location Configuration Information
- Neighbor Report
- Link Measurement

- Transmit Stream /Category Measurement

Die nur Anfrage Aufgaben sind:

- Measurement Pause

Die nur Antwort Aufgaben sind:

- Measurement Pilot

Jetzt möchte ich nur kurz die verschiedenen Aufgaben beschreiben.

3.1.1 Beacon

Das Beacon Anfrage/Antwort Paar ermöglicht es einem AP eine Liste mit APs, auf einem bestimmten oder allen Kanälen, von einem zweiten AP zu bekommen.

3.1.2 Frame

Das Frame Anfrage/Antwort Paar gibt eine Liste des Traffics auf einem bestimmten Kanal und eine Zählung der Empfangen Frames eines APs zurück.

3.1.3 Channel Load

Das Channel Load Anfrage/Antwort Paar gibt die Auslastung eines Kanals auf dem angefragten AP zurück.

3.1.4 Noise Histogram

Das Noise Histogram Anfrage/Antwort Paar gibt das Power Histogramm von nicht 802.11 Geräten zurück, vorausgesetzt CCA zeigt das keine WLAN Daten unterwegs sind.

3.1.5 STA Statistics

Das STA Statistics Anfrage/Antwort Paar gibt eine Gruppe von AP Countern und die durchschnittliche Zugriffsverzögerung einer BSS zurück. Zu diesen Countern gehören unter anderem:

- Anzahl übertragender Fragmente
- Anzahl übertragender Multicast Frames
- Anzahl fehlerhafter Frames

- Anzahl erfolgreicher RTS Übertragungen
- Anzahl fehlerhafter RTS Übertragungen
- Anzahl fehlerhafter FCS

3.1.6 Location Configuration Information

Das Location Anfrage/Antwort Paar gibt die Position des angefragten APs in der Form Breitengrad, Längengrad und Höhenlage. Die Höhenlage beinhaltet z.b.: das Stockwerk.

3.1.7 Neighbor Report

Das Neighbor Report Anfrage/Antwort Paar gibt Informationen über Nachbarn zurück, welche Kandidaten für eine BSS Übergabe sind.

3.1.8 Link Measurement

Das Link Measurement Anfrage/Antwort Paar gibt Informationen über RF Eigenschaften einer Verbindung zwischen einem AP und einem Klient zurück. Diese Messung zeigt die momentane Verbindungsqualität an.

3.1.9 Transmit Stream /Category Measurement

Das Transmit Stream /Category Measurement Anfrage/Antwort Paar gibt Informationen über QoS zurück.

3.1.10 Measurement Pause

Die Measurement Pause Anfrage gibt die Pause zwischen einzelnen Messungen an.

3.1.11 Measurement Pilot

Die Measurement Pilot Antwort gibt einen Teil der Informationen des Beacon Frames an, wobei diese Messung nicht periodisch und in kürzeren Intervallen als das Beacon Intervall wiederholt wird.

[Soc07, p. 3-6]

3.2 WLAN und IPv6

In diesem Abschnitt werden nicht die Vorteile von IPv6 gegenüber IPv4 aufgezeigt, sondern nur die Auswirkungen von IPv6 auf WLAN. Da es zu diesem Thema zur Zeit

nur wenig Referenzen gibt wurde versucht selbst ein paar Auswirkungen herauszufinden. Als erstes ist es wichtig zu wissen wo IPv6 und WLAN im Protokollstack angesiedelt sind.

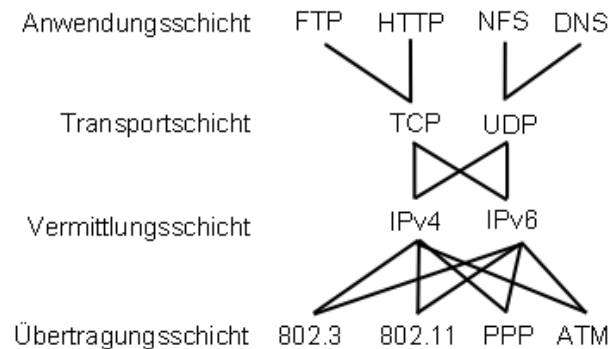


Abbildung 3.1: TCP/IP-Protokollstapel

Abbildung 3.1 zeigt den TCP/IP Protokollstack bei dem in der Vermittlungsschicht IPv6 neben IPv4 zu sehen ist. Wohin WLAN (802.11) in der Übertragungsschicht angesiedelt ist. Das IPv6 und WLAN auf verschiedenen Schichten angesiedelt sind sagt aus das es für WLAN keinen Unterschied macht ob IPv6 oder IPv4 das Protokoll in der Vermittlungsschicht verwendet wird. Für WLAN hat IPv6 also nur die Auswirkung des längern Headers, dieser wächst von 20 Byte (IPv4) auf 40 Byte (IPv6).

3.2.1 Bandbreitenversuch

Um herauszufinden ob IPv6 Geschwindigkeitsvorteile auf WLAN hat wurde folgender Versuchsaufbau erstellt.

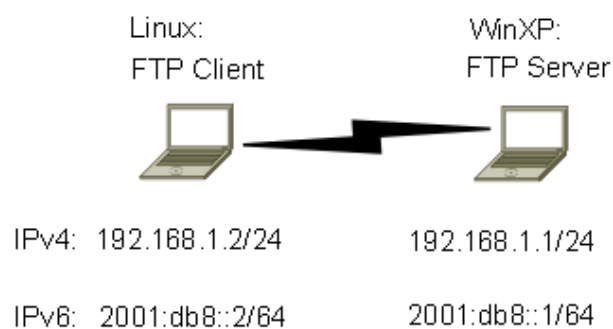


Abbildung 3.2: IPv6 Versuch

Hierbei wurden zwei Laptops per WLAN in einem Ad-Hoc Netzwerk verbunden und ihnen jeweils eine IPv4 beziehungsweise eine IPv6 Adresse gegeben. Unter Windows XP SP2 gibt es keine grafische Möglichkeit einer Netzwerkkarte eine IPv6 Adresse zuzuweisen, es funktioniert nur auf der Kommandozeile mit folgenden Befehlen.

```

1 netsh interface ipv6 set address "Drahtlose Netzwerkverbindung" 2001:db8::1
2 netsh interface ipv6 add route 2001:db8::/64 "Drahtlose Netzwerkverbindung" publish=yes

```

Listing 3.1: IPv6 Konfiguration unter Windows XP

Bevor man den Befehl eingibt muss man unter Systemsteuerung → Netzwerkverbindung → Eigenschaften der Drahtlosen Netzwerkverbindung → Installieren → Protokoll → Hinzufügen das Protokoll IPv6 hinzufügen. Unter Linux reicht es auf der Kommandozeile folgenden Befehl abzusetzen:

```
1 ifconfig ath0 add 2001:db8::2/64
```

Listing 3.2: IPv6 Konfiguration unter Linux

Danach wurde eine 94MB große Datei vom FTP Server der unter Windows XP lief auf den Linux Rechner kopiert. Als FTP Server kann die Freeware Version von Xlight [xli07] zum Einsatz, als FTP Client gftp [gft05]. Dabei ergaben sich folgende Übertragungsraten:

VERSION	MESSUNG 1	MESSUNG 2	MESSUNG 3
IPv6	1717,73KB/s	1719,46KB/s	1781,59KB/s
IPv4	1779,52KB/s	1751,56KB/s	1680,78KB/s

Tabelle 3.1: IPv6 vs IPv4

Diese Tabelle zeigt das sich kein Geschwindigkeitsvorteil bzw. Nachteil aus der Verwendung von IPv6 ergibt. Eine kurze Einführung in das Thema IPv6 im lokalen Netzwerk (Ethernet) findet sich unter [Kap07]

Kapitel 4

Theoretische Grundlagen von LBS

4.1 Ausgangslage

In diesem Kapitel werden die einzelnen Systeme theoretisch analysiert und verglichen. Die Recherche ergab, dass es sechs verschiedene Varianten sind. Diese werden im folgenden näher betrachtet.

- Differenzmethode
- Trilateration mit Mittelwerten
- Trilateration mit Normalverteilung
- Durchschnittsposition mit Normalverteilung
- Trilateration mit Abstandsberechnung
- Schnittpunkt zweier Kreise mit Mittelwerten

Alle dieser Verfahren arbeiten in 2 Phasen, einer Offline- und Onlinephase. In der Offlinephase wird das Gebiet in dem die Positionsbestimmung funktionieren soll mit Messpunkten versehen. An diesen Messpunkten werden Feldstärkemessungen durchgeführt. Dabei werden bei jedem Messpunkt mehrere Messwerte (>10) pro Access Point aufgenommen und in einer Datenbank gespeichert. Das Programm, dass diese Aufgabe übernimmt ist unter Punkt [5.4](#) zu finden.

In der Onlinephase werden die Daten aus der Offlinephase mit den Daten des Snifferprogramms verwendet um die Position heraus zu finden. Wie das funktioniert wird in den Kapitel [4.1.1](#) bis [4.1.6](#) beschrieben.

4.1.1 Differenzmethode

Bei der Differenzmethode, so wie bei allen anderen auch, muss zuerst eine Radiomap erstellt werden. Wie diese Radiomap aussehen kann ist in der Tabelle 4.1 dargestellt. Dabei ist es sinnvoll mehrerer Werte pro Messpunkt (M1 - M5) aufzunehmen und aus diesen einen Mittelwert zu bilden da sich somit Messfehler nicht so stark auswirken. [Sar03]

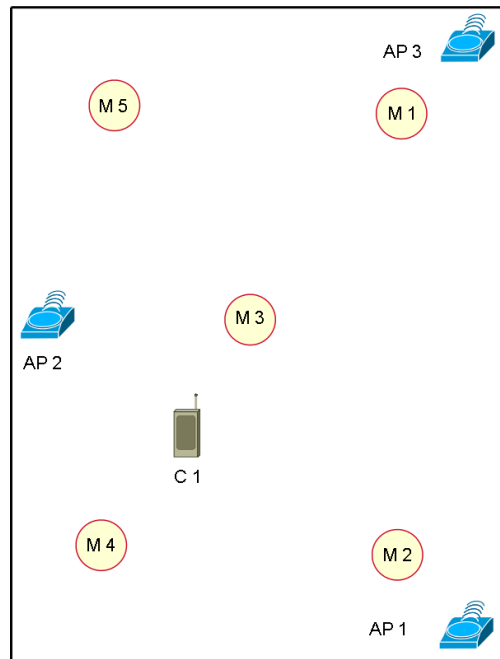


Abbildung 4.1: Differenzmethode Offlinephase

	AP1	AP2	AP3
M1	10	20	8
M2	20	15	10
M3	4	10	13
M4	6	9	20
M5	8	10	9

Tabelle 4.1: Differenzmethode Offlinephase

In der Onlinephase werden dann die von den APs aufgenommenen Daten verwendet um die Position des Klienten (C1) zu ermitteln.

Für die Positionsbestimmung wird als erstes eine Differenztafel erstellt wie in Abbildung 4.3 zu sehen. Die Werte in dieser Tabelle ergeben sich durch die Subtraktion der Klientwerte aus Tabelle 4.2 mit den Werten der Offlinephase (Tabelle 4.1). Es wird

also der Wert $C1xAP1$ ¹ vom Werte $M1xAP1$ abgezogen, danach der Wert $C1xAP2$ vom Wert $M1xAP2$ sowie der Wert $C1xAP3$ von $M1xAP3$. Dies wird in den Formeln 4.1 , 4.2 und 4.3 nochmals gezeigt. Nach diesem Prinzip werden auch die Zeilen M2, M3, M4 und M5 errechnet.

$$Diff.AP1 = C1xAP1 - M1xAP1 \quad (4.1)$$

$$Diff.AP1 = C1xAP2 - M1xAP2 \quad (4.2)$$

$$Diff.AP1 = C1xAP3 - M1xAP3 \quad (4.3)$$

	AP1	AP2	AP3
C1	13	15	9

Tabelle 4.2: Differenzmethode Klient

	AP1	AP2	AP3	DIFF. AP1	DIFF. AP2	DIFF. AP3	Summe
M1	10	20	8	3	-5	1	-1
M2	20	15	10	-7	0	-1	-8
M3	4	10	13	9	5	4	18
M4	6	9	20	7	6	11	24
M5	8	10	9	5	5	0	10

Tabelle 4.3: Differenzmethode Onlinephase

Nach der Erstellung der Differenztafel werden die einzelnen Zeilen dieser Tabelle aufsummiert. Es werden also die Werte der Spalten Diff. AP1 bis Diff. AP3 aus jeder Zeile aufsummiert und in der Spalte Summe gespeichert. Die Position des Klienten wird dort angenommen wo die Summe den Wert hat der am nächsten bei Null liegt.

4.1.2 Trilateration

Bei der Trilateration handelt es sich um eine Verfahren zur Berechnung des Schnittpunktes dreier Kreise im zweidimensionalen Raum. Das Verfahren kann auch im dreidimensionalen Raum verwendet werden, für das Thema LBS ist dies jedoch nicht erforderlich. Bei der Positionsbestimmung im zweidimensionalen Raum mittels Trilateration gibt es jedoch drei Grundvoraussetzungen [Wik08, vgl.]:

1. Alle drei Punkte müssen in einer Ebene liegen ($z=0$)

¹Das x zeigt an das es sich um den Schnittpunkt der Zeile C1 mit der Spalte AP1 handelt.

2. Ein Punkt muss im Ursprung 0/0 (x/y) liegen
3. Ein weiterer Punkt, neben dem im Ursprung, muss auf der x-Achse liegen

Die Ausgangslage für die Positionsberechnung bilden folgende 3 Formeln.

$$r_1^2 = x^2 + y^2 + z^2 \quad (4.4)$$

$$r_2^2 = (x - d)^2 + y^2 + z^2 \quad (4.5)$$

$$r_3^2 = (x - i)^2 + (y - j)^2 + z^2 \quad (4.6)$$

Zieht man die Formel 4.5 von der Formel 4.4 ab und löst nach x auf erhält man folgendes.

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d} \quad (4.7)$$

Setzt man die erhaltene Formel 4.7 in die Formel 4.4 ein erhält man den Schnittpunkt der ersten beiden Kreise.

$$y^2 + z^2 = r_1^2 - \frac{(r_1^2 - r_2^2 + d^2)^2}{4d^2} \quad (4.8)$$

Setzt man diese Formel 4.8 gleich mit der Formel 4.6 erhält man den y-Wert des gesuchten Schnittpunktes.

$$y = \frac{r_1^2 - r_3^2 - x^2 + (x - j)^2 + j^2}{2j} = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x \quad (4.9)$$

Somit sind die Formeln 4.7 und 4.9 die Lösung für das Problem. Die Werte d,i und j sind zuvor gemessene Werte. Die Abstände r1, r2 und r3 können durch eine abgewandelte Form der Differenzmethode oder Normalverteilungsmethode (siehe Punkt 4.1.3) bestimmt werden.

Mittels der Abbildung 4.2 wird die Funktionsweise der Trilateration nochmal beschrieben. Auf das Beispiel LBS umgelegt befinden sich an den Punkten P1, P2 und P3 die APs zur Feldstärkemessung. Die Abstände d,i und j müssen, wie oben schon erwähnt, vor der Positionsbestimmung abgemessen werden. Die Werte r1, r2 sowie r3 werden aus der Feldstärke berechnet und in die Formeln 4.7 und 4.9 eingesetzt um den Punkt B, also den Aufenthaltsort des Klienten, zu bestimmen.

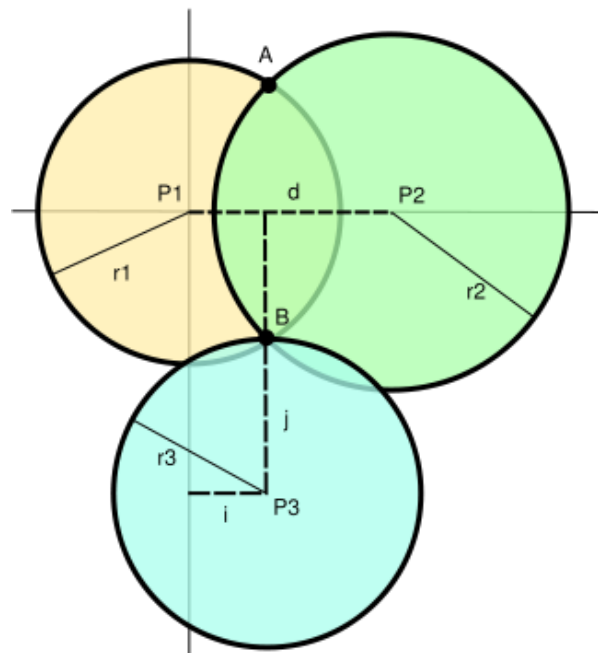


Abbildung 4.2: Trilateration [Wik08]

4.1.3 Normalverteilung und Trilateration

Bei der Normalverteilung mit Trilateration werden zwei Verfahren kombiniert um die Position des Klienten zu bestimmen. Als erstes muss wie bei der Differenzmethode, in der Offlinephase, eine Radiomap erstellt werden. Hierbei unterscheidet sich der Aufbau jedoch in einem wesentlichen Punkt. Bei der Differenzmethode wird pro Messpunkt der Durchschnittswert (Erwartungswert μ) gespeichert, bei der Normalverteilungsmethode wird zusätzlich zum Erwartungswert auch noch die Standardabweichung σ berechnet und gespeichert.

Die Tabelle die den Erwartungswert und die Standardabweichung aufnimmt hat folgendes Layout.

	ID	AP1E	AP2E	AP3E	AP1S	AP2S	AP3S
Datentypen	id (Primary Key)	float4	float4	float4	float4	float4	float4

Tabelle 4.4: Normalverteilung Tabellenlayout

Die Werte ap1e, ap2e und ap3e speichern den Erwartungswert pro Referenzpunkt, die Werte ap1s, ap2s und ap3s speichern die Standardabweichung an jedem Referenzpunkt.

In der Onlinephase wird die Wahrscheinlichkeit berechnet das sich ein Klient bei einem bestimmten Messpunkt aufhält. Dabei wird folgendermaßen vorgegangen. Zu Be-

ginn wird aus der Radiomap für jeden Messpunkt der Erwartungswert μ , die Standardabweichung σ sowie die aktuelle Feldstärke x des Klienten gelesen und in folgende Formel eingesetzt:

$$u = \frac{(x + 0.5) - \mu}{\sigma} \quad (4.10)$$

Mit dem Wert u findet man in der Tabelle für die Normalverteilung, die unter [Pap01, vgl. s.734] zu finden ist, die Wahrscheinlichkeit das sich ein Klient bei einem Messpunkt aufhält. Als Ergebnis wird der Messpunkt mit der höchsten Wahrscheinlichkeit ausgegeben.

4.1.4 Durchschnittsposition

Bei diesem Verfahren wird zuerst, aus der Sicht jedes APs, der Referenzpunkt gesucht der die höchste Aufenthaltswahrscheinlichkeit besitzt. Die Suche des Referenzpunktes wurde bereits in Abschnitt 4.1.3 beschrieben. Das Ergebnis dieser Suche könnte zum Beispiel aussehen wie in folgender Auflistung sowie in Abbildung 4.3 zu sehen.

- AP1 → Klient steht bei Referenzpunkt 2
- AP2 → Klient steht bei Referenzpunkt 6
- AP3 → Klient steht bei Referenzpunkt 1

Des weiteren sind für jeden AP die Abstände zu den Referenzpunkten bekannt. Dieses Wissen wird sich hier zu nutze gemacht. Es werden jedoch nur die Abstände zu einem AP benötigt, diese könnten wie folgt aussehen.

REFERENZPUNKT	X ABSTAND ZU AP1	Y ABSTAND ZU AP1
1	-100cm	1530cm
2	-100cm	960cm
6	312,5cm	1530cm

Tabelle 4.5: AP1 Abstände zu Referenzpunkten

Die Hauptaufgabe des Algorithmus ist es nun die X und Y Abstände auf zu summieren und durch drei zu teilen. Für das Beispiel sieht das wie folgt aus.

$$x = \frac{\text{punkt1x} + \text{punkt2x} + \text{punkt6x}}{3} = \frac{-100\text{cm} + (-100\text{cm}) + 312,5\text{cm}}{3} = 37,5\text{cm} \quad (4.11)$$

$$y = \frac{\text{punkt1y} + \text{punkt2y} + \text{punkt6y}}{3} = \frac{1530\text{cm} + 960\text{cm} + 1530\text{cm}}{3} = 1340\text{cm} \quad (4.12)$$

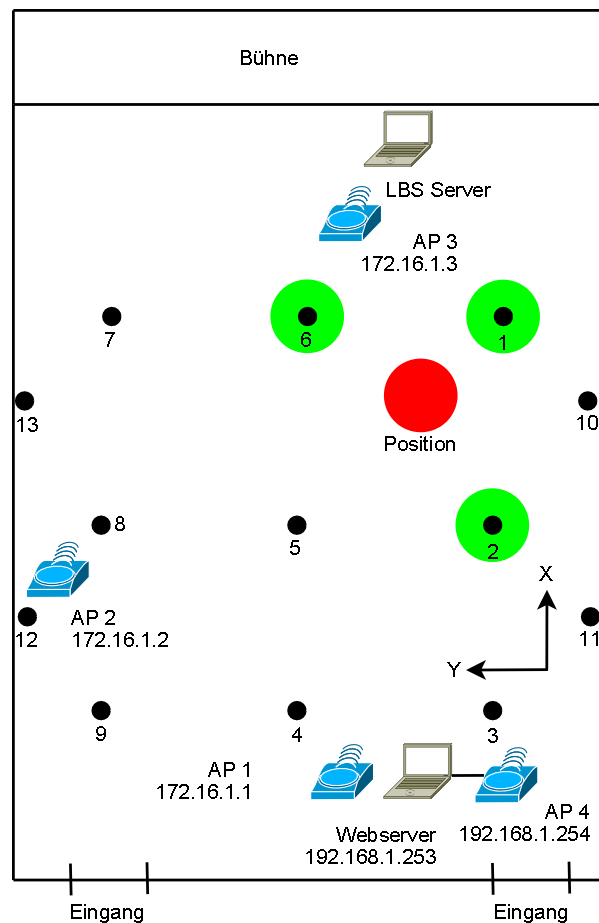


Abbildung 4.3: Durchschnittsposition

Als Ergebnis liefert der Algorithmus die gemittelte Position des Klienten. In diesem Beispiel würde der Klient 37,5cm in X Richtung und 1340cm in Y Richtung vom AP1 entfernt stehen.

4.1.5 Abstandsberechnung

Bei diesem Verfahren wird der Abstand des Klienten zu einem AP mithilfe eines empirischen Modells berechnet. Eine Einführung bietet Abschnitt 2.4.2. Für die Abstandsberechnung wurde im Festsaal alle zwei Meter die Feldstärke eines Klienten aufgenommen und in der Tabelle 4.6 festgehalten.

Mit diesen Feldstärken und Abständen sowie der linearen Regression wird jetzt versucht eine Funktion zu finden die am besten in die Messwerte passt. Dies geschieht in Matlab mit folgendem Programm.

```
1 %Werte Definieren
2 %WERTE AUS DEM FESTSAAL FÜR ACER N50
3 x = [0.1 2 4 6 8 10 12 14 16 18]';
4 y = [71.6 42.875 41.5 40.8 39.916 36.8 36.6 32.9 31.3 31.4]';
5 %Werte berechnen
```

ABSTAND [M]	FELDESTÄRKE
0,1	71.6
2	42.875
4	41.5
6	40.8
8	39.916
10	36.8
12	36.6
14	32.9
16	31.3
18	31.4

Tabelle 4.6: Feldstärke in verschiedenen Entfernungen

```

6 T = [ones(size(x)) exp(-x) x.^2 x.^3 x.*exp(-x) ]
7 a = T\y
8 X = (0:0.1:18)'
9 Y = [ones(size(X)) exp(-X) X.^2 X.^3 X.*exp(-X)]*a
10 %Ergebniss plotten
11 plot(X,Y,'-','x,y','o'), grid on

```

Listing 4.1: Matlab Programm zur Abstandsberechnung

Das Programm in Listing 4.1 hat die Formel 4.13 als Grundlage.

$$f_{(x)} = a_0 + a_1 * e^{-x} + a_2 * x^2 + a_3 * x^3 + a_4 * x * e^{-x} \quad (4.13)$$

Mit Hilfe der Regression hat das Programm die Parameter a_0 bis a_4 wie folgt bestimmt.

VARIABLE	WERT
a_0	44,588
a_1	32,4025
a_2	-0,0998
a_3	0,0033
a_4	-19,5541

Tabelle 4.7: Koeffizienten für das Abstandsverfahren

Setzt man die Parameter aus Tabelle 4.7 in die Formel 4.13 und lässt sich das Ergebnis ausgeben sieht man, dass die Funktion sehr genau innerhalb der Messpunkte liegt, siehe Abbildung 4.4.

4.1.6 Schnittpunkt zweier Kreise

Der Schnittpunkt zweier Kreise ist eine Art Trilateration. Damit dem Klient einer der beiden Schnittpunkte als Position zugeordnet werden kann werden die Daten eines dritten AP benötigt.

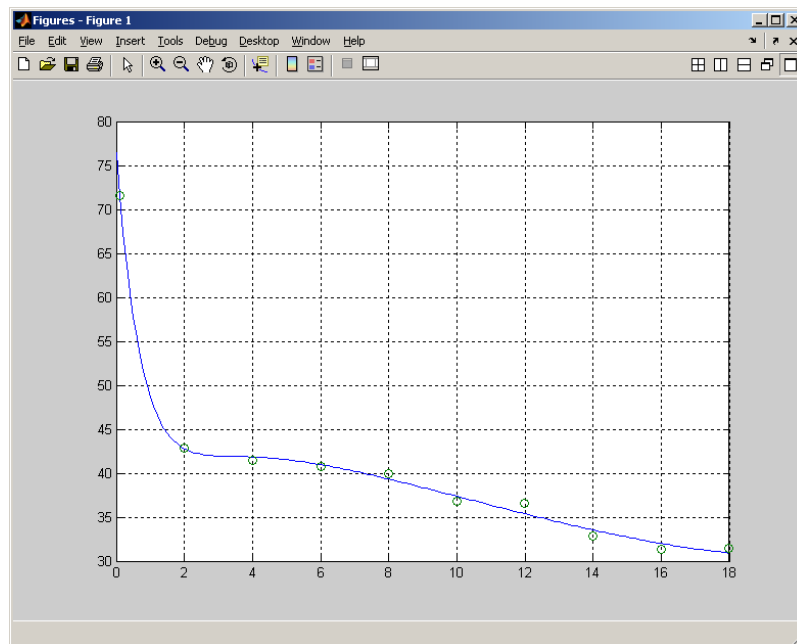


Abbildung 4.4: Abstandsmodell

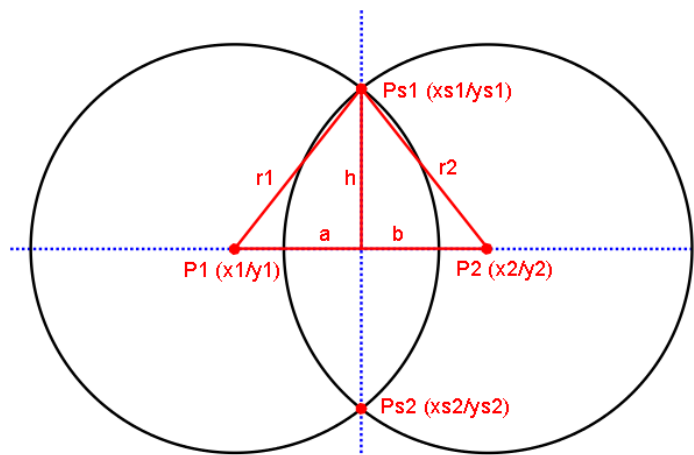


Abbildung 4.5: Schnittpunkt zweier Kreise [Kow08]

Voraussetzung für die Berechnung sind die Radien r_1 und r_2 sowie die x und y Werte der Kreismittelpunkte P_1 und P_2 . Der Abstand der Punkte P_1 und P_2 wird mit d bezeichnet und berechnet sich wie folgt. [Kow08, vgl.]

$$d = a + b = \sqrt{\Delta x^2 + \Delta y^2} \quad (4.14)$$

Der Abstand a ergibt sich wie folgt.

$$a = \frac{r_1^2 - r_2^2 + d^2}{2 * d} \quad (4.15)$$

Des weitern ist noch der Werte h interessant.

$$h = \sqrt{r_1^2 - a^2} \quad (4.16)$$

Mit diesen Formeln lassen sich die beiden Schnittpunkte P_{s1} und P_{s2} berechnen. Der erste Schnittpunkt wird durch die Formeln 4.17 und 4.18 ausgedrückt. Den zweiten Schnittpunkt errechnet man mit den Formeln 4.19 und 4.20. [Kow08, vgl.]

$$xs_1 = x_1 + \left(\frac{a}{d} * \Delta x - \left(\frac{h}{d}\right) * \Delta y\right) \quad (4.17)$$

$$ys_1 = y_1 + \left(\frac{a}{d} * \Delta y + \left(\frac{h}{d}\right) * \Delta x\right) \quad (4.18)$$

$$xs_2 = x_1 + \left(\frac{a}{d} * \Delta x + \left(\frac{h}{d}\right) * \Delta y\right) \quad (4.19)$$

$$ys_2 = y_1 + \left(\frac{a}{d} * \Delta y - \left(\frac{h}{d}\right) * \Delta x\right) \quad (4.20)$$

Nach der Bestimmung der beiden Schnittpunkte muss noch die Wahl getroffen werden bei welchem Schnittpunkt sich der Klient aufhält. Dies geschieht indem zuerst die Abstände der beiden Schnittpunkte vom AP2 berechnet werden und von diesen Abständen der gemessene Abstand des Klienten von AP2 abgezogen wird. Als Ergebnis wird der Schnittpunkt, der die geringere Differenz aufweist, ausgegeben.

4.2 Exkurs: Der RSSI Wert

Um den RSSI Wert besser verstehen zu können wird hier ein kurzer Einblick in die DSSS Technologie, die bei WLAN auf dem OSI Layer Eins eingesetzt wird, gegeben. Bei WLAN ist der Layer Eins in 2 Teilschichten unterteilt, wie in der Abbildung 4.6 zu sehen ist. Es handelt sich dabei um den PLCP und den PMD Sublayer.

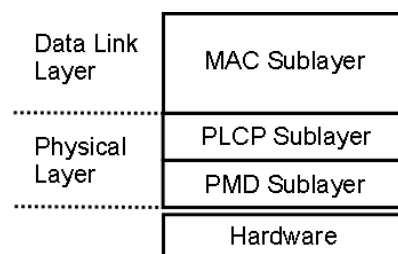


Abbildung 4.6: WLAN Layer Eins und Zwei [Soc03, vgl. s 28]

Die Aufgabe des PLCP ist es eine einheitlich Schnittstelle zwischen MAC Layer und den medienabhängigen PMD bereit zu stellen. Der PLCP nimmt die Daten (MPDUs) des MAC Layer entgegen und wandelt sie für den PMD Layer um. Der PMD wiederum bringt die Daten in ein Format, dass über das drahtlose Medium übertragen werden kann.

Der PLCP und der PMD kommunizieren über sogenannte Serviceprimitiven. Ein Serviceprimitiv setzt sich aus einem Dienstenamen sowie einer Information zusammen. Der WLAN Standard unterscheidet dabei zwei Kategorien. Die erste Kategorie stellt eine Peer-to-Peer Kommunikation zwischen MACs bereit, die zweite eine Kommunikation zwischen den Sublayern PLCP und PDM. In den beiden Kategorien gibt es jeweils drei Formen von Serviceprimitiven [[Rec06](#), vgl. s. 69]:

- Request
- Indicate
- Confirm

Um eine Vorstellung von den Serviceprimitiven zu bekommen werden hier einige aufgezählt:

- PHY-DATA.Request: Diese Serviceprimitive wird vom MAC generiert um Daten an den PHY zu übertragen.
- PHY-DATA.Indication: Diese Serviceprimitive zeigt eine Datenübertragung vom PHY um MAC Layer an.
- PHY-CCA.Indication: Über diesen Serviceprimitiv teilt der PHY dem MAC den Status des Übertragungsmedium an.
- PHY-CCARESET.Confirm: Damit bestätigt der PHY dem MAC das ein Reset an der CCA-State Maschine durchgeführt wurde.

Nach der kurzen Einführung in den PHY Layer folgt eine Einführung in die Signalspreizung bei WLAN. Bei der Signalspreizung wird ein schmalbandiges Nutzsignal in ein Breitbandiges übergeführt. Für diese Spreizung wird eine Codefolge verwendet die im Vergleich zu anderen Verfahren nicht vom Nutzsignal abhängig ist. Der Vorteil dieser Methode ist eine wesentlich höhere Störuneempfindlichkeit sowie ein gewisser Abhörschutz. Für den Standard 802.11 sind zwei Spreizband-Technologien vorgesehen [[Rec06](#), vgl. s. 73].

- FHSS Technologie
- DSSS Technologie

4.2.1 FHSS Technologie

Bei FHSS handelt es sich um eine Technologie die mittels einer Frequenzumtastung zwischen vielen Frequenzen hin und her schaltet. Jede dieser Frequenzen bildet dabei einen Kanal. Manche Systeme arbeiten mit mehreren tausend Kanälen. Der Wechsel zwischen den einzelnen Kanälen ist in einer sogenannten Codefolge festgelegt. Die zur Verfügung stehende Bandbreite von 2,4 - 2,4835 GHz wird laut IEEE 802.11 und ETSI in 79 Frequenzunterbänder mit jeweils 1MHz aufgeteilt. Die Datenrate ist bei der FHSS Technologie auf 2MBit/s beschränkt. [Rec06, vgl. s. 74]

4.2.2 DSSS Technologie

Die DSSS Technologie setzt das bekannte und sehr weit verbreitete Signalspreizverfahren, oftmals auch als Pseudo-Noise-Verfahren bezeichnet, um. Dabei wird mittels einer digitalen Codefolge der Träger des Nutzsignales moduliert, dabei ist die Bitrate des Codes sehr viel größer als die Bandbreite des Nutzsignals. Bei DSSS wird laut IEEE 802.11, IEEE 802.11b sowie ETSI die Bandbreite (2,4 - 2,4835 GHz) in 13 Frequenzunterbänder mit einer Bandbreite von jeweils 22 MHz und 30 MHz Center Frequenzabstand unterteilt. Es findet im Gegensatz zur FHSS Technologie kein Wechsel zwischen den 13 Kanälen statt.

Bei der Signalspreizung handelt es sich um ein System, dass ein schmalbandiges Nutzsignal von hoher Leistung in ein breitbandiges Signal mit niedriger Leistung überführt, siehe Abbildung 4.7. [Rec06, vgl. s. 87]

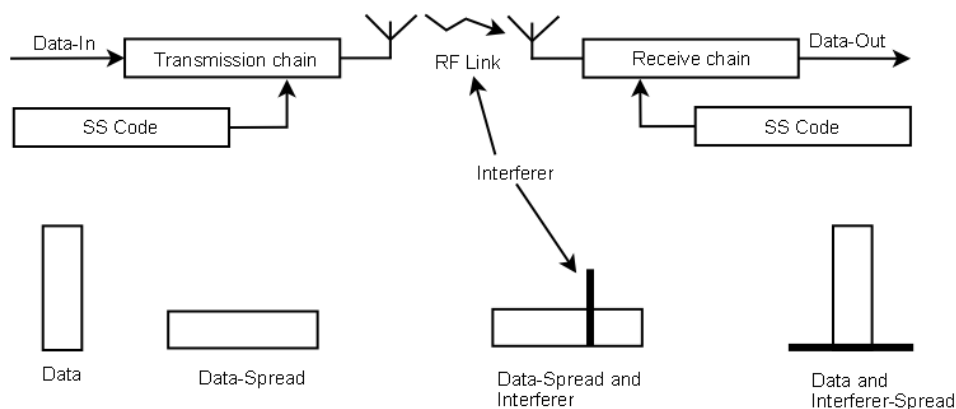


Abbildung 4.7: WLAN Funkstrecke [Neu06, vgl. s. 12]

In der Abbildung 4.7 ist zu sehen dass ein schmalbandiges Störsignal (Interferer) beim Signalspreizen auf der Empfängerseite in ein breitbandiges Stören mit geringer Leistung gespreizt wird. Dieses breitbandige aber schwache Stören hat nur einen sehr geringen Einfluss auf das Nutzsignal.

Die Funkschnittstelle überträgt das Nutzsignal wie in Abbildung 4.8 dargestellt.

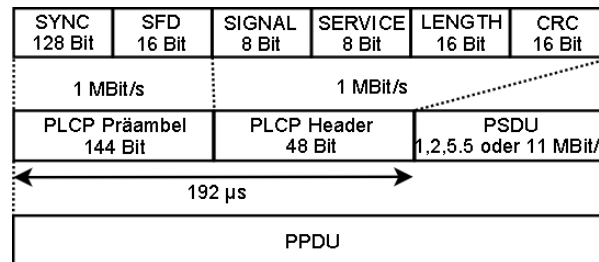


Abbildung 4.8: DSSS Frameformat [Rec06, vgl. s. 99]

Das SYNC Feld dient der Synchronisation, das SFD Feld zeigt das Ende der Präambel an. Der PLCP Header beinhaltet folgende Felder, Signal, Service, Length und CRC. Das Signal Feld zeigt die Datenrate der MPDU an. Das Service Feld gibt Auskunft darüber mit welcher Geschwindigkeit der Datenteil übertragen wird. Im Length Feld wird die Zeit, in μs , festgelegt die für die Übertragung der PSDU benötigt wird. Das letzte Feld des PCLP Headers ist eine Checksumme.

Im PSDU Feld werden die Daten des MAC Layers (MPDU) übertragen. Es hat eine Länge von 0 - 4095 Byte. [Rec06, vgl. s. 99]

Wie auch beim FHSS-PMD stehen dem DSSS-PMD für Steuerung und Datenaustausch verschiedene Serviceprimitiven zur Verfügung. Der für LBS wichtigste ist dabei der PMD_RSSI.Indicate. Dieser zeigt die die Feldstärke des empfangenen Paketes an. Der Wert für RSSI wird anhand des PLCP Headers bestimmt.

PMD_RSSI.Indicate(RSSI): Hierbei handelt es sich um eine optionale Serviceprimitive, mit der der PMD den PLCP und dem MAC die Signalstärke (RSSI = Received Signal Strength Indication) anzeigen kann. Die Anzeige des RSSI-Werts erfolgt über den RSSI-Parameter, der eine Länge von 8 Bits hat und somit 256 Werte annehmen kann. Der RSSI-Wert wird durch den PHY mit dem Beginn des SFD bis zum Ende des PLCP-Headers bestimmt. Dieser Wert kann auch in Verbindung mit dem Ergebnis der PMD_SQ.Indicate Serviceprimitive für die CS/CCA-Prozedur verwendet werden. [Rec06, p. 105]

4.2.3 RSSI und Verschlüsselung

Allgemein ist es bei jedem Paket möglich beziehungsweise sogar verpflichtend die Feldstärke zu lesen da der PMD anhand dieser entscheiden ob er dem PCLP Signaliert das ein Paket eingetroffen ist oder nicht. Ist es jedoch erwünscht neben dem RSSI Wert auch die IP Adresse zu sniffen ergeben sich beim Einsatz einer Verschlüsse-

lungsmethode Probleme. Dies ist darauf zurück zu führen das bei der Verschlüsselung der gesamte Framebody (siehe Abbildung 4.9), der die Daten der oberen Layer enthält, verschlüsselt wird. Die MAC Adresse des Klienten hingegen steckt nicht im Frame Body sonder MAC Header.

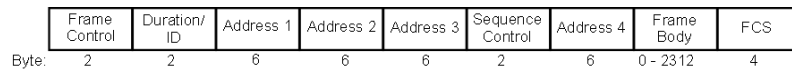


Abbildung 4.9: MAC Frameformat [Soc03, vgl. s. 34]

4.3 Kalibrieren

Mit Kalibrieren ist eine Anpassung auf verschiedene Hardwarekomponenten gemeint. Die Klientgeräte wie PDA, Laptop oder Mobiltelefon unterscheiden sich in ihre Antennencharakteristik und Sendeleistung, was in gleichen Abständen vom AP zu Unterschiedlichen Feldstärkewerten führt.

Im ersten Schritt muss dem System bekanntgemacht werden mit welchem Gerätetyp der Klient arbeitet. Dies könnte auf der Startseite des „Wlan für St.Pölten“ mittels einer Liste aus der der Klient seinen Gerätetyp auswählt, erledigt werden. Folgende drei Gerätetypen wurden bei der Teststellung verwendet.

- PDA, Acer N50
- PSP
- Laptop, HP nc6120

4.3.1 Radiomap

Die Methoden die mit einer Radiomap arbeiten benötigen pro Gerätetyp eine Radiomap. Es muss also für jeden Gerätetyp eine eigene Radiomap mit Durchschnittswerten sowie eine mit dem Erwartungswert und der Standardabweichung angelegt werden. Je nach Anzahl der Messpunkte und Gerätetypen kann dieser Vorgang mehrere Stunden in Anspruch nehmen. Die Erstellung der Radiomap für den PDA (Acer N50) hat pro Messpunkt etwa 5 Minuten gedauert, dies macht bei 13 Messpunkten 65 Minuten. Wiederholt man den Vorgang mit 2 weiteten Geräten kommt man auf 195 Minuten was etwas 3,2 Stunden entspricht.

4.3.2 Kalibrierungspunkt

Eine weitete Möglichkeit für die Kalibrierung ist ein Kalibrierungspunkt. Hierzu könnte es zwei Varianten geben.

Variante 1 Der Klient bekommt über einen Webserver eine Karte des Rathausplatzes präsentiert und muss auf seinen Standort klicken. Danach wird von diesem Punkt eine Messung der Feldstärke durchgeführt und je nach Ergebnis eine Radiomap ausgewählt oder neu erstellt. Bei der Erstellung einer neuen, nur für den besagten Klient gültigen, Radiomap könnte die Werte einer Default Radio Map einfach um den Wert x erhöht oder verringert werden.

Variante 2 Der Klient muss sich auf einen bestimmten Punkt des Rathausplatzes stellen, welcher in der Offlinephase definiert wurde und dort eine Referenzmessung durchführen lassen. Da beim Anklicken der Position auf der Karte, in Variante 1, leicht Ungenauigkeiten auftreten können, sollte bei der Variante 2 eine genauere Kalibrierung entstehen.

4.3.3 Selbstkalibrierung der Umwelteinflüsse

Das selbstkalibrierende System erledigt seine Aufgabe ohne zutun des Klienten. Bei diesem System misst der AP2 in der Offlinephase die Feldstärke eines Paktes das vom AP1 ausgesendet wird. Wie in Abbildung 4.10 links zu sehen könnte dieses Paket eine Feldstärke von -40dBi aufweisen. In der Onlinephase werden dann ebenfalls die Pakete vom AP1 beim AP2 gemessen, durch Störungen wie das Wetter, Menschen oder Hochspannungen durch Mikrowellen, wird sich die Feldstärke, wie in der Abbildung 4.10 rechts zu sehen ist, verschlechtern. Die Unterschiedlichen Feldstärken der Pakete von AP1 in Richtung AP2 weisen also direkt auf Umwelteinflüsse hin. Je nachdem wie stark die Feldstärke abnimmt müssen die Radiomaps angepasst werden.

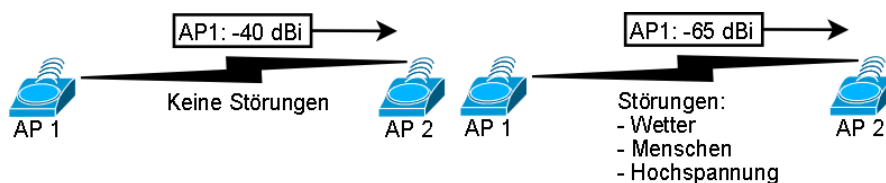


Abbildung 4.10: Selbstkalibrierung

4.3.4 Abstandsberechnung

Bei der Methode die die Abstandsberechnung einsetzt, siehe Abschnitt 6.8, müssen die Koeffizienten angepasst werden. Die Koeffizienten für die PSP und den Laptop HP nc6120 wurden mit dem Programm aus Listing 4.1 berechnet. Hierbei wurde nur das y Array mit den Feldstärke laut den Messungen in Tabelle 4.9 angepasst. Vergleicht man die Feldstärkewerte an den einzelnen Messpunkte ist gut zu sehen, dass die PSP im Vergleich zum Laptop und PDA mit einer weitaus niedrigeren Sendeleistung arbeitet. Ohne einer Kalibrierung wäre es somit unmöglich eine exakte Position zu berechnen. Die Ergebnisse der Berechnungen sind in der Tabelle 4.8 zu sehen.

VARIABLE	ACER N50	PSP	HP NC 6120
a_0	44,588	34,7057	43,4900
a_1	32,4025	24,5274	32,1607
a_2	-0,0998	-0,1209	-0,1211
a_3	0,0033	0,0045	0,0046
a_4	-19,5541	-18,3988	32,8209

Tabelle 4.8: Gegenüberstellung der Koeffizienten

POSITION	ACER N50	PSP	HP NC 6120
0	71,6	55,23	75,55
1	42,875	32,63	56,4
2	41,5	32,13	44,33
3	40,8	30,5	40,88
4	39,916	30,16	39,125
5	36,8	27,15	34,92
6	36,6	24,588	34,125
7	32,9	23,058	32,4
8	31,3	22,42	31,09
9	31,4	21,57	30,857

Tabelle 4.9: Gegenüberstellung der Feldstärken

4.4 Optimieren

Die Optimierung des Systems zielt darauf ab den Aufenthaltsort des Klienten genauer zu bestimmen. Die Ungenauigkeit soll von 10m auf wenige Meter abnehmen. Hierfür gibt es drei entscheidende Einflussfaktoren.

- Der Erste Faktor ist die Anzahl der APs. Versuche mit MS Radar haben gezeigt das die Genauigkeit der Positionierung durch eine Erhöhung der Anzahl an APs um einen gewissen Grad verbessert werden kann. [Jeh07, vgl. s. 11]

- Der zweite Faktor ist die Anzahl der Referenzpunkte. Versuche, mit Intel Place Lab, haben gezeigt das die Referenzpunkte einen Abstand von maximal 10m nicht überschreiten sollen da sich die Genauigkeit ansonsten rapide verschlechtert. [Jeh07, vgl. s. 13]
- Der dritte und letzte Faktor ist die Anzahl der gemittelte Messwerte für die aktuelle Klientfeldstärke. Mit der Erhöhung der Anzahl der Werte die für den Mittelwert herangezogen werden können Signalschwankungen besser ausgeglichen werden. Der Nachteil an der Erhöhung ist die Zeit die für das Sammeln der Daten benötigt wird. Die Dauer des Lesen der Klientwerte steigt dabei proportional mit der Anzahl an Messwerten.

Neben diesen Methoden sollte eine Verbesserung der Kalibrierung eine Steigerung der Genauigkeit erzielen. Eine Verbesserung der Kalibrierung kann erzielt werden indem z.B.: mehrere Radiomaps, wie unter Punkt 4.3.1 beschrieben, erstellt werden. Weiters könnte man bei der Abstandsberechnung, siehe Abschnitt 4.1.5, eine Funktion suchen die die Messwerte noch exakter darstellt.

Kapitel 5

LBS - Die Komponenten

In diesem Kapitel möchte wird eine Lösung des Problems LBS ohne Klientsoftware vorstellen.

5.1 LBS Systemarchitektur

Als erstes wird ein Überblick über die Systemarchitektur gegeben, siehe Abbildung 5.1.

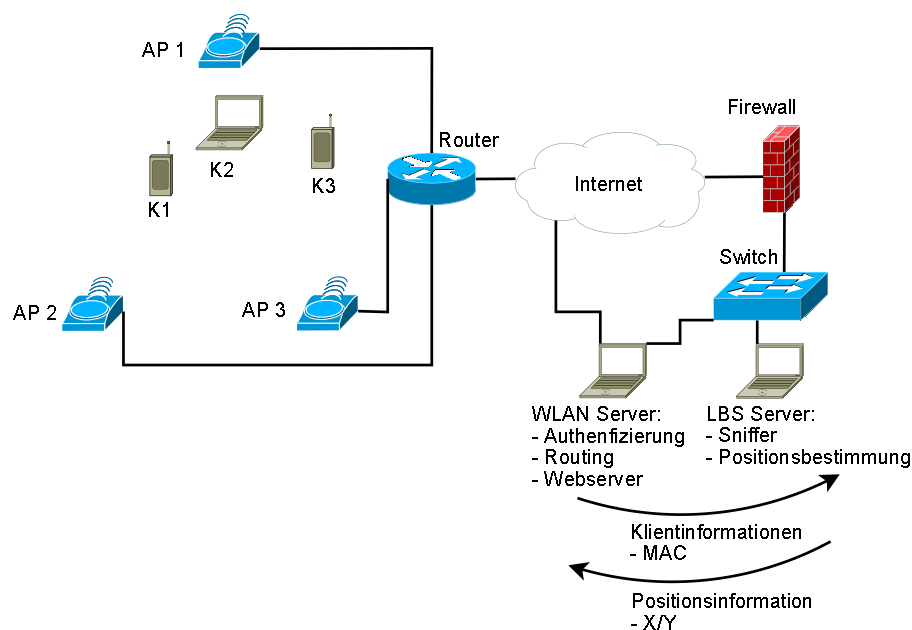


Abbildung 5.1: Systemarchitektur

Die Klienten (K1, K2 und K3) befinden sich im Sendebereich der APs (AP1, AP2 und AP3). Die APs dienen als Sensoren die die IP-Pakete der Klienten auffangen und deren Feldstärke messen. Nach dem messen der Feldstärke senden die APs die Infor-

mationen Feldstärke, MAC Adresse und IP Adresse, an den LBS Server. Dabei wird ein Klient Server Protokoll benutzt. Auf den APs wird der Kismet Server und auf dem LBS Server der Kismet Klient installiert. Kismet ist ein passiver Netzwerk-Sniffer, der zum Aufspüren von 802.11 Netzwerken verwendet werden kann. Die Wahl fiel auf Kismet da dieser Open Source ist und über eine sehr gute Dokumentation verfügt.

Für den Transport der Daten wird zwischen dem Router und der Firewall ein IPSEC Tunnel, siehe Abschnitt 5.7, aufgebaut. Der LBS Server speichert die Feldstärke für jeden Klienten und AP in einer Datenbank ab, siehe Abschnitt 5.5.

Neben dem LBS Server gibt es im „WLAN der Stadt St.Pölten“ den WLAN Server. Dieser ist für die Authentifizierung, das Routing sowie der Präsentation einer Webseite zuständig. Alle Klienten die im WLAN Surfen wollen müssen sich zuerst am WLAN Server authentifizieren werden. Dabei werden deren IP und MAC Adresse gespeichert und dem LBS Server übergeben, siehe Abschnitt 5.6. Der LBS Server sucht in seiner Datenbank die Feldstärke des Klienten zu dem der WLAN Server eine Positionsinformation benötigt und errechnet seine Position. Die Position wird in Form von X/Y Koordinaten an den WLAN Server zurückgegeben.

5.2 LBS Server

5.2.1 Grundsystem

Als Server kommt ein Debian Linux in der aktuellen Stablen Version (etch) zum Einsatz. Zu Empfehlen ist eine Netzwerkinstallation (netinst¹), diese setzt eine bestehende Internetverbindung voraussetzt. Bei der Frage nach dem Partitionierungsschema ist die Auswahl „Separate /home-, /usr-, /var-, und /tmp-Partitionen“ am besten geeignet. Bei der Auswahl der Zusatzsoftware wählt man nichts aus um das System möglichst schlank zu halten.

Ist die Installation abgeschlossen geht es daran die benötigten Programme zu installieren. Als erstes sollte man einen SSH Server installieren um auch remote arbeiten zu können.

5.2.2 Datenbank

Als Datenbank kommt PostgreSQL in der Version 8.1 zum Einsatz. Diese Datenbank wurde gewählt da sie Open Source und sehr stabil ist. Nach der Installation muss ein

¹<http://www.debian.org/CD/netinst/>

Datenbankbenutzer sowie eine Datenbank angelegt werden, dies geschieht mit folgenden Befehlen.

```
1 #su - postgres
2 postgres#createuser lbs --pwprompt
3 Geben Sie das Passwort der neuen Rolle ein:
4 Geben Sie es noch einmal ein:
5 Soll die neue Rolle ein Superuser sein? (j/n) n
6 Soll die neue Rolle Datenbanken erzeugen dürfen? (j/n) n
7 Soll die neue Rolle weitere neue Rollen erzeugen dürfen? (j/n) n
8 CREATE ROLE
9 postgres#createdb lbs --encoding=UTF-8 --owner=lbs
10 CREATE DATABASE
11 postgres#exit
```

Listing 5.1: PostgreSQL konfigurieren

Um mit der Datenbank zu arbeiten eignet sich das Kommandozeilen Tool psql beziehungsweise das grafische Tool pgadmin3 mit welchem man von jedem Rechner im Netzwerk aus auf die Datenbank zugreifen kann.

5.2.3 Perl

Die Programme wurden in Perl verfasst. Perl ist bei der Standardinstallation von Debian in der Version 5.8.8 enthalten, man muss nur zusätzlich das Perl Datenbankmodul für PostgreSQL installieren.

5.3 Access Points installieren

Als APs kamen folgende Geräte zum Einsatz:

- IBM Thinkpad T42p mit mini-PCI WLAN Modul (Chipsatz Atheros 5212)
- Compaq Armada M700 mit PCMCIA WLAN Karte (Chipsatz Atheros 5212)
- Neoware Thinclient mit PCMCIA WLAN Karte (Chipsatz Atheros 5212)

Ich habe mich für den Atheros Chipsatz entschieden da dieser unter Linux sehr gut unterstützt wird. Als Betriebssystem läuft auf den APs ein Debian Linux welches wie unter Punkt 5.2.1 zu installieren ist. Zusätzlich müssen noch der WLAN Treiber Madwifi sowie der Sniffer Kismet installiert werden. Kismet ist ein passiver WLAN Sniffer zum Aufspüren von Funknetzwerken. Er unterliegt der GPL und läuft unter Linux, BSD, MAC und Windows.

5.3.1 WLAN Treiber

Der Treiber für diesen Chipsatz ist unter ² zu bekommen. Hierbei verwende ich die aktuelle Version 0.9.4 des Treiber. Die Installation des Madwifi Treibers ist im Abschnitt B.4 beschrieben.

²<http://madwifi.org/>

5.3.2 Kismet

Als WLAN Sniffer kommt Kismet ³ in der Version 2006.04.R1 zum Einsatz. In der Konfigurationsdatei, die unter „/etc/kismet/kismet.conf“ zu finden ist, müssen folgende Änderungen vorgenommen werden.

```
1 #vi /etc/kismet/kismet.conf
2 source=madwifi_g,wifi0,AtherosG
3 defaultchannels=IEEE80211g:1,7,13,2,8,3,9,4,10,5,11,6,12
4 allowedhosts=127.0.0.1,172.16.0.0/16
5 bindaddress=127.0.0.1,172.16.0.0/16
6 maxclients=1
```

Listing 5.2: Kismet Konfiguration

Weitere Informationen zur Konfiguration von Kismet findet man unter ⁴

5.4 Erstellen der Radiomap - Offlinephase

In der Offlinephase wird die Radiomap erstellt die in der Onlinephase verwendet wird um die Position eines Klienten festzustellen. Die Erstellung der Radiomap ist relativ einfach, zuerst werden auf dem zu vermessenden Bereich Referenzpunkte (Messpunkte) festgelegt. Als nächstes wird der Kismet Server auf den APs gestartet sowie auf dem Server die Kismet Klienten die die gemessenen Werte in einer Datenbank ablegen. Die Tabelle welche die Messwerte speichert hat folgenden Aufbau:

	ID	MAC	SIGNAL	IP	POSITION	AP
Datentypen	id (Primary Key)	macaddr	int4	inet	int4	int4

Tabelle 5.1: Offlinephase Tabellenlayout

Die folgende Aufzählung beschreibt den Inhalt der einzelnen Spalten:

- id: Ist der automatisch generierte Primary Key
- mac: Speichert die MAC Adresse des Klienten
- signal: Speichert die Signalstärke
- ip: Ist die IP Adresse des Klienten
- position: Gibt den Messpunkt an
- ap: Gibt den Access Point an bei dem der Messwert aufgenommen wurde

Hier ein Auszug aus dieser Tabelle.

³<http://www.kismetwireless.net/>

⁴<http://www.kismetwireless.net/documentation.shtml>

	id	mac	signal	ip	position	ap
1	1	00:02:78:85:17:76	35	192.168.1.1	1	1
2	2	00:02:78:85:17:76	30	192.168.1.1	1	2
3	3	00:02:78:85:17:76	32	192.168.1.1	1	1
4	4	00:02:78:85:17:76	37	192.168.1.1	1	2
5	5	00:02:78:85:17:76	35	192.168.1.1	1	1
6	6	00:02:78:85:17:76	39	192.168.1.1	1	3
7	7	00:02:78:85:17:76	30	192.168.1.1	1	2

Listing 5.3: Auszug der Tabelle scan

Die MAC- und IP Adresse wurde aufgenommen um sicherzustellen das sich keine anderen Werte, von fremden Klienten, in die Tabelle befinden. Danach wurde von Messpunkt zu Messpunkt gegangen und Traffic generiert. Der Traffic entstand durch das Betrachten einer Webseite. Für diesen Test wurde auf einem Laptop ein Apache Webserver installiert. Das Programm zur Aufnahme der Daten ist ein modifiziertes Sniffer Programm welches im Abschnitt 5.5 näher erläutert wird. Die Modifikationen werden hier jedoch kurz erklärt. Als erstes muss zu Beginn des Programms angegeben werden mit welchem AP sich das Programm verbindet, dies geschieht mit der variable \$AP, sowie der aktuelle Messpunkt, in der variable \$POS. Das Programm muss für jeden AP einmal gestartet werden und für jeden neuen Messpunkt, mit einem aktuellen Wert in \$POS neu gestartet werden.

Der zweite Unterschied zum Sniffer Programm besteht darin das einfach alle Werte in die Tabelle aufgenommen werden und nicht darauf geachtet wird ob für den Klient bereits ein Eintrag vorhanden wird. Dies geschieht in folgendem Codeabschnitt:

```

1 $sqlinsert = qq/INSERT INTO scan (mac,signal ,ip ,position ,ap) VALUES ( '$MAC' , $Signal , '$IP' , '$POS' , '
  $AP' ) /;
2 $sqlinsert = $dbh->do($sqlinsert)
3 or warn $dbh->errstr;
```

Listing 5.4: Offlinephase Datenaufnahme

Um nicht jedesmal die Mittelwerte bzw. den Erwartungswert und die Standardabweichung neu aus der Tabelle scan errechnen zu müssen werden diese in eine zwei neue Tabellen eingetragen. Hierbei handelt es sich um die Tabelle „data“ und „normalverteilung“. Beispielhaft wird hier die Tabelle „data“ beschrieben. Ihr Aufbau sieht wie folgt aus:

	ID	AP1	AP2	AP3
Datentypen	id (Primary Key)	float4	float4	float4

Tabelle 5.2: Data Tabellenlayout

In den Spalten ap1 bis ap3 werden die Mittelwerte der einzelnen Messpunkte gespeichert, diese haben den Datentyp float4 und nicht int4 da bei der Berechnung der Mittelwerte meist nicht ganzzahlige Zahlen herauskommen. Der Messpunkt für den die Werte gelten ist in der Spalte id gespeichert. Die Berechnung der Mittelwerte und das Eintragen in die Tabelle wird nicht von einem Programm erledigt sondern muss von

Hand gemacht werden. In der „normalverteilung“ Tabelle welche für die Normalverteilungsmethode, siehe Abschnitt 4.1.3, benötigt wird bietet noch zusätzlich drei Felder in denen die Standardabweichung gespeichert wird.

5.5 Lesen der Feldstärke

In diesem Abschnitt erhalten sie einen Einblick in das Sniffer Programm. Dieses Programm baut eine Verbindung zu den Kismet Servern auf und empfängt von diesen die Feldstärke (RSSI), MAC Adresse sowie IP Adresse.

Im Listing 5.2.3 ist zu sehen wie man das DBI Modul, aus Abschnitt B.6 Zeile 8 installiert, alle anderen Module sind bei Perl bereits inkludiert.

Als nächster Schritt wird eine Verbindung zum Kismet Server hergestellt und mittels des Kismet Protokolls das Senden der MAC- und IP Adresse sowie der Feldstärke aktiviert

```

1 $socket = IO::Socket::INET->new(PeerPort=>2501,PeerAddr=>$kismetip,Proto=>'tcp ');
2     or croak ("Fehler beim Socket");
3 $select = IO::Select->new($socket);
4     or croak ("Fehler beim Select");
5 $flags = fcntl($socket,F_GETFL,0);
6     or croak ("Fehler beim holen des Flags");
7 fcntl($socket,F_SETFL,$flags | O_NONBLOCK);
8     or croak ("Fehler bei setzen von Non-Blocking");
9 $protokoll = 'CLIENT';
10 $fields = 'mac,signal,ip';
11 $socket->send(sprintf("!1 ENABLE %s %s\n",$protokoll,$fields),0);
12     or croak ("Fehler bei der Serverkommunikation");$

```

Listing 5.5: Verbindung zum Kismet Server

Zeile 1 öffnet einen Socket zur IP Adresse welche in \$kismetip gespeichert ist. Danach wird die Verbindung mit dem Socket in den Zeilen 3,5 und 7 als „Non Blocking“ aufgebaut. Die Zeilen 9 bis 11 weisen den Kismet Server an aus dem Protokoll „CLIENT“ die Felder „mac,signal,ip“ an den Klient zu senden. Weitere Protokolle und Felder die der Kismet Server beherrscht findet man unter ⁵.

Die Empfangenen Daten werden in einer Datenbank abgelegt. Der Aufbau der Datenbankverbindung ist im Abschnitt B.6 beschrieben.

Die Hauptaufgabe des Programms, das lesen der Feldstärke, wird in der folgenden Schleife abgearbeitet. Hierbei wird vom in der Zeile 3 vom Socket gelesen und in Zeile 7 mit der „split“ Anweisung der gelesen String in 4 Teilstrings zerlegt, wobei nur die letzten 3 Variablen, \$MAC, \$Signal und \$IP, für dieses Programm interessant sind. Danach werden die Daten mittels Regulärer Ausdrücke auf Plausibilität geprüft und anschließend in die Datenbank geschrieben.

In der Datenbank werden die 3 aktuellsten Messwerte pro Klient pro AP gehalten, dazu wird zuerst geprüft ob es für den aktuellen Klient, in Kombination mit dem AP,

⁵http://www.757.org/~joat/wiki/index.php/Kismet_and_Perl

bereits einen Eintrag gibt, dies geschieht in der Zeile 11. Ist kein Eintrag vorhanden wird dieser in Zeile 24 angelegt. Ist bereits ein Eintrag vorhanden wird dieser auf den neuesten Stand gebracht.

```

1 while (!select(undef,undef,undef,0.25))
2 {
3     foreach($select->can_read(0))
4     {
5         $rv = $_->recv($data, POSIX::BUFSIZ,0);
6         die "Kann nicht vom Server lesen" unless defined ($rv);
7         ($Protokoll, $MAC, $Signal, $IP) = split(/\ /,$data);
8
9         --- Ausgabe ausgelassen ---
10
11         if (@row = $dbh->selectrow_array("SELECT pos FROM sniffer WHERE mac='$MAC' and ap
12             = '$ap'"))
13         {
14             $posneu = 0;
15             $zahlselect = $row[0];
16
17             --- Ausgabe ausgelassen ---
18
19             $sqlupdate = qq/UPDATE sniffer SET "signal$zahlselect"='$Signal', ip='$IP
20                 ', pos='$posneu' WHERE mac = '$MAC' and ap='$ap' /;
21             $zahlupdate = $dbh->do($sqlupdate) or warn $dbh->errstr;
22
23             }
24             else
25             {
26                 $sqlinsert = qq/INSERT INTO sniffer (mac,ip,signal1,signal2,signal3,ap,pos
27                     ) VALUES ('$MAC','$IP','$Signal','$Signal','$Signal','$ap',1) /;
28                 $zahlinsert = $dbh->do($sqlinsert) or warn $dbh->errstr;
29             }
30         }
31     }
32 }

```

Listing 5.6: Daten abarbeiten

Die Tabelle die das Programm benutzt hat folgenden Aufbau.

	ID	MAC	IP	SIGNAL1	SIGNAL2	SIGNAL3	AP	POS
Datentypen	id (Primary Key)	macaddr	inet	int4	int4	int4	int4	int4

Tabelle 5.3: Onlinephase Sniffer Tabellenlayout

Wie schon erwähnt hält die Tabelle die drei aktuellsten Messwerte in den Spalten signal1, signal2 und signal3. Das Feld pos hat die Aufgabe die aktuelle Signalspalte zu speichern, damit ist es möglich über alle Signalspalten zu rotieren.

5.6 Identifizierung des Klienten

Um einem Klient einen bestimmten Dienst anzubieten oder eine bestimmte Webseite zu zeigen ist es entscheidend die IP Adresse des Klienten zu kennen. Das Programm im Abschnitt 5.5 liefert die IP Adresse und die dazu gehörende MAC Adresse. Diese Werte legt es in einer Datenbank ab.

Die Skriptsprache PHP welche beim WLAN für die Stadt St.Pölten zum Einsatz kommt ist in der Lage die IP Adresse des Klienten zu bestimmen. Hierzu wird die ab PHP 4.1 vordefinierte Variable `REMOTE_ADDR` verwendet.

```

1 <?
2 $ipadresse = "$REMOTE_ADDR";
3 echo "Ihre IP ist: $ipadresse ";
4 ?>

```

Listing 5.7: IP Adresse mit PHP auslesen

Sollte beim WLAN eine Verschlüsselungstechnologie zum Einsatz kommen ergibt sich das Problem das die IP Adresse nicht mit dem Sniffer Programm gelesen werden kann, der Grund dafür wurde unter Punkt 4.2.3 beschrieben. Diese Problem kann mittels eines Globalen DHCP Servers und der DHCP Relay Funktion des APs gelöst werden. Abbildung 5.2 beschreibt die Funktionsweise von DHCP Relay.

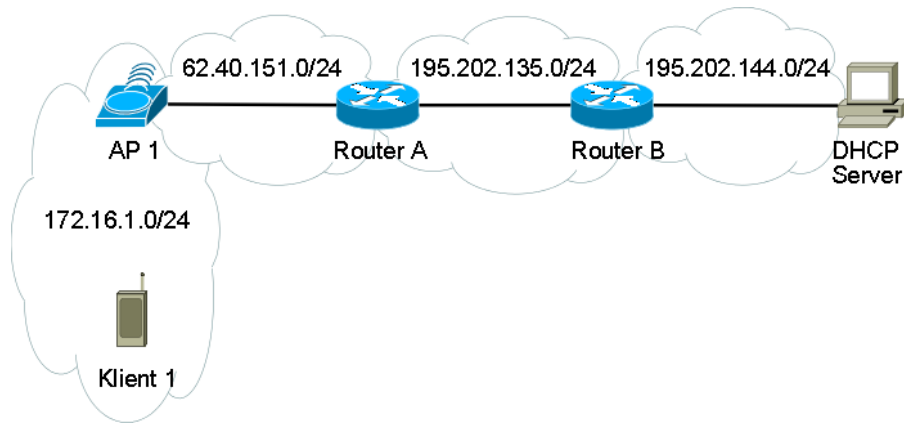


Abbildung 5.2: DHCP Relay

Der Klient sendet eine Broadcast DHCP Anfrage welche von AP1 empfangen wird. Dieser wandelt den Broadcast in eine Unicast um und schickt in an den DHCP Server. Der DHCP Server schickt seine Antwort zum AP1 zurück welcher sie wiederum an den Klient1 weiterleitet. Da im WLAN der Stadt St.Pölten fast ausschließlich Cisco Router 851w zum Einsatz kommen wird die Konfiguration Beispielhaft anhand des Routers 851w erklärt. Eine genau Beschreibung ist unter ⁶ zu finden.

```

1 AP1#conf t
2 AP1(config)#int dot11Radio 0
3 AP1(config-if)#ip helper-address ?
4 A.B.C.D IP destination address
5 global Helper-address is global
6 vrf VRF name for helper-address (if different from interface VRF)

```

Listing 5.8: ip-helper

Folgende Befehl helfen bei der Fehlersuche.

```

1 AP1#show ip helper-address
2 AP1#show dhcp server

```

Listing 5.9: ip-helper-show

Nachdem nun geklärt wurde wie der Klient zu einer IP Adresse kommt wird noch die Installation und Konfiguration eines DHCP Servers beschrieben. Als DHCP Server

⁶http://www.cisco.com/en/US/docs/ios/12_4/ip_addr/configuration/guide/hipdhcpr.html

wird der "dhcp3" verwendet. Die Beschreibung der Installation wird im Abschnitt B.7 beschrieben.

Das Konfigurationsfile des DHCP Servers findet man in folgendem Verzeichnis "/etc/dhcp3/dhcpd.conf". Die grundlegende Konfiguration wird in der Manual Page von "dhcpd.conf", siehe ⁷, beschrieben.

Als letzte Zeile gehört in das Konfigurationsfile folgende eingefügt.

```
1 #vi /etc/dhcp3/dhcpd.conf
2 on commit {execute("/usr/bin/perl","dhcp.pl","leased-address","hardware")}
```

Listing 5.10: DHCP Konfiguration

Diese Zeile ruft nach jeder DHCP Anfrage das Skript dhcp.pl auf und übergibt ihm die Parameter leased-address und "hardware". Hinter diesen Parameter befinden sich die vergebene IP Adresse sowie die MAC Adresse des Klienten der die IP bekommen hat. Das Skript hat dann nur mehr die Aufgabe die beiden Parameter in einer Datenbank abzulegen.

5.7 Transport der Daten

Da sich der Lokalisierungsserver und die APs nicht im selben Netz befinden müssen die Daten zum Server geroutet werden. Würde auf den APs NAT eingesetzt werden wäre es aber unmöglich zu bestimmen welcher Klient welche Daten anfordert. Die Lösung dieses Problems stellt ein side-to-side Tunnel, welcher mit IPSEC realisiert wird dar. Siehe Abbildung 5.3.

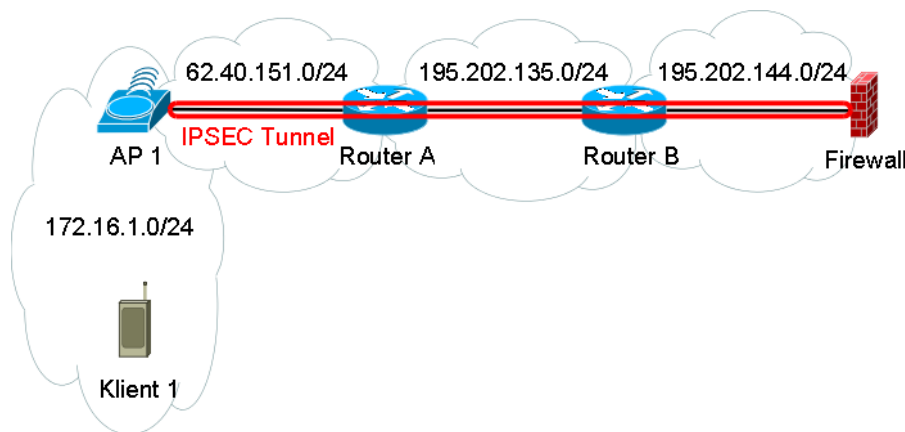


Abbildung 5.3: IPSEC Tunnel

Nun folgt ein Konfigurationsbeispiel für IPSEC auf den Geräte Cisco Router 851w sowie Cisco PIX 515. Als erstes wird die Konfiguration auf dem 851w gefunden. In

⁷<http://www.daemon-systems.org/man/dhcpd.conf.5.html>

den Zeilen 1-4 wird die Konfiguration der Phase 1 (ISAKMP) vorgenommen. Es ist zu beachten das remote address" durch die IP Adresse des Kommunikationspartners ersetzt werden muss. Die Zeilen 6-11 definieren die Phase 2 (IPSEC) zusätzlich muss noch die Access Liste angelegt werden (Zeile 15). Als letzter Schritt muss die Krypto Map noch an ein Interface gebunden werden was in Zeile 13 geschieht.

```
1 crypto isakmp policy 1
2   hash md5
3   authentication pre-share
4   crypto isakmp key DWLFDSSP3 address "remote adress"
5 !
6 crypto ipsec transform-set normal esp-des esp-md5-hmac
7 crypto map fruewald 1 ipsec-isakmp
8   set peer "remote address"
9   set transform-set normal
10  match address 101
11 !
12 interface FastEthernet4
13   crypto map fruewald
14 !
15 access-list 101 permit ip any "remote network"
```

Listing 5.11: Router 851w IPSEC Konfiguration

Die Konfiguration der PIX ist ähnlich der des Routers, deswegen wird die Konfiguration im Abschnitt [B.8](#) beschrieben.

Kapitel 6

Bestimmung der Position

6.1 Teststellung Festsaal

Da es zum Zeitpunkt der Erstellung der Arbeit keine Möglichkeit gab eine Teststellung auf dem Rathausplatz aufzubauen wurde entschieden eine Teststellung im Festsaal der Fachhochschule St.Pölten aufzubauen. Der Festsaal hat im Vergleich zu anderen Räumlichkeiten der Fachhochschule folgende Vorteile. Er ist größer, eben und weist die gleiche Geometrie wie der Rathausplatz auf. Als Alternative Räumen kämen noch das Netzwerktechnik Labor, das Audimax beziehungsweise die Aula in Frage.

6.2 Aufbau

Beim Aufbau im Festsaal der Fachhochschule, siehe Abbildung 6.1 kamen folgende Geräte zum Einsatz.

BEZEICHNUNG	GERÄT
AP1	IBM Thinkpad T42p
AP2	Neoware Thinclient
AP3	Compaq Armada M700
AP4	Cisco 1230AG
PDA	Acer n50
LBS Server	Compaq Armada M700
Webserver	IBM Thinkpad T42p
Switch	Cisco 2950

Tabelle 6.1: Legende zu Abbildung 6.1

Um einen Überblick über die Teststellung zu bekommen zeigen die Abbildungen 6.2 und 6.3 den Festsaal der Fachhochschule St.Pölten. In den Abbildungen 6.4 und 6.5 sind die APs eins und drei zu sehen.

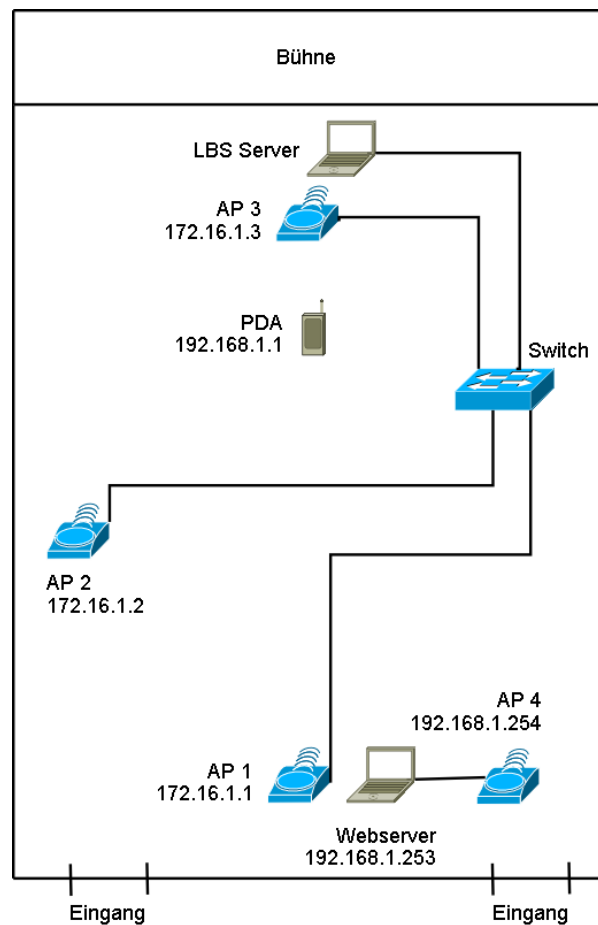


Abbildung 6.1: Festsaal Aufbau

Im ersten Arbeitsschritt werden Referenzmessungen an den Messpunkten durchgeführt.

6.3 Referenzpunkte aufnehmen

Es wurden an jedem Referenzpunkt mindestens zehn Werte aufgenommen. Der Abschnitt 5.4 beschreibt das Snifferprogramm zur Datensammlung. Die Abbildung 6.6 zeigt die Anordnung der Messpunkte sowie deren Abstände zueinander und zu den APs.

6.4 Methode 1: Differenzmethode mit Mittelwerten

Die Theoretische Beschreibung der Differenzmethode findet man unter Punkt 4.1.1. Hier wird die Praktische Umsetzung sowie die erzielten Ergebnisse beschrieben. Alle folgenden Programme besitzen gleiche Codeabschnitte die nur einmal hier beschrie-



Abbildung 6.2: Festsaal linke Seite



Abbildung 6.3: Festsaal rechte Seite

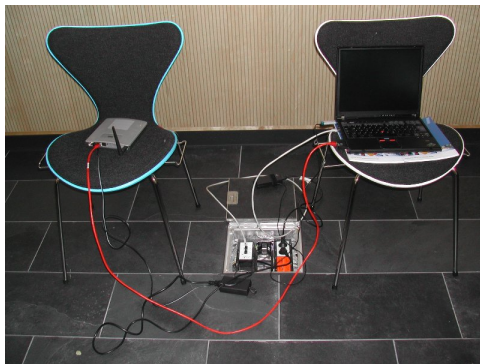


Abbildung 6.4: AP1, AP4 und Webserver

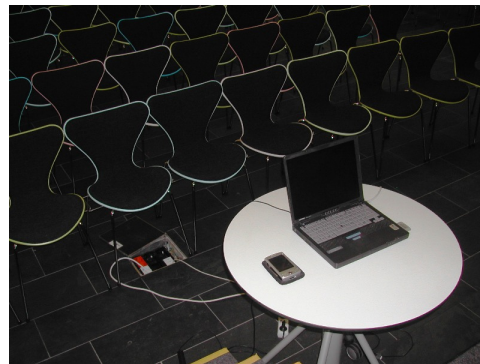


Abbildung 6.5: AP3 und LBS Server

ben werden.

Zu Beginn des Programms werden Module geladen die Perl für die Abarbeitung des Programms benötigt. Als nächstes wird eine Verbindung zur Datenbank hergestellt und aus dieser die normalisierten Messwerte an den Referenzpunkten und die aktuelle Klientfeldstärke gelesen, siehe Listing 6.1 und 6.2. Die Feldstärke des Klienten pro AP wird in den Variablen „c1ap1“, „c1ap2“ und „c1ap3“ gespeichert.

```

1 $query = "SELECT ap1,ap2,ap3 FROM data";
2 $sth = $dbh->prepare($query);
3 $sth->execute();
4 $sth->bind_columns(undef,\ $ap1data ,\ $ap2data ,\ $ap3data);
5 while( $sth->fetch())
6 {
7     push (@ap1, $ap1data);
8     push (@ap2, $ap2data);
9     push (@ap3, $ap3data);
10 }
```

Listing 6.1: Werte aus Datenbank lesen

```

1 $sth->bind_columns(undef,\ $signal1 ,\ $signal2 ,\ $signal3 ,\ $sniffer);
2 while ( $sth->fetch())
3 {
4     if ( $sniffer == 1)
5     {
6         $c1ap1 = ($signal1 + $signal2 + $signal3) / 3;
7     }
8     if ( $sniffer == 2)
9     {
```

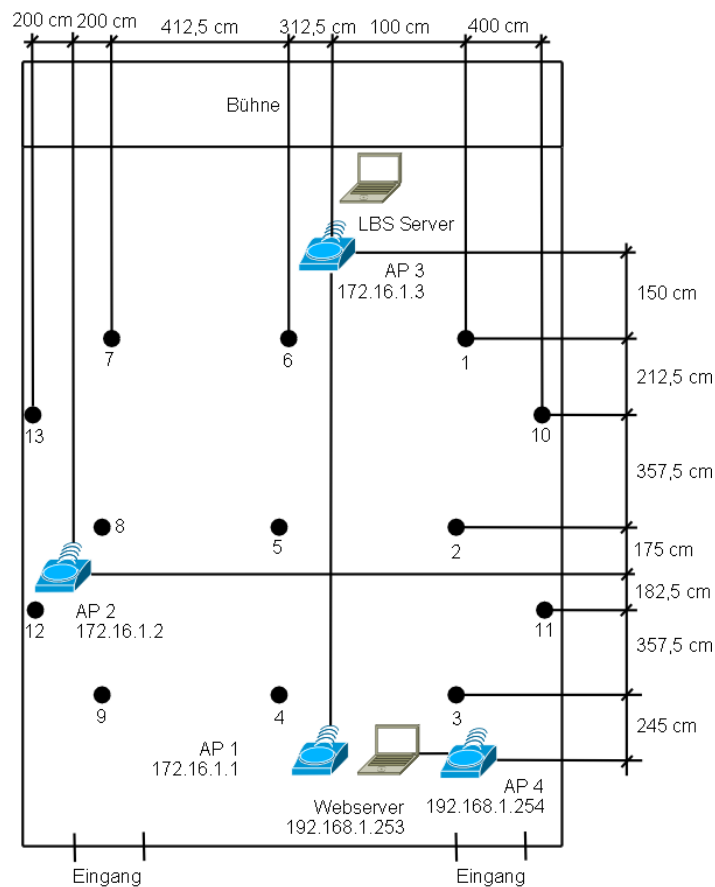


Abbildung 6.6: Referenzpunkte

```

9          $clap2 = ($signal1 + $signal2 + $signal3) / 3;
10       }
11       if ($sniffer == 3)
12       {
13           $clap3 = ($signal1 + $signal2 + $signal3) / 3;
14       }

```

Listing 6.2: Klientwerte lesen

Nachdem nun die Referenzwerte und die Werte des Klienten bekannt sind können die Arrays mit den Diff-Werten, wie in Tabelle 4.3 zu sehen, angelegt werden. Bevor mit der Positionsbestimmung begonnen werden kann muss das Array mit den aufsummierten „diff“ Werten angelegt werden, dies geschieht in Listing 6.4.

```

1 foreach (@ap1)
2 {
3     $diff1 = $clap1 - $_;
4     push (@diffap1, $diff1);
5 }
6 foreach (@ap2)
7 {
8     $diff2 = $clap2 - $_;
9     push (@diffap2, $diff2);
10 }
11 foreach (@ap3)
12 {
13     $diff3 = $clap3 - $_;
14     push (@diffap3, $diff3);
15 }

```

Listing 6.3: Arrays mit Diffwerten anlegen

```

1 for ($i=0;$i<13;$i++)
2 {
3     $summe = $diffap1[$i] + $diffap2[$i] + $diffap3[$i];
4     push (@summediff, $summe);
5 }$

```

Listing 6.4: Anlegen des endgültigen Summenarrays

Zuletzt kommt das Herzstück dieses Programms. Der Algorithmus welcher im „summediff“ Array nach dem Wert sucht der am nächsten bei Null liegt, siehe Listing 6.5.

```

1 $x = 0;
2 $akt = $summediff[0];
3 $pos = 0;
4
5 foreach (@summediff)
6 {
7     if (abs($akt) < abs($_))
8     {
9         $x++;
10    }
11    elseif (abs($akt) > abs($_))
12    {
13        $akt = $_;
14        $pos = $x;
15        $x++;
16    }
17    else
18    {
19        $x++;
20    }
21 }$
22 print "Position: $pos\n";

```

Listing 6.5: Positionssuche

Hier eine kurze Beschreibung der wichtigsten Variablen. Da im summediff-Array auch Negative Werte stehen können arbeitet der Algorithmus mit absoluten Werten. Abbildung 6.7 beschreibt die Arbeitsweise des Algorithmus aus Listing 6.5.

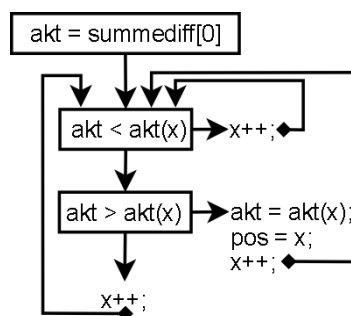


Abbildung 6.7: Differenzmethode Positionsbestimmung

- x: Hält die aktuelle Position im „summediff“ Array und wird bei jedem Schleifendurchlauf erhöht.
- akt: Beinhaltet die aktuell kleinste Zahl.

- akt(x): Speichert die Zahl aus dem summediff Array mit der gerade gearbeitet wird.
- pos: Hält die Position der aktuell kleinsten Zahl im Array.

6.4.1 Vorgehensweise

Hier wird beschrieben wie die einzelnen Komponenten zusammenspielen beziehungsweise gestartet werden. Als erstes muss auf den APs der Kismet Server gestartet werden, diese Aufgabe erledigt das Skript in Listing B.12. Als nächstes muss das Snifferprogramm aus Abschnitt 5.5 gestartet werden. Beim starten muss dem Programm die Nummer sowie die IP Adresse des APs übergeben werden. Wurden diese zwei Aufgaben erledigt kann mit der Positionsbestimmung begonnen werden. Als erstes muss der Klient an irgendeiner Position im Festsaal Traffic generieren welcher von den APs aufgefangen wird. Nachdem jeder AP mindestens drei Messwerte aufgenommen hat kann das Positionierungsprogramm gestartet werden. Hierzu führt man auf der Eingabezeile unter Linux folgenden Befehl aus.

```
1 #perl auswertung.pl
```

Listing 6.6: Positionierungsprogramm starten

In der Tabelle 6.2 sind die Feldsträkwerte zu sehen mit denen das Programm arbeitet. Diese Werte stammen aus Tabelle 6.4 die drei Einträge pro AP aufsummiert und durch 3 geteilt wurden um den Mittelwert zu erhalten.

	AP1	AP2	AP3
Klient	33,33	35,33	38,33

Tabelle 6.2: Beispiel - Differenzmethode Klient

Die Tabelle 6.3 zeigt die programminternen Strukturen. Die Werte AP1, AP2 und AP3 sind die Mittelwerte aus der Offlinephase. In den Spalten Diff.AP1, Diff.AP2 und Diff.AP3 stehen die Differenzen von den Klientwerten und den Werten der Offlinephase.

In der Spalte Summe wurden die Werte Diff.AP1, Diff.AP2 und Diff.AP3 aufsummiert. Der Algorithmus aus der Abbildung 6.7 sucht aus dieser Spalte den Wert der am nächsten bei Null liegt und gibt dessen ID zurück. Die ID ist gleichbedeutend mit dem Referenzpunkt. In diesem Beispiel ist das Ergebnis der Referenzpunkt 13.

6.4.2 Ergebnisse

Um vergleichbare Ergebnisse zu erzielen wurde der Klient genau beim Referenzpunkt fünf positioniert und drei Messwerte durch das Snifferprogramm in der Datenbank ge-

ID	AP1	AP2	AP3	DIFF. AP1	DIFF. AP2	DIFF. AP3	Summe
1	34,92	36,36	40,46	-1,59	-1,03	-2,13	-4,75
2	37,71	34,38	31,5	-4,38	0,96	6,83	3,41
3	45,7	34,91	28,83	-12,37	0,43	9,5	-2,44
4	45,6	36,88	28,14	-12,27	-1,55	10,19	-3,63
5	36,14	40	30,13	-2,81	-4,67	8,2	0,72
6	35	39	37,23	-1,67	-3,67	1,1	-4,24
7	33,1	40,11	36,75	0,23	-4,78	1,58	-2,97
8	34,82	44,28	34,13	-1,49	-8,95	4,2	-6,24
9	37,05	38,38	32,67	-3,72	-3,05	5,66	-1,11
10	37,89	39,09	35,43	-4,56	-3,76	2,9	-5,42
11	42,55	33,6	26,11	-9,22	1,73	12,12	4,63
12	40,23	42,48	35,4	-6,9	-7,15	2,9	-11,15
13	33,53	39,25	34,31	-0,2	-3,92	4,02	-0,1

Tabelle 6.3: Beispiel - Differenzmethode

speichert. Alle folgenden Programme arbeiten mit diesen Werten. Für den Referenzpunkt fünf sieht die Datenbank in der die Klientwerte gespeichert werden wie folgt aus.

ID	MAC	IP	SIGNAL1	SIGNAL2	SIGNAL3	AP	POS
1	00:02:78:85:17:76	192.168.1.1	40	40	35	3	3
2	00:02:78:85:17:76	192.168.1.1	34	32	34	1	1
3	00:02:78:85:17:76	192.168.1.1	32	40	34	2	1

Tabelle 6.4: Messpunkt 5 - Sniffer Tabellenlayout

Wenn von X und Y Abständen gesprochen wird sind diese immer aus Sicht von AP1 zu sehen, wobei +X in Richtung Bühne zeigt und +Y in Richtung Mensa. Der Messpunkt 5 hat die X/Y Koordinaten 1530/312,5cm.

Als Ergebnis liefert die erste Methode den Referenzpunkt 13 welcher bei den Koordinaten 1317,5/1125 liegt. Der Abstand dieser Punkte kann mit der Formel 6.1 bestimmt werden.

$$Fehler = \sqrt{\Delta x^2 + \Delta y^2} = \sqrt{(1317,5 - 1530)^2 + (1125 - 312,5)^2} = 839,8cm \quad (6.1)$$

Die Differenzmethode mit Mittelwerten bestimmt den Aufenthaltsort mit einer Genauigkeit von 8,4m.

6.5 Methode 2: Trilateration mit Mittelwerten

Bei der Trilaterationsmethode müssen zuerst die Abstände der Referenzpunkte zu den APs aufgenommen werden. Die Abstände sind in der Abbildung 6.6 ersichtlich. Diese Abstände werden als Array in das Programm aufgenommen, siehe Listing 6.7

```
1 @ap1meter=(1533.3,965.2,246.6,397.1,1009.6,1561.6,1693.1,1203,765.3,1409.2,782.9,1276.2,1732.5);
2 @ap2meter=(1187.7,941.4,1071.1,744.5,541.6,904.3,751.7,201.6,549.2,1428,1337.5,351.1,611.2);
3 @ap3meter=(180.3,726.9,1438.5,1468.6,784.9,346.6,740.4,1021.8,1607.7,617.6,1187.9,1557.8,1181.9);
```

Listing 6.7: Abstände zu den Referenzpunkten

Wie bei der Differenzmethode müssen die Messwerte an den Referenzpunkten aus der Datenbank geholt werden und in einem Array abgespeichert werden, der Quellcode dazu ist in Listing 6.1 zu sehen. Die aktuellen Feldstärkewerte müssen ebenfalls aus der Datenbank geholt werden, dies ist auch bei allen Programmen gleich und in Listing 6.2 zu sehen.

Danach wird pro AP ein Array angelegt in dem die Differenz von aktuellem Klientwert und jedem Referenzpunkt gespeichert wird. In diesen drei Array wird nun nach dem Eintrag gesucht der am nächsten bei Null liegt. Die geschieht mit dem Algorithmus aus Listing 6.5. Somit ist nun aus Sicht jedes APs der Referenzpunkt bekannt der dem Aufenthaltsort des Klienten am nächsten kommt. Also z.B.: aus Sicht von AP1 steht der Klient bei Referenzpunkt 2, aus Sicht von AP2 beim Referenzpunkt 3 und aus Sicht von AP3 bei Referenzpunkt 4. Da im Listing 6.7 die Abstände von APs zu Referenzpunkten abgelegt sind kann nun mit diesen in die Formeln, 4.7 und 4.9, der Trilateration eingesetzt werden, siehe Listing 6.8. Diese Formeln liefern dann den X und Y Abstand in Zentimetern vom AP1.

```
1 $x = ((( $r1*$r1 ) - ($r2*$r2) + ($d*$d) ) / (2*$d));
2 $y = ((( ($r1*$r1) - ($r3*$r3) + ($i*$i) + ($j*$j) ) / (2*$j) ) - (( $i / $j ) * $x));
3 print "x:$x y:$y\n"; $
```

Listing 6.8: Trilateration

6.5.1 Vorgehensweise

Das Programm sucht zu jedem der drei Messpunkte einen Referenzpunkt der möglichst nahe am Aufenthaltsort des Klienten liegt. Hierzu wird, wie bei der vorherigen Methode, eine Array mit Differenzwerten angelegt, siehe Tabelle 6.3. Bei diesem Beispiel wird jedoch nicht eine Summe über diese Arrays gebildet sondern in jeder einzelnen Diff.AP Spalte der Wert gesucht der am nächsten bei Null liegt.

Mit dem errechneten Referenzpunkt aus Tabelle 6.5 kann man im Listing 6.7 den Abstand von jedem AP zum Referenzpunkt suchen, siehe Abbildung 6.8.

Die Abstände aus Tabelle 6.6 werden als r1, r2 und r3 in die Formeln (4.7, 4.9) der Trilateration eingesetzt, siehe Listing 6.8.

AP	REFERENZPUNKT	KLEINSTER WERT DER DIFF.AP SPALTEN AUS TABELLE 6.3
1	13	-0,2
2	3	0,43
3	6	1,1

Tabelle 6.5: Beispiel - Trilateration mit Mittelwerten - Aufenthalt

AP	REFERENZPUNKT	ABSTAND
1	13	1732,5cm
2	3	1071,1cm
3	6	346,6cm

Tabelle 6.6: Beispiel - Trilateration mit Mittelwerten - Abstände

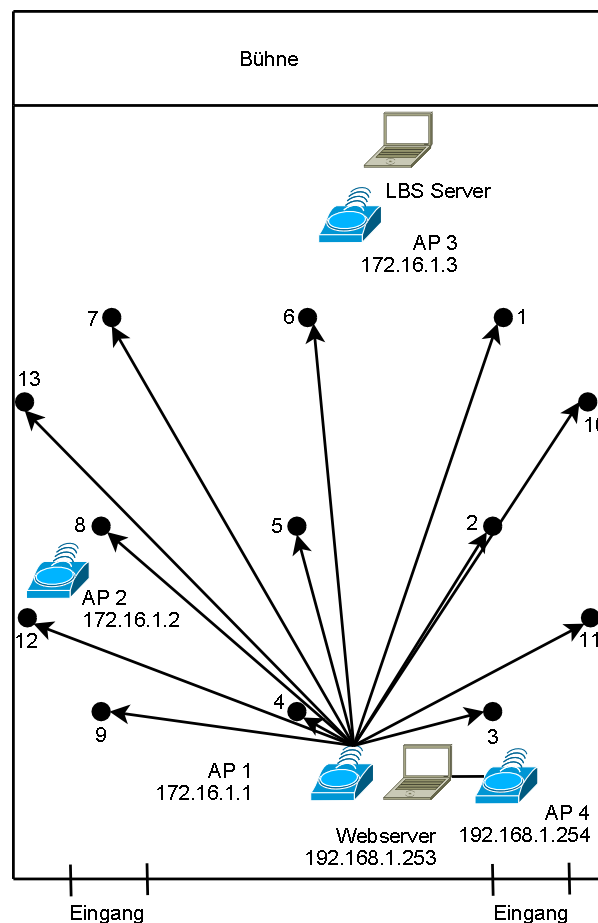


Abbildung 6.8: Abstände vom AP1 zu den Referenzpunkten findet man in Listing 6.7

6.5.2 Ergebnisse

Hier wird wie schon bei der vorherigen Methode der Abstand zwischen reeller und errechneter Position verglichen. Das Programm gibt den Aufenthaltsort an bei X/Y

1698/294. Berechnet man mit der Formel 6.1 den Abstand zum wirklichen Aufenthaltsort bei Messpunkt fünf ergibt sich eine Differenz von 1,7m.

6.6 Methode 3: Trilateration mit Normalverteilung

Wie die Methode „Trilateration mit Mittelwerten“ in Abschnitt 6.5, wird zur Positions-berechnung die Trilateration verwendet. Anstatt jedoch nur mit den Mittelwerten der einzelnen Referenzpunkte zu arbeiten wird bei dieser Methode, in der Offlinephase, für jeden Referenzpunkt ein Erwartungswert sowie eine Standardabweichung berechnet und in der Datenbank gespeichert, siehe Tabelle 4.4.

Im Programm werden diese Werte aus der Tabelle gelesen und in einem Array abgespeichert, dies ist zu sehen in Listing 6.9.

```

1 $query = "SELECT ap1e,ap2e,ap3e,ap1s,ap2s,ap3s FROM normalverteilung ";
2 $sth = $dbh->prepare($query);
3 $sth->execute();
4 $sth->bind_columns(undef,\ $ap1e,\ $ap2e,\ $ap3e,\ $ap1s,\ $ap2s,\ $ap3s);
5 while($sth->fetch())
6 {
7     push (@ap1e,$ap1e);
8     push (@ap2e,$ap2e);
9     push (@ap3e,$ap3e);
10    push (@ap1s,$ap1s);
11    push (@ap2s,$ap2s);
12    push (@ap3s,$ap3s);
13 }$

```

Listing 6.9: Normalverteilungswerte aus Tabelle laden

Um arbeiten zu können müssen noch die aktuellen Werte des Klienten aus der Tabelle Sniffer geholt werden, die ist zu sehen in Listing 6.2.

Der Algorithmus welcher hier eingesetzt wird sucht aus Sicht jedes APs den Messpunkt der die höchste Wahrscheinlichkeit besitzt das sich der Klient dort aufhält. Als Beispiel wird hier die Suche vom AP1 gezeigt.

```

1 $pp1 = 0.0;
2 $pneu1 = 0.0;
3 $position1 = 0;
4 for ($i=0;$i<13;$i++)
5 {
6     $pneu1 = &rechne($clap1,$ap1e[$i],$ap1s[$i]);
7     if ($pp1 > $pneu1)
8     {
9         #pp größer als pneu -> tue nichts
10    }
11    elseif ($pp1 < $pneu1)
12    {
13        $pp1 = $pneu1;
14        $position1 = $i;
15    }
16    else
17    {
18        #pp = pneu -> tue nichts
19    }
20 }

```

Listing 6.10: Bestimmung der Aufenthaltswahrscheinlichkeit

Zu Beginn des Programms werden 3 Hilfsvariablen erstellt. Der Interessanteste Codeabschnitt steht in Zeile 6, hier wird die Funktion „rechne“ aufgerufen. Diese Funktion berechnet die Wahrscheinlichkeit das sich der Klient mit der Feldstärke, welche

in der Variable `c1ap1` gespeichert ist, beim aktuellen Messpunkt aufhlt. Der Wahrscheinlichkeitswert wird in der Variable `pneu` abgelegt. Die `if`-Anweisungen von Zeile 7 bis 19 vergleichen dann den gerade berechneten Wert mit der bis dato groten Wahrscheinlichkeit. Ist die neue Wahrscheinlichkeit kleiner als die bis dato grote wird nichts unternommen. Sollte die neue Wahrscheinlichkeit jedoch groer sein als die aktuell grote wird diese zur aktuellen Wahrscheinlichkeit. Der dazugehrende Referenzpunkt wird in der Variable `i` abgelegt.

Wie die Funktion „rechne“ arbeitet wird in folgendem Listing 6.11 gezeigt.

```

1 sub rechne
2 {
3   $negativ = 0;
4   $column = 0;
5   @{"0.0"} = (0.5000, 0.5040, 0.5080, 0.5120, 0.5160, 0.5199, 0.5239, 0.5279, 0.5319, 0.5359);
6   @{"0.1"} = (0.5398, 0.5438, 0.5478, 0.5517, 0.5557, 0.5596, 0.5636, 0.5675, 0.5714, 0.5753);
7   @{"0.2"} = (0.5793, 0.5832, 0.5871, 0.5910, 0.5948, 0.5987, 0.6026, 0.6064, 0.6103, 0.6141);
8
9   — Ausgabe ausgelassen —
10
11  @{"3.2"} = (0.9993, 0.9993, 0.9994, 0.9994, 0.9994, 0.9994, 0.9994, 0.9995, 0.9995, 0.9995);
12  @{"3.3"} = (0.9995, 0.9995, 0.9995, 0.9996, 0.9996, 0.9996, 0.9996, 0.9996, 0.9996, 0.9997);
13  @{"3.4"} = (0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9998);
14
15  $x = $_[0];
16  $e = $_[1];
17  $v = $_[2];
18
19  $u = (($x+0.5) - $e)/$v;
20  if ($u < 0.0)
21  {
22      $u *= -1;
23      $negativ = 1;
24  }
25  if ($u > 3.5)
26  {
27      $u = 3.39;
28  }
29  $row = sprintf("%.1f", $u);
30  $column = substr($row, 3, 1);
31  if ($negativ == 0)
32  {
33      $p1 = ${"$row"}[$column];
34  }
35  else
36  {
37      $p1 = 1 - ${"$row"}[$column];
38      $negativ = 0;
39  }
40
41  ### TEIL 2
42
43  $u = (($x-0.5) - $e)/$v;
44  if ($u < 0.0)
45  {
46      $u *= -1;
47      $negativ = 1;
48  }
49  if ($u > 3.5)
50  {
51      $u = 3.39;
52  }
53  $row = sprintf("%.1f", $u);
54  $column = substr($row, 3, 1);
55  if ($negativ == 0)
56  {
57      $p2 = ${"$row"}[$column];
58  }
59  else
60  {
61      $p2 = 1 - ${"$row"}[$column];
62      $negativ = 0;
63  }
64  $p = ($p1 - $p2);
65  return $p;
66 }

```

Listing 6.11: Wahrscheinlichkeits-Algorithmus

Im Listing 6.11 sind die interessanten Abschnitte in den Zeilen 19 und 43. Mit diesen Berechnungen und der Normalverteilungstabelle wird die Aufenthaltswahrscheinlichkeit berechnet.

Der letzte Teil des Programms berechnet dann mittels Trilateration die Position des Klienten. Dazu werden, wie schon in Abschnitt 6.5 die Referenzpunkte in Entfernungen umgerechnet und in die Formeln für die Trilateration eingesetzt.

6.6.1 Vorgehensweise

Wie schon beschrieben unterscheidet sich diese Methode in der Offlinephase. Hier werden nicht die Mittelwerte an den einzelnen Referenzpunkten gespeichert sondern der Erwartungswert und die Standardabweichung. Die Tabelle 6.7 zeigt die Offlinephase dieser Methode. In den Spalten AP1E, AP2E und AP3E ist der Erwartungswert für jeden Referenzpunkt gespeichert. Die Spalten AP1S, AP2S und AP3S speichern die Standardabweichung zum jeweiligen Erwartungswert.

ID	AP1E	AP2E	AP3E	AP1S	AP2S	AP3S
1	34,92	36,36	40,46	4,36	3,83	1,85
2	37,71	34,38	31,5	3,09	7,29	1,97
3	45,7	34,91	28,83	6,18	3,94	3,46
4	45,6	36,88	28,14	6,29	3,23	4,49
5	36,14	40	30,13	1,86	4,16	4,45
6	35	39	37,23	4,35	4,4	4,59
7	33,1	40,11	36,75	4,3	4,44	4,06
8	34,82	44,28	34,13	3,47	5,6	3,44
9	37,05	38,38	32,67	5,4	5,19	3,99
10	37,89	39,09	35,43	4,17	5,39	3,98
11	42,55	33,6	26,11	4,76	4,14	4,01
12	40,23	42,48	35,4	3,44	3,63	5,77
13	33,53	39,25	34,31	4,33	2,7	3,57

Tabelle 6.7: Normalverteilung Offlinephase

Der hier vorgestellte Algorithmus sucht für jeden AP den Referenzpunkt bei dem der Klient die höchste Aufenthaltswahrscheinlichkeit besitzt. Das Ergebnis dieser Suche stellt Tabelle 6.8 dar.

In der Tabelle 6.9 sind die Abstände der Referenzpunkte zu den jeweiligen APs zu sehen.

AP	REFERNZPUNKT
1	13
2	1
3	10

Tabelle 6.8: Beispiel - Trilateration mit Normalverteilung - Aufenthalt

AP	REFERNZPUNKT	ABSTAND
1	13	1732,5cm
2	1	1187,7cm
3	10	617,6cm

Tabelle 6.9: Beispiel - Trilateration mit Normalverteilung - Abstände

6.6.2 Ergebnisse

Lässt man das Programm ablaufen errechnet es den Aufenthaltsort bei der Position 1622,3/206,5cm. Das Bedeutet einen Fehler von 1,4m.

6.7 Methode 4: Durchschnittsposition mit Normalverteilung

Bei der Methode „Durchschnittsposition mit Normalverteilung“ wird wie unter Punkt 6.6 für jeden AP der Referenzpunkt gesucht welche die höchste Aufenthaltswahrscheinlichkeit des Klienten besitzt. Des weiteren sind im Programm die X und Y Abstände der Referenzpunkte zu AP1 in zwei Arrays gespeichert.

```
1 @ap1x = (-100, -100, -100, 312.5, 312.5, 312.5, 725, 725, 725, -500, -500, 1125, 1125);
2 @ap1y = (1530, 960, 245, 245, 960, 1530, 1530, 960, 245, 1317.5, 602.5, 602.5, 1317.5);
```

Listing 6.12: X und Y Abstände zu AP1

Die Position errechnet das Programm, wie in Abschnitt 4.1.4 beschrieben, indem es die X und Y aufsummiert und durch drei teilt, siehe Listing 6.13.

```
1 $x = ($ap1x[$position1] + $ap1x[$position2] + $ap1x[$position3]) / 3;
2 $y = ($ap1y[$position1] + $ap1y[$position2] + $ap1y[$position3]) / 3;
3 print "x:$x y:$y\n";
```

Listing 6.13: Berechnen der Position

6.7.1 Vorgehensweise

Bei dieser Methoden werden wie bei der vorherigen zuerst die Referenzpunkte gesucht die die höchste Aufenthaltswahrscheinlichkeit besitzen. Diese Aufgabe übernimmt der gleiche Algorithmus, deswegen können die Ergebnisse aus Tabelle 6.8 auch bei dieser Methode angewendet werden.

Der Unterschied liegt in der Positionsberechnung. In der vorherigen Methode wurde die Trilateration verwendet. Bei der Methode „Durchschnittsposition mit Normalverteilung“ werden die X und Y Abstände der gefunden Referenzpunkte aufsummiert und durch drei geteilt, siehe Tabelle 6.10. Die X und Y Abstände werden dem Listing 6.12 entnommen.

	AP	REFERENZPUNKT	X ABSTAND	Y ABSTAND
	1	13	1125cm	1317,5cm
	2	1	-100cm	1530cm
	3	10	-500cm	1317,5cm
Summe			525cm	4165cm
Summe / 3			175cm	1388,3cm

Tabelle 6.10: Beispiel - Durchschnittsposition - Ergebnisse

6.7.2 Ergebnisse

Die Durchschnittsposition mit Normalverteilung errechnet die Position des Klienten bei den Koordinaten 175/1388,3cm. Somit beträgt der Fehler bei der Positionsbestimmung 17,3m.

6.8 Methode 5: Trilateration mit Abstandsberechnung

Bei dieser Methode kommt wie schon bei den Methoden 2 und 3 die Trilateration zum Einsatz. Der Unterschied ist jedoch das die Abstände zum AP nicht über Referenzpunkte ermittelt werden sondern mit einer Formel berechnet werden. Wie man diese Formel erhält wurde bereits im Abschnitt 4.1.5 beschrieben.

Die ersten Codezeilen des Programms sind wie bei den vorherigen. Als erstes müssen die benötigten Module geladen werden, danach muss eine Verbindung zur Datenbank aufgebaut werden und die aktuellen Feldstärkewerte des Klienten gelesen werden. Nachdem das Erledigt ist wird mit der Formel 4.13 ein Array angelegt in dem für 10cm Abstände die Feldstärke berechnet ist, siehe Listing 6.14. Dabei werden die Koeffizienten a_0 bis a_4 aus der Tabelle 4.7 verwendet.

```

1 $a0 = 44.0588;
2 $a1 = 32.4025;
3 $a2 = -0.0998;
4 $a3 = 0.0033;
5 $a4 = -19.5541;
6 for ($i=0;$i <= 20;$i = $i+0.1)
7 {
8     $apcm = $a0 + $a1*exp(-$i) + $a2*($i*$i) + $a3*($i*$i*$i) + $a4*$i*exp(-$i);
9     push (@apabstand, $apcm);
10 }
```

Listing 6.14: Array mit Feldstärkewerten anlegen

Nach dem Anlegen des Arrays muss man den Wert im Array finden welcher der aktuellen Feldstärke des Klienten am nächsten kommt. Diese Aufgabe übernimmt das Listing 6.15. Es gibt die Position des Wertes an welcher der Klient Feldstärke am nächsten liegt. Das Listing 6.15 muss für alle drei APs abgearbeitet werden, hier ist nur der Code für den AP1 aufgelistet.

```

1 $pos1 = 0;
2 for ($ap1=0;$ap1<=180;$ap1++)
3 {
4     $ap1plus = $ap1 + 1;
5     if ($clap1 < $apabstand[$ap1] && $clap1 > $apabstand[$ap1plus])
6     {
7         $pos1 = $ap1;
8     }
9 }

```

Listing 6.15: Abstand aus Tabelle suchen

Das Ergebnis aus Listing 6.15 muss nun mit zehn Multipliziert werden um auf den Abstand in Zentimetern vom AP zu kommen. Die Multiplikation mit zehn muss vorgenommen werden da im Array @apabstand nur Werte alle 10cm aufgenommen wurden. Die Genauigkeit des Ergebnisses beträgt also +/- 5cm. Hat man von jedem AP den Abstand zum Klient in der Tabelle gefunden wird die Trilateration benutzt die Position des Klienten zu bestimmen.

```

1 $r1 = $pos1 * 10;
2 $r2 = $pos2 * 10;
3 $r3 = $pos3 * 10;
4 $d = 1730;
5 $i = 785;
6 $j = 825;
7 $x = ((($r1*$r1)-($r2*$r2)+($d*$d))/(2*$d);
8 $y = (((($r1*$r1)-($r3*$r3)+($i*$i)+($j*$j))/(2*$j))-((($i/$j)*$x));
9 print "x:$x y:$y\n";

```

Listing 6.16: Trilateration durchführen

Das Ergebnis ist die Entfernung aus der Sicht von AP1 in Zentimeter.

6.8.1 Vorgehensweise

Bei dieser Methode wird der Abstand zwischen jedem AP und dem Klienten mit der Formel 4.13 berechnet. Hierzu wird mit der Formeln ein Array bestehend aus Abständen und Feldstärkewerten angelegt, siehe Tabelle 6.11.

Mit den gemittelten Feldstärkewerten des Klienten aus Tabelle 6.2 kann in diesem Array der passende Abstand gesucht werden.

Mit den Abständen aus Tabelle 6.12 wird eine Trilateration durchgeführt um den Standort des Klienten zu bestimmen.

6.8.2 Ergebnisse

Die fünfte Methode errechnet den Aufenthaltsort beim Punkt 1213,7/-19,5cm. Das bedeutet eine Abstand zum wirklichen Aufenthaltsort von 4,6m.

ABSTAND IN DEZIMETERN	FELDESTÄRKE
0	76,46
0.1	71,61
0.2	67,38
0.3	63,71
0.4	60,52
0.5	57,76
0.6	55,37
0.7	53,31
0.8	51,53
0.9	49,99
1	48,69
...	...
19.6	30,57
19.7	30,56
19.8	30,55
19.9	30,54

Tabelle 6.11: Berechnete Abstandswerte

SENSOR	KLIENT FELDESTÄRKE	ABSTAND ZUM AP AUS TABELLE 6.11
AP1	33,33	14,2m
AP2	35,33	12m
AP3	38,33	9m

Tabelle 6.12: Beispiel - Abstandsberechnung - Zwischenergebnis

6.9 Methode 6: Schnittpunkt zweier Kreise mit Mittelwerten

Das Programm berechnet die Abstände r_1 , r_2 und r_3 wie unter Abschnitt 6.5 Beschrieben. Zur Positionsermittlung kommt jedoch nicht die Trilateration zum Einsatz sonder der Algorithmus aus Abschnitt 4.1.6, siehe Listing 6.17

```

1 $x1 = 0;
2 $y1 = 0;
3 $x3 = 1680;
4 $y3 = 0;
5
6 $dx = $x3 - $x1;
7 $dy = $y3 - $y1;
8 $d = sqrt(($dx*$dx)+($dy*$dy));
9 $a = (( $r1*$r1 )-( $r3*$r3 )+($d*$d)) / (2*$d);
10 $h2 = ( $r1*$r1 )-( $a*$a);
11 $h = sqrt($h2);
12 $xs1 = $x1 + ( $a/$d)*$dx - ( $h/$d)*$dy;
13 $ys1 = $y1 + ( $a/$d)*$dy + ( $h/$d)*$dx;
14 $xs2 = $x1 + ( $a/$d)*$dx + ( $h/$d)*$dy;
15 $ys2 = $y1 + ( $a/$d)*$dy - ( $h/$d)*$dx;
16
17 $xap2 = 825;
18 $yap2 = 785;
19 $deltax1 = $xs1 - $xap2;
20 $deltay1 = $ys1 - $yap2;
21 $deltax2 = $xs2 - $xap2;
22 $deltay2 = $ys2 - $yap2;
23 $abstand1 = sqrt(($deltax1*$deltax1) + ($deltay1*$deltay1));
24 $abstand2 = sqrt(($deltax2*$deltax2) + ($deltay2*$deltay2));

```

```

25 $test1 = $abstand1 - $r2;
26 $test2 = $abstand2 - $r2;
27 if ($test1 < $test2)
28 {
29     print "abstand1 x:$xs1 y:$ys1\n";
30 }
31 elseif($test1 > $test2)
32 {
33     print "abstand2 x:$xs2 y:$ys2\n";
34 }
35 else
36 {
37     print "abstand1 = abstand2; Lösung=Abstand1 x:$xs1 y:$ys1\n";
38 }

```

Listing 6.17: Schnittpunkt zweier Kreise

In den Zeilen 1 bis 4 werden die Anfangsbedingungen festgelegt. Die Berechnung der Abstände r_1 bis r_3 ist in diesem Listing nicht zu sehen. Die Zeilen 6 bis 15 bestimmen die Schnittpunkte der Kreise r_1 und r_3 . Zur Auswahl des Schnittpunktes werden die Zeilen 17 bis 38 verwendet. Hierbei werden zuerst die Abstände der Schnittpunkte zu AP2 berechnet. Danach wird r_2 von diesen Abständen abgezogen und der kleiner der beiden Werte gesucht, dieser stellt den Aufenthaltsort des Klienten dar.

6.9.1 Vorgehensweise

Bei dieser Methode berechnet den Schnittpunkt zweier Kreise und wählt auf Grund eines Dritten Wertes den Aufenthaltsort des Klienten. Für die Berechnung sind die geschätzten Abstände des Klienten von jedem AP nötig. Die Berechnung erfolgt analog zur Methode „Trilateration mit Mittelwerten“. Die errechneten Abstände sind in Tabelle 6.13 nochmals zusammengefasst.

AP	REFERENZPUNKT	ABSTAND
1	13	1732,5cm = r1
2	3	1071,1cm = r2
3	6	346,6cm = r3

Tabelle 6.13: Beispiel - Schnittpunkt zweier Kreise - Abstände

Der Algorithmus berechnet nun den Schnittpunkt der Kreise r_1 und r_3 wie im Kapitel 4.1.6 beschrieben. Als Ergebnis hält man die Schnittpunkte P_{S1} 1697,6/346,2 und P_{S2} 1697,6/-346,2, siehe Abbildung 6.9.

Nachdem die Schnittpunkte P_{S1} und P_{S2} bekannt sind berechnet das Programm die Abstände a und b . Ist der Wert a kleiner als der Wert b fällt die Wahl auf den Schnittpunkt P_{S1} , umgekehrt fällt die Wahl auf den Schnittpunkt P_{S2} .

6.9.2 Ergebnisse

Die letzte Methode zur Positionsbestimmung gibt den Aufenthaltsort beim Punkt 1697,6/346,2cm an. Dieser Punkt ist 12,6m vom wirklichen Aufenthaltsort des Klienten entfernt.

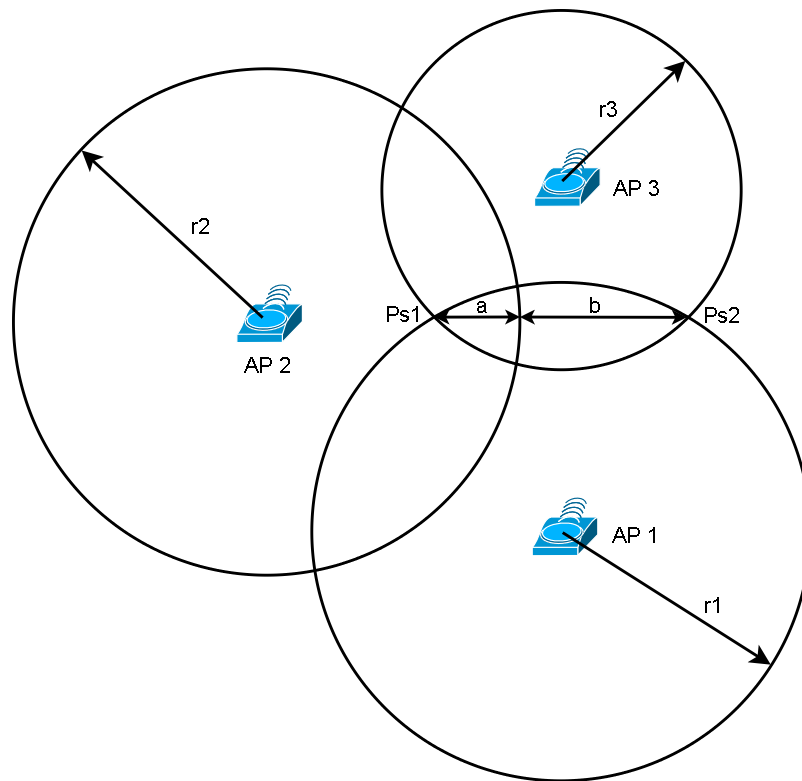


Abbildung 6.9: Schnittpunkt zweier Kreise

6.10 Visualisierung

Zur Visualisierung der Ergebnisse wird GnuPlot ¹ verwendet. Um GnuPlot in Perl verwenden zu können wird das Modul „Graphics-GnuplotIF“ in der Version 1.4 ² benötigt. Die Installation ist im Abschnitt B.13 beschrieben. Um das Modul in Perl nutzen zu können muss es geladen werden, dies geschieht innerhalb des Programms mit folgender Zeile.

```
1 use Graphics::GnuplotIF qw(GnuplotIF);
```

Listing 6.18: Perl GnuPlot Modul einbinden

Eine Beschreibung zur verwendung des Moduls findet man unter ³.

Der Quellcode des Programms ist im Abschnitt A.8 zu finden. Gestartet wird das Programm wie in Listing 6.19 gezeigt. Dabei müssen als Parameter der X und Y Abstand vom AP1 angegeben werden. Die Angabe muss in Zentimetern erfolgen.

```
1 #perl perlgplot.pl X Y
```

¹<http://www.gnuplot.info/>

²<http://search.cpan.org/~mehner/Graphics-GnuplotIF-1.4/>

³<http://search.cpan.org/~mehner/Graphics-GnuplotIF-1.4/lib/Graphics/GnuplotIF.pm>

Listing 6.19: Visualisierung starten

Die Ausgabe des Programms ist in Abbildung 6.10 zu sehen.

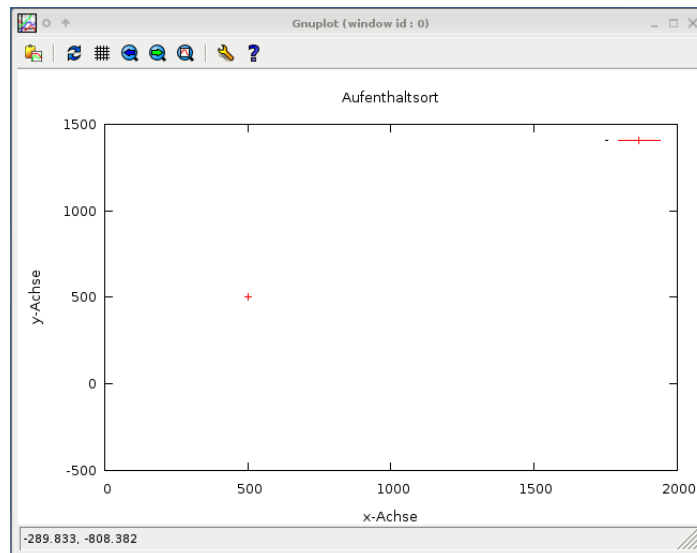


Abbildung 6.10: Gnuplot ausgabe

6.11 Ergebnisse

In der Tabelle 6.14 sind die Ergebnisse der einzelnen Methoden nochmals zusammengefasst. Eine Bewertung der Algorithmen anhand der Teststellung im Festsaal ist jedoch schwer vorzunehmen, da der Festsaal nicht die „elektromagnetischen“ Eigenschaften des Rathausplatzes besitzt.

METHODE	1	2	3	4	5	6
POSITIONSABWEICHUNG	8,4m	1,7m	1,4m	17,3m	4,6m	1,7m

Tabelle 6.14: Vergleich der Abweichungen

Kapitel 7

Ausblick

Aufbauend auf diese Arbeit sind folgende Weiterentwicklungen notwendig. Als Erstes sollte die Teststellung auf den Rathausplatz erweitert werden und die Genauigkeit optimiert werden.

Zu Verbesserung der Genauigkeit stehen zwei Grundlegende Verfahren zur Verfügung.

Sensoren Einer größere Anzahl an Sensoren (APs) sollte die Genauigkeit stark verbessern.

Kalibrieren Durch eine Kalibrierung wird das System Hardware unabhängig. Die Kalibrierungsmethoden können in zwei Klassen unterteilt werden. Die erste Klasse sind die Klient abhängigen Methoden die zweite Klasse die Klient unabhängigen. Bei Systemen der ersten Klasse muss der Klient seinen Standort dem LBS System vor Beginn der Positionierung bekannt geben. Messungen von Access Point in der Offline- und Onlinephase stellen die zweite Klasse an Kalibrierungsmethoden dar.

Anhang A

Quellcode

A.1 Sniffer

```
1  #!/usr/bin/perl -w
2  #
3  # Kismet Client by wess
4  # Daten werden vom Kismet Server gelesen, ueberprueft und in eine Datenbank geschrieben
5  # Daten sind MAC Adresse, Singalstaerke und IP Adresse
6  #     Beim Wlan darf keine Verschluesselung zum Einsatz kommen da die IP sonst nicht
7  #     ausgelesen werden kann da sie im Verschluesselten Payload des Layer2 Frames steht
8  # Für jeden Kismet Server muss eine eigene Instanz des Clients gestartet werden wobei
9  #     beim starten als Parameter die Postition und IP Adresse des AccessPoints (Sniffers)
    uebergeben
10 #     werden muss, die Position ist ein Integer, die IP eine Normale IP Adresse. Der Aufruf
    koennte so aussehen
11 #     ~perl sniffer.pl 1 172.16.1.254 (fuer den ersten AP mit der angegebenen IP)
12
13 # TODO Liste
14 # Konsolenparamter auf plausibilitaet pruefen
15
16 use IO::Select;
17 use IO::Socket;
18 use Socket;
19 use Fcntl;
20 use Carp;
21 use POSIX;
22 use warnings;
23 use DBI;
24
25 ###
26 # Konsolenparamter uebernehmen
27 ###
28
29 $ap = $ARGV[0];
30 $kismetip = $ARGV[1];
31
32 ###
33 # Verbindung mit dem Kismet Server
34 ###
35
36 $socket = IO::Socket::INET->new(PeerPort=>2501,PeerAddr=>$kismetip,Proto=>'tcp');
37 or croak ("Fehler beim Socket");
38
39 $select = IO::Select->new($socket) or croak ("Fehler beim select");
40
41 $flags = fcntl($socket,F_GETFL,0) or croak ("Fehler beim holen des Flags");
42 fcntl($socket,F_SETFL,$flags | O_NONBLOCK) or croak ("Fehler bei setzen von Non Blocking");
43
44 $protocol = 'TIME';
45 $socket->send(sprintf("!0 REMOVE %s\n",$protocol),0) or croak ("fehler bei remove");
46
47 $protocol = 'CLIENT';
48 $fields = 'mac,signal,ip';
49 $socket->send(sprintf("!1 ENABLE %s %s\n",$protocol, $fields),0) or croak ("fehler enable");
50
51 ###
52 # Verbindung mit der Datenbank
53 ###
54
55 $dbname="lbs";
56 $username="lbs";
57 $password="lbs";
```

```

58 $dbh = DBI->connect("dbi:Pg:database=$dbname;host=127.0.0.1;port=5432","$username","$password") or
    die "scheisse";
59
60 ###
61 # In einer Schleife vom Socket lesen und die gelesenen Daten auf Plausibilität prüfen
62 # Bei der Überprüfung der Parameter MAC, Signal, und IP müssen für die aktuelle Umgebung
63 # neue Regex gefunden werden, in den Zeilen 75, 85 und 95!
64 ###
65
66 while (!select(undef,undef,undef,0.25))
67 {
68     foreach($select->can_read(0))
69     {
70         $rv = $_->recv($data, POSIX::BUFSIZ,0);
71         die "Kann nicht vom Server lesen" unless defined($rv);
72         ($Protokoll, $MAC, $Signal, $IP) = split(/\ /,$data);
73
74         if ($MAC =~ /00:02:78:85:17:76/)
75         {
76             print "MAC gefunden \n";
77         }
78         else
79         {
80             print "KEINE MAC \n";
81             next;
82         }
83
84         if ($Signal =~ /^[0-9]{1,2}$/)
85         {
86             print "Signal gefunden \n";
87         }
88         else
89         {
90             print "KEIN Signal \n";
91             next;
92         }
93
94         if ($IP =~ /192\.168\.1\.\d/)
95         {
96             print "IP gefunden \n";
97         }
98         else
99         {
100             print "KEINE IP \n";
101             next;
102         }
103         print "$MAC, $Signal, $IP\n";
104     }
105 # SELECT (prüfen ob es den Client schon gibt)
106 ###
107 if (@row = $dbh->selectrow_array("SELECT pos FROM sniffer WHERE mac='$MAC' and ap='$ap'"))
108 {
109     ##
110     # UPDATE (Ist die Client schon vorhanden -> Werte updaten)
111     # Es werden pro Client und AP 3 Signalwerte aufgenommen bzw. ständig ueberschrieben
112     ###
113     $posneu = 0;
114     $zahlselect = $row[0];
115     if ($zahlselect <= 0 )
116     {
117         print "Error 1\n";
118     }
119     elsif ($zahlselect < 3)
120     {
121         $posneu = $zahlselect+1;
122     }
123     elsif ($zahlselect == 3)
124     {
125         $posneu = 1;
126     }
127     else
128     {
129         print "Error 2\n";
130     }
131
132     $sqlupdate = qq/UPDATE sniffer SET "signal$zahlselect"='$Signal', ip='$IP', pos='$posneu'
        WHERE mac = '$MAC' and ap='$ap'/;
133     $zahlupdate = $dbh->do($sqlupdate) or warn $dbh->errstr;
134     print "Updated: $zahlupdate\n";
135 }
136 else
137 {
138     ###
139     # INSERT (Gibt es den Client (MAC) noch nicht -> anlegen; alle 3 Spalten bekommen den selben
        Wert)
140     ###
141     $sqlinsert = qq/INSERT INTO sniffer (mac,ip,signal1,signal2,signal3,ap,pos) VALUES ('$MAC'
        , '$IP', '$Signal', '$Signal', '$Signal', '$ap',1)/;
142     $zahlinsert = $dbh->do($sqlinsert) or warn $dbh->errstr;
143     print "Inserted: $zahlinsert\n";

```

```

144 }
145
146     }
147 }
148 print "ENDE";
149 $dbh->disconnect();
150 exit (0);

```

Listing A.1: sniffer.pl

A.2 Methode 1: Differenzmethode mit Mittelwerten

```

1  #!/usr/bin/perl -w
2
3  # Positionsbestimmung des am nächsten gelegenen Referenzpunktes by wess
4  # Das Programm liest die in der Offlinephase gewonnen Daten aller APs (Sniffer)
5  # bei den jeweiligen Referenzpunkte aus der Datenbank und vergleicht sie mit
6  # den aktuellen Daten des Clients. Die Feldstärken der Referenzpunkte muessen gemittelt sein.
7  # Der Messpunkt mit der geringsten Differenz zum den Werten den Clients ist
8  # der geschaeetzte Standort
9
10 use IO::Select;
11 use IO::Socket;
12 use Socket;
13 use Fcntl;
14 use Carp;
15 use POSIX;
16 use warnings;
17 use DBI;
18
19 ###
20 # Verbindung mit der Datenbank herstellen
21 ###
22
23 $dbname="lbs ";
24 $username="lbs ";
25 $password="lbs ";
26 $dbh = DBI->connect("dbi:Pg:database=$dbname;host=127.0.0.1;port=5432","$username","$password") or
    die "datenbank connection error";
27
28 ###
29 # Referenzpunkte aus Datenbank auslesen
30 # In den Arrays @ap1 bis @ap3 stehen die gemessen Feldstärken bei den einzelnen
31 # Positionen. Array Index 0 ist gleich erster Messpunkt, Array Index 1 ist
32 # gleich zweiter Messpunkt usw
33 ###
34
35 $query = "SELECT ap1,ap2,ap3 FROM data";
36 $sth = $dbh->prepare($query);
37 $sth->execute();
38 $sth->bind_columns(undef,\$ap1data,\$ap2data,\$ap3data);
39 while($sth->fetch())
40 {
41     push (@ap1,$ap1data);
42     push (@ap2,$ap2data);
43     push (@ap3,$ap3data);
44 }
45
46 ###
47 # Aktuelle Feldstärke des Clients aus der Datenbank auslesen
48 # In den Variablen clap1 bis clap3 stehen die Messwerte des gesuchten Clients
49 ##
50
51 $query = "SELECT signal1,signal2,signal3,ap FROM sniffer";
52 $sth = $dbh->prepare($query);
53 $sth->execute();
54 $sth->bind_columns(undef,\$signal1,\$signal2,\$signal3,\$sniffer);
55 while ($sth->fetch())
56 {
57     if ($sniffer == 1)
58     {
59         $clap1 = ($signal1 + $signal2 + $signal3) / 3;
60     }
61     if ($sniffer == 2)
62     {
63         $clap2 = ($signal1 + $signal2 + $signal3) / 3;
64     }
65     if ($sniffer == 3)
66     {
67         $clap3 = ($signal1 + $signal2 + $signal3) / 3;
68     }
69 }
70

```

```

71 ###
72 # Arrays mit Diff Werten anlegen
73 ###
74
75 foreach (@ap1)
76 {
77     $diff1 = $clap1 - $_;
78     push (@diffap1, $diff1);
79 }
80
81 foreach (@ap2)
82 {
83     $diff2 = $clap2 - $_;
84     push (@diffap2, $diff2);
85 }
86
87 foreach (@ap3)
88 {
89     $diff3 = $clap3 - $_;
90     push (@diffap3, $diff3);
91 }
92
93 ###
94 # Summe Diff Array anlegen
95 ###
96
97 for ($i=0; $i<13; $i++)
98 {
99     $summe = $diffap1[$i] + $diffap2[$i] + $diffap3[$i];
100     push (@summediff, $summe);
101 }
102
103 ###
104 # Im Summe Diff Array die Position des Elements das am naechsten bei Null liegt
105 # suchen
106 ###
107
108 $x = 0; #x - haelt die aktuelle position im array, wird bei jeden durchlauf
           erhoeht
109 $akt = $summediff[0]; #akt - haelt die aktuell kleinste zahl
110 $pos = 0; #pos - haelt die position der kleinsten Zahl
111
112 foreach (@summediff)
113 {
114     if (abs($akt) < abs($_))
115     {
116         $x++;
117     }
118     elsif (abs($akt) > abs($_))
119     {
120         $akt = $_;
121         $pos = $x;
122         $x++;
123     }
124     else
125     {
126         $x++;
127     }
128 }
129 $pos++; # Plus 1 da der Index bei Null beginnt die Referenzpunkte jedoch bei 1
130 print "Referenzpunkt: $pos \n";
131 exit(0);

```

Listing A.2: auswertung.pl

A.3 Methode 2: Trilateration mit Mittelwerten

```

1  #!/usr/bin/perl -w
2
3  # Positionsbestimmung mittels Trilateration by wess
4  # Das Programm liest die in der Offline Phase gewonnen Daten aller APs ein
5  # und vergleicht die aktuelle Feldstärke von Sniffer 1 mit dem Wert von AP1
6  # bzw Sniffer 2 mit dem Wert von AP2 und Sniffer3 mit dem Wert von AP3
7  # diese 3 Vergleiche ergeben 3 Referenzpunkte, die Referenzpunkte werden
8  # in Entfernungen (also AP zu Referenzpunkt) umgerechnet so bekommt man
9  # 3 Entfernungswerte, diese 3 Entfernungswerte kann man bei der Trilateration
10 # benutzen
11
12 use IO::Select;
13 use IO::Socket;
14 use Socket;
15 use Fcntl;
16 use Carp;

```

```

17 use POSIX;
18 use warnings;
19 use DBI;
20
21 ###
22 # Verbindung mti der Datenbank herstellen
23 ###
24
25 $dbname="lbs ";
26 $username="lbs ";
27 $password="lbs ";
28 $dbh = DBI->connect("dbi:Pg:database=$dbname;host=127.0.0.1;port=5432","$username","$password") or
    die "datenbank connection error";
29
30 ###
31 # Referenzpunkte aus der Datenbank auslesen
32 # In den Arrays @ap1 bis @ap3 stehen die gemessen Feldstärken bei den einzelnen
33 # Positionen. Array Index 0 ist gleich erster Messpunkt, Array Index 1 ist
34 # gleich zweiter Messpunkt usw
35 # In den Variablen clap1 bis clap3 stehen die Messwerte des gesuchten Clients
36 ###
37
38 $query = "SELECT ap1,ap2,ap3 FROM data";
39 $sth = $dbh->prepare($query);
40 $sth->execute();
41 $sth->bind_columns(undef,\$ap1data,\$ap2data,\$ap3data);
42 while($sth->fetch())
43 {
44     push (@ap1,$ap1data);
45     push (@ap2,$ap2data);
46     push (@ap3,$ap3data);
47 }
48
49
50 ###
51 # Aktuelle Feldstärke des Clients aus der Datenbank auslesen
52 ###
53
54 $query = "SELECT signal1,signal2,signal3,ap FROM sniffer";
55 $sth = $dbh->prepare($query);
56 $sth->execute();
57 $sth->bind_columns(undef,\$signal1,\$signal2,\$signal3,\$sniffer);
58 while ($sth->fetch())
59 {
60     if ($sniffer == 2)
61     {
62         $clap1 = ($signal1 + $signal2 + $signal3) / 3;
63     }
64     if ($sniffer == 3)
65     {
66         $clap2 = ($signal1 + $signal2 + $signal3) / 3;
67     }
68     if ($sniffer == 1)
69     {
70         $clap3 = ($signal1 + $signal2 + $signal3) / 3;
71     }
72 }
73
74
75 ###
76 # Diff AP1
77 ###
78
79 foreach(@ap1)
80 {
81     $diff1=$clap1 - $_;
82     #print "diff1: $diff1\n";
83     push (@diffap1,$diff1)
84 }
85
86 ###
87 # Diff AP2
88 ###
89 foreach(@ap2)
90 {
91     $diff2=$clap2 - $_;
92     #print "diff2: $diff2\n";
93     push (@diffap2,$diff2)
94 }
95
96 ###
97 # Diff AP3
98 ###
99 foreach(@ap3)
100 {
101     $diff3=$clap3 - $_;
102     #print "diff3: $diff3\n";
103     push (@diffap3,$diff3)
104 }
105

```

```

106
107
108 ###
109 # Abstand von AP1
110 ###
111
112 $x1 = 0;
113 $akt1 = $diffap1[0];
114 $pos1 = 0;
115
116 foreach (@diffap1)
117 {
118     if (abs($akt1) < abs($_))
119     {
120         $x1++;
121     }
122     elsif (abs($akt1) > abs($_))
123     {
124         $akt1 = $_;
125         $pos1 = $x1;
126         $x1++;
127     }
128     else
129     {
130         $x1++;
131         # sollte die kleinste zahl und die aktuelle gleich sein tue nichts
132     }
133 }
134
135 ###
136 # Abstand von AP2
137 ###
138 $x2 = 0;
139 $akt2 = $diffap2[0];
140 $pos2 = 0;
141
142 foreach (@diffap2)
143 {
144     if (abs($akt2) < abs($_))
145     {
146         $x2++;
147     }
148     elsif (abs($akt2) > abs($_))
149     {
150         $akt2 = $_;
151         $pos2 = $x2;
152         $x2++;
153     }
154     else
155     {
156         $x2++;
157         # sollte die kleinste zahl und die aktuelle gleich sein tue nichts
158     }
159 }
160
161 ###
162 # Abstand von AP3
163 ###
164 $x3 = 0;
165 $akt3 = $diffap3[0];
166 $pos3 = 0;
167
168 foreach (@diffap3)
169 {
170     if (abs($akt3) < abs($_))
171     {
172         $x3++;
173     }
174     elsif (abs($akt3) > abs($_))
175     {
176         $akt3 = $_;
177         $pos3 = $x3;
178         $x3++;
179     }
180     else
181     {
182         $x3++;
183         # sollte die kleinste zahl und die aktuelle gleich sein tue nichts
184     }
185 }
186 print "position ap1: $pos1\n";
187 print "position ap2: $pos2\n";
188 print "position ap3: $pos3\n";
189
190 ###
191 # Arrays die die Entfernung des APs zum Messpunkt angeben
192 # Um die Formel der Trilateration anwenden zu koennen wird das System
193 # transformiert, wie folgt
194 # ----- AP1 wird zu AP1
195 # ----- AP2 wird zu AP3

```



```

196 # ----- AP3 wird zu AP2
197 # Die Transformation bezieht sich auf die Array @aplmeter bis @ap3meter und
198 # die Werte des Klienten $clap1 bis clap3
199 # ORIGINAL WERTE
200 # @aplmeter =
      (1533.3,965.2,246.6,397.1,1009.6,1561.6,1693.1,1203,765.3,1409.2,782.9,1276.2,1732.5);
201 # @ap2meter = (1187.7,941.4,1071.1,744.5,541.6,904.3,751.7,201.6,549.2,1428,1337.5,351.1,611.2);
202 # @ap3meter =
      (180.3,726.9,1438.5,1468.6,784.9,346.6,740.4,1021.8,1607.7,617.6,1187.9,1557.8,1181.9);
203 ###
204
205 ###
206 # Abstände zu den APs bevor Transformiert wird
207 ###
208
209 @aplmeter = (1533.3,965.2,246.6,397.1,1009.6,1561.6,1693.1,1203,765.3,1409.2,782.9,1276.2,1732.5);
210 @ap2meter = (1187.7,941.4,1071.1,744.5,541.6,904.3,751.7,201.6,549.2,1428,1337.5,351.1,611.2);
211 @ap3meter = (180.3,726.9,1438.5,1468.6,784.9,346.6,740.4,1021.8,1607.7,617.6,1187.9,1557.8,1181.9)
      ;
212
213 $r1 = $aplmeter[$spos1];
214 $r2 = $ap2meter[$spos2];
215 $r3 = $ap3meter[$spos3];
216
217 print "Entfernung von apl $r1 von ap2: $r2 von ap3: $r3; \n";
218
219 ###
220 # Transformatierten
221 # r1 => R1
222 # r2 => R3
223 # r3 => R2
224 ###
225
226 $R1 = $r1;
227 $R2 = $r3;
228 $R3 = $r2;
229
230 ###
231 # Trilateration aus sicht von API
232 ###
233
234 $d = 1730;
235 $i = 785;
236 $j = 825;
237 $x = ((($R1*$R1)-($R2*$R2)+($d*$d)) / (2*$d));
238 $y = (((($R1*$R1)-($R3*$R3)+($i*$i)+($j*$j)) / (2*$j))-((($i/$j)*$x));
239 print "x:$x y:$y\n";
240
241 exit(0);

```

Listing A.3: auswertung1.pl

A.4 Methode 3: Trilateration mit Normalverteilung

```

1  #!/usr/bin/perl -w
2
3
4  # Positionsbestimmung mittels Normalverteilung by wess
5  # Das Programm berechnet die Wahrscheinlichkeit das ein Messwert bei einem bestimmten
      Referenzpunkt liegt
6  # der Messpunkt mit der höchsten Wahrscheinlichkeit wird ausgewählt
7  # Man bekommt also 3 Messpunkte bei denen der Klient stehen könnte
8
9  use IO::Select;
10 use IO::Socket;
11 use Socket;
12 use Fcntl;
13 use Carp;
14 use POSIX;
15 use warnings;
16 use DBI;
17
18 ###
19 # Verbindung mit der Datenbank herstellen
20 ###
21
22 $dbname="lbs ";
23 $username="lbs ";
24 $password="lbs ";
25 $dbh = DBI->connect("dbi:Pg:database=$dbname;host=127.0.0.1;port=5432","$username","$password") or
      die "datenbank connection error";
26
27 ###

```

```

28 # Referenzpunkte aus der Datenbank auslesen
29 # Es wird immer der Erwartungswert und die Standardabweichung ausgelesen
30 # und für jeden AP ein eigenes Array gespeichert, dabei ist der Arrayindex gleich dem
    Referenzpunktindex
31 # ap1e, ap2e und ap3e speichern die Erwartungswerte pro AP
32 # ap1s, ap2s und ap3s speichern die Standardabweichung pro AP
33 ###
34
35 $query = "SELECT ap1e,ap2e,ap3e,ap1s,ap2s,ap3s FROM normalverteilung";
36 $sth = $dbh->prepare($query);
37 $sth->execute();
38 $sth->bind_columns(undef,\$ap1e,\$ap2e,\$ap3e,\$ap1s,\$ap2s,\$ap3s);
39 while($sth->fetch())
40 {
41     push (@ap1e,$ap1e);
42     push (@ap2e,$ap2e);
43     push (@ap3e,$ap3e);
44     push (@ap1s,$ap1s);
45     push (@ap2s,$ap2s);
46     push (@ap3s,$ap3s);
47 }
48
49 ###
50 # Aktuelle Feldstärke des Clients aus der Datenbank auslesen
51 # In den Variablen clap1 bis clap3 stehen die Messwerte des gesuchten Clients
52 ##
53
54 $query = "SELECT signal1,signal2,signal3,ap FROM sniffer";
55 $sth = $dbh->prepare($query);
56 $sth->execute();
57 $sth->bind_columns(undef,\$signal1,\$signal2,\$signal3,\$sniffer);
58 while ($sth->fetch())
59 {
60     if ($sniffer == 1)
61     {
62         $clap1 = ($signal1 + $signal2 + $signal3) / 3;
63     }
64     if ($sniffer == 2)
65     {
66         $clap2 = ($signal1 + $signal2 + $signal3) / 3;
67     }
68     if ($sniffer == 3)
69     {
70         $clap3 = ($signal1 + $signal2 + $signal3) / 3;
71     }
72 }
73
74
75 ###
76 # In einer Schleife für jeden Messpunkt Wahrscheinlichkeit ausrechnen das RSSI dort liegt
77 # und die Position merken bei der die Wahrscheinlichkeit am höchsten ist.
78 # Das muss für jeden AP wiederholt werden.
79 # pp: hält die aktuell höchste Wahrscheinlichkeit
80 # position: hält den index der höchsten Wahrscheinlichkeit pp – also den Referenzpunkt
81 # pneu: hält die nächste Wahrscheinlichkeit welche mit pp verglichen wird – pneu wird mithilfe
82 #       der normalverteilung in der funktion rechne bestimmt
83 # ACHTUNG: Der Wert "i" in der Schleife muss je nach Anzahl der Messpunkte angepasst werden.
84 #
85 # Suche des Messpunktes des AP1
86 ###
87 $pp1 = 0.0;
88 $pneu1 = 0.0;
89 $position1 = 0;
90 for ($i=0;$i<13;$i++)
91 {
92     $pneu1 = &rechne($clap1,$ap1e[$i],$ap1s[$i]);
93     if ($pp1 > $pneu1)
94     {
95         #pp größer als pneu -> tue nichts
96     }
97     elsif ($pp1 < $pneu1)
98     {
99         $pp1 = $pneu1;
100        $position1 = $i;
101    }
102    else
103    {
104        #pp = pneu -> tue nichts
105    }
106 }
107 ###
108 # Suche des Messpunktes des AP2
109 ###
110 $pp2 = 0.0;
111 $pneu2 = 0.0;
112 $position2 = 0;
113 for ($i=0;$i<13;$i++)
114 {
115     $pneu2 = &rechne($clap2,$ap2e[$i],$ap2s[$i]);
116     if ($pp2 > $pneu2)

```

```

117     {
118         #pp größer als pneu -> tue nichts
119     }
120     elsif ($pp2 < $pneu2)
121     {
122         $pp2 = $pneu2;
123         $position2 = $i;
124     }
125     else
126     {
127         #pp = pneu -> tue nichts
128     }
129 }
130 ###
131 # Suche des Messpunktes des AP3
132 ###
133 $pp3 = 0.0;
134 $pneu3 = 0.0;
135 $position3 = 0;
136 for ($i=0;$i<13;$i++)
137 {
138     $pneu3 = &rechne($clap3,$ap3e[$i],$ap3s[$i]);
139     if ($pp3 > $pneu3)
140     {
141         #pp größer als pneu -> tue nichts
142     }
143     elsif ($pp3 < $pneu3)
144     {
145         $pp3 = $pneu3;
146         $position3 = $i;
147     }
148     else
149     {
150         #pp = pneu -> tue nichts
151     }
152 }
153
154 print "position ap1: $position1\n";
155 print "position ap2: $position2\n";
156 print "position ap3: $position3\n";
157
158 ###
159 # Trilateration mit den Messpunkten durchführen
160 ###
161
162 ###
163 # Abstände zu den APs bevor Transformatiert wird
164 ###
165
166 @ap1meter = (1533.3,965.2,246.6,397.1,1009.6,1561.6,1693.1,1203,765.3,1409.2,782.9,1276.2,1732.5);
167 @ap2meter = (1187.7,941.4,1071.1,744.5,541.6,904.3,751.7,201.6,549.2,1428,1337.5,351.1,611.2);
168 @ap3meter = (180.3,726.9,1438.5,1468.6,784.9,346.6,740.4,1021.8,1607.7,617.6,1187.9,1557.8,1181.9)
169 ;
170 $r1 = $ap1meter[$position1];
171 $r2 = $ap2meter[$position2];
172 $r3 = $ap3meter[$position3];
173
174 print "Entfernung von ap1 $r1 von ap2: $r2 von ap3: $r3; \n";
175
176 ###
177 # Transformatierten
178 # r1 => R1
179 # r2 => R3
180 # r3 => R2
181 ###
182
183 $R1 = $r1;
184 $R2 = $r3;
185 $R3 = $r2;
186
187 ###
188 # Trilateration
189 ###
190
191 $d = 1730;
192 $i = 785;
193 $j = 825;
194 $x = ((($R1*$R1)-($R2*$R2)+($d*$d)) / (2*$d));
195 $y = (((($R1*$R1)-($R3*$R3)+($i*$i)+($j*$j)) / (2*$j))-((($i/$j)*$x));
196 print "x:$x y:$y\n";
197
198 exit(0);
199
200 ###
201 # Funktion
202 # Berechnet die Wahrscheinlichkeit das Rssi an Position i liegt
203 ###
204 sub rechne
205 {

```

```

206 $negativ = 0;
207 $column = 0;
208
209 @["0.0"] = (0.5000, 0.5040, 0.5080, 0.5120, 0.5160, 0.5199, 0.5239, 0.5279, 0.5319, 0.5359);
210 @["0.1"] = (0.5398, 0.5438, 0.5478, 0.5517, 0.5557, 0.5596, 0.5636, 0.5675, 0.5714, 0.5753);
211 @["0.2"] = (0.5793, 0.5832, 0.5871, 0.5910, 0.5948, 0.5987, 0.6026, 0.6064, 0.6103, 0.6141);
212 @["0.3"] = (0.6179, 0.6217, 0.6255, 0.6293, 0.6331, 0.6368, 0.6406, 0.6443, 0.6480, 0.6517);
213 @["0.4"] = (0.6554, 0.6591, 0.6628, 0.6664, 0.6700, 0.6736, 0.6772, 0.6808, 0.6844, 0.6879);
214 @["0.5"] = (0.6915, 0.6950, 0.6985, 0.7019, 0.7054, 0.7088, 0.7123, 0.7157, 0.7190, 0.7224);
215 @["0.6"] = (0.7257, 0.7291, 0.7324, 0.7357, 0.7389, 0.7422, 0.7454, 0.7486, 0.7517, 0.7549);
216 @["0.7"] = (0.7580, 0.7611, 0.7642, 0.7673, 0.7704, 0.7734, 0.7764, 0.7794, 0.7823, 0.7852);
217 @["0.8"] = (0.7881, 0.7910, 0.7939, 0.7967, 0.7995, 0.8023, 0.8051, 0.8078, 0.8106, 0.8133);
218 @["0.9"] = (0.8159, 0.8186, 0.8212, 0.8238, 0.8264, 0.8289, 0.8315, 0.8340, 0.8365, 0.8389);
219 @["1.0"] = (0.8413, 0.8438, 0.8461, 0.8485, 0.8508, 0.8531, 0.8554, 0.8577, 0.8599, 0.8621);
220 @["1.1"] = (0.8643, 0.8665, 0.8686, 0.8708, 0.8729, 0.8749, 0.8770, 0.8790, 0.8810, 0.8830);
221 @["1.2"] = (0.8849, 0.8869, 0.8888, 0.8907, 0.8925, 0.8944, 0.8962, 0.8980, 0.8997, 0.9015);
222 @["1.3"] = (0.9032, 0.9049, 0.9066, 0.9082, 0.9099, 0.9115, 0.9131, 0.9147, 0.9162, 0.9177);
223 @["1.4"] = (0.9192, 0.9207, 0.9222, 0.9236, 0.9251, 0.9265, 0.9279, 0.9292, 0.9306, 0.9319);
224 @["1.5"] = (0.9332, 0.9345, 0.9357, 0.9370, 0.9382, 0.9394, 0.9406, 0.9418, 0.9429, 0.9441);
225 @["1.6"] = (0.9452, 0.9463, 0.9474, 0.9484, 0.9495, 0.9505, 0.9515, 0.9525, 0.9535, 0.9545);
226 @["1.7"] = (0.9554, 0.9564, 0.9573, 0.9582, 0.9591, 0.9599, 0.9608, 0.9616, 0.9625, 0.9633);
227 @["1.8"] = (0.9641, 0.9649, 0.9656, 0.9664, 0.9671, 0.9678, 0.9686, 0.9693, 0.9699, 0.9706);
228 @["1.9"] = (0.9713, 0.9719, 0.9726, 0.9732, 0.9738, 0.9744, 0.9750, 0.9756, 0.9761, 0.9767);
229 @["2.0"] = (0.9772, 0.9778, 0.9783, 0.9788, 0.9793, 0.9798, 0.9803, 0.9808, 0.9812, 0.9817);
230 @["2.1"] = (0.9821, 0.9826, 0.9830, 0.9834, 0.9838, 0.9842, 0.9846, 0.9850, 0.9854, 0.9857);
231 @["2.2"] = (0.9861, 0.9864, 0.9868, 0.9871, 0.9875, 0.9878, 0.9881, 0.9884, 0.9887, 0.9890);
232 @["2.3"] = (0.9893, 0.9896, 0.9898, 0.9901, 0.9904, 0.9906, 0.9909, 0.9911, 0.9913, 0.9916);
233 @["2.4"] = (0.9918, 0.9920, 0.9922, 0.9925, 0.9927, 0.9929, 0.9931, 0.9932, 0.9934, 0.9936);
234 @["2.5"] = (0.9938, 0.9940, 0.9941, 0.9943, 0.9945, 0.9946, 0.9948, 0.9949, 0.9951, 0.9952);
235 @["2.6"] = (0.9953, 0.9955, 0.9956, 0.9957, 0.9959, 0.9960, 0.9961, 0.9962, 0.9963, 0.9964);
236 @["2.7"] = (0.9965, 0.9966, 0.9967, 0.9968, 0.9969, 0.9970, 0.9971, 0.9972, 0.9973, 0.9974);
237 @["2.8"] = (0.9974, 0.9975, 0.9976, 0.9977, 0.9977, 0.9978, 0.9979, 0.9979, 0.9980, 0.9981);
238 @["2.9"] = (0.9981, 0.9982, 0.9982, 0.9983, 0.9984, 0.9984, 0.9985, 0.9985, 0.9986, 0.9986);
239 @["3.0"] = (0.9987, 0.9987, 0.9987, 0.9988, 0.9988, 0.9989, 0.9989, 0.9989, 0.9990, 0.9990);
240 @["3.1"] = (0.9990, 0.9991, 0.9991, 0.9991, 0.9992, 0.9992, 0.9992, 0.9992, 0.9993, 0.9993);
241 @["3.2"] = (0.9993, 0.9993, 0.9994, 0.9994, 0.9994, 0.9994, 0.9994, 0.9995, 0.9995, 0.9995);
242 @["3.3"] = (0.9995, 0.9995, 0.9995, 0.9996, 0.9996, 0.9996, 0.9996, 0.9996, 0.9996, 0.9997);
243 @["3.4"] = (0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9998);
244
245 $x = $_[0];
246 $e = $_[1];
247 $v = $_[2];
248
249 #print "x1: $x\n";
250 #print "e1: $e\n";
251 #print "v1: $v\n";
252
253 $u = (($x+0.5) - $e)/$v;
254 #print "u1: $u und ";
255 if ($u < 0.0)
256 {
257     $u *= -1;
258     $negativ = 1;
259 }
260 if ($u > 3.5)
261 {
262     $u = 3.39;
263 }
264
265 $row = sprintf("%.1f", $u);
266
267 $column = substr($u, 3, 1);
268 #print "row1 = $row und column1 = $column\n";
269
270 if ($negativ == 0)
271 {
272     $p1 = ${"$row"}[$column];
273 }
274
275 else
276 {
277     $p1 = 1 - ${"$row"}[$column];
278     $negativ = 0;
279 }
280
281 ##### zweiten teil berechnen
282 #print "x2: $x\n";
283 #print "e2: $e\n";
284 #print "v2: $v\n";
285
286
287 $u = (($x-0.5) - $e)/$v;
288 #print "u2: $u\n";
289 if ($u < 0.0)
290 {
291     $u *= -1;
292     $negativ = 1;
293 }
294 if ($u > 3.5)
295 {

```

```

296         $u = 3.39;
297     }
298
299     $row = sprintf("%.1f", $u);
300     $column = substr($u, 3, 1);
301     #print "row2 = $row und column2 = $column\n";
302     if ($negativ == 0)
303     {
304         $p2 = ${"$row"}[$column];
305     }
306     else
307     {
308         $p2 = 1 - ${"$row"}[$column];
309         $negativ = 0;
310     }
311
312     #print "p1: $p1\n";
313     #print "ps: $p2\n";
314     $p = ($p1 - $p2);
315     #print "2: $p\n";
316     return $p;
317 }

```

Listing A.4: auswertung2.pl

A.5 Methode 4: Durchschnittsposition mit Normalverteilung

```

1  #!/usr/bin/perl -w
2
3
4  # Positionsbestimmung mittels Normalverteilung und Durchschnittsposition by wess
5  # Das Programm berechnet die Wahrscheinlichkeit das ein Messwert bei einem bestimmten
   Referenzpunkt liegt
6  # der Messpunkt mit der höchsten Wahrscheinlichkeit wird ausgewählt
7  # Man bekommt also 3 Messpunkte bei denen der Klient stehen könnte
8  # Bei Der Durchschnittsposition werden die X und Y Werte der 3 Messpunkte aufsummiert und durch 3
   geteilt.
9
10 use IO::Select;
11 use IO::Socket;
12 use Socket;
13 use Fcntl;
14 use Carp;
15 use POSIX;
16 use warnings;
17 use DBI;
18
19 ###
20 # Verbindung mit der Datenbank herstellen
21 ###
22
23 $dbname="lbs ";
24 $username="lbs ";
25 $password="lbs ";
26 $dbh = DBI->connect("dbi:Pg; database=$dbname; host=127.0.0.1; port=5432", "$username", "$password") or
   die "datenbank connection error";
27
28 ###
29 # Referenzpunkte aus der Datenbank auslesen
30 # Es wird immer der Erwartungswert und die Standardabweichung ausgelesen
31 # und für jeden AP ein eigenem Array gespeichert, dabei ist der Arrayindex gleich dem
   Referenzpunktindex
32 # ap1e, ap2e und ap3e speichern die Erwartungswerte pro AP
33 # ap1s, ap2s und ap3s speichern die Standardabweichung pro AP
34 ###
35
36 $query = "SELECT ap1e, ap2e, ap3e, ap1s, ap2s, ap3s FROM normalverteilung";
37 $sth = $dbh->prepare($query);
38 $sth->execute();
39 $sth->bind_columns(undef, \ $ap1e, \ $ap2e, \ $ap3e, \ $ap1s, \ $ap2s, \ $ap3s);
40 while ($sth->fetch())
41 {
42     push (@ap1e, $ap1e);
43     push (@ap2e, $ap2e);
44     push (@ap3e, $ap3e);
45     push (@ap1s, $ap1s);
46     push (@ap2s, $ap2s);
47     push (@ap3s, $ap3s);
48 }
49
50 ###
51 # Aktuelle Feldstärke des Clients aus der Datenbank auslesen
52 # In den Variablen clap1 bis clap3 stehen die Messwerte des gesuchten Clients

```

```

53 ##
54
55 $query = "SELECT signal1,signal2,signal3,ap FROM sniffer";
56 $sth = $dbh->prepare($query);
57 $sth->execute();
58 $sth->bind_columns(undef,\ $signal1,\ $signal2,\ $signal3,\ $sniffer);
59 while ($sth->fetch())
60 {
61     if ($sniffer == 1)
62     {
63         $clap1 = ($signal1 + $signal2 + $signal3) / 3;
64     }
65     if ($sniffer == 2)
66     {
67         $clap2 = ($signal1 + $signal2 + $signal3) / 3;
68     }
69     if ($sniffer == 3)
70     {
71         $clap3 = ($signal1 + $signal2 + $signal3) / 3;
72     }
73 }
74
75 ###
76 # In einer Schleife für jeden Messpunkt Wahrscheinlichkeit ausrechnen das RSSI dort liegt
77 # und die Position merken bei der die Wahrscheinlichkeit am höchsten ist.
78 # Das muss für jeden AP wiederholt werden.
79 # pp: hält die aktuell höchste Wahrscheinlichkeit
80 # position: hält den index der höchsten Wahrscheinlichkeit pp – also den Referenzpunkt
81 # pneu: hält die nächste Wahrscheinlichkeit welche mit pp verglichen wird – pneu wird mithilfe
82 #       der normalverteilung in der funktion rechne bestimmt
83 # ACHTUNG: Der Wert "i" in der Schleife muss je nach Anzahl der Messpunkte angepasst werden.
84 #
85 # Suche des Messpunktes des AP1
86 ###
87 $pp1 = 0.0;
88 $pneu1 = 0.0;
89 $position1 = 0;
90 for ($i=0;$i<13;$i++)
91 {
92     $pneu1 = &rechne($clap1,$ap1e[$i],$ap1s[$i]);
93     if ($pp1 > $pneu1)
94     {
95         #pp größer als pneu -> tue nichts
96     }
97     elsif ($pp1 < $pneu1)
98     {
99         $pp1 = $pneu1;
100        $position1 = $i;
101    }
102    else
103    {
104        #pp = pneu -> tue nichts
105    }
106 }
107 ###
108 # Suche des Messpunktes des AP2
109 ###
110 $pp2 = 0.0;
111 $pneu2 = 0.0;
112 $position2 = 0;
113 for ($i=0;$i<13;$i++)
114 {
115     $pneu2 = &rechne($clap2,$ap2e[$i],$ap2s[$i]);
116     if ($pp2 > $pneu2)
117     {
118         #pp größer als pneu -> tue nichts
119     }
120     elsif ($pp2 < $pneu2)
121     {
122         $pp2 = $pneu2;
123         $position2 = $i;
124     }
125     else
126     {
127         #pp = pneu -> tue nichts
128     }
129 }
130 ###
131 # Suche des Messpunktes des AP3
132 ###
133 $pp3 = 0.0;
134 $pneu3 = 0.0;
135 $position3 = 0;
136 for ($i=0;$i<13;$i++)
137 {
138     $pneu3 = &rechne($clap3,$ap3e[$i],$ap3s[$i]);
139     if ($pp3 > $pneu3)
140     {
141         #pp größer als pneu -> tue nichts
142     }

```

```

143         elsif ($pp3 < $pneu3)
144         {
145             $pp3 = $pneu3;
146             $position3 = $i;
147         }
148         else
149         {
150             #pp = pneu -> tue nichts
151         }
152     }
153
154     print "position ap1: $position1\n";
155     print "position ap2: $position2\n";
156     print "position ap3: $position3\n";
157
158     ###
159     # Durchschnittsposition berechnen
160     # Durchschnittsposition = X und Y Werte der 3 Referenzpunkte aufsummieren und durch 3 teilen.
161     # Die X und Y Abstände werden aus Sicht des API berechnet.
162     # Die Abstände aus Sicht des API stehen in den Arrays @aplx und @aply.
163     # Positiv X ist von API aus zur Bühne
164     # Positiv Y ist von API aus zur Mensa
165     ###
166
167     @aplx = (-100, -100, -100, 312.5, 312.5, 312.5, 725, 725, 725, -500, -500, 1125, 1125);
168     @aply = (1530, 960, 245, 245, 960, 1530, 1530, 960, 245, 1317.5, 602.5, 602.5, 1317.5);
169
170     $x = ($aplx[$position1] + $aplx[$position2] + $aplx[$position3]) / 3;
171     $y = ($aply[$position1] + $aply[$position2] + $aply[$position3]) / 3;
172     print "x:$x y:$y\n";
173
174     exit(0);
175
176     ###
177     # Funktion
178     # Berechnet die Wahrscheinlichkeit das Rssi an Position i liegt
179     ###
180     sub rechne
181     {
182         $negativ = 0;
183         $column = 0;
184
185         @{"0.0"} = (0.5000, 0.5040, 0.5080, 0.5120, 0.5160, 0.5199, 0.5239, 0.5279, 0.5319, 0.5359);
186         @{"0.1"} = (0.5398, 0.5438, 0.5478, 0.5517, 0.5557, 0.5596, 0.5636, 0.5675, 0.5714, 0.5753);
187         @{"0.2"} = (0.5793, 0.5832, 0.5871, 0.5910, 0.5948, 0.5987, 0.6026, 0.6064, 0.6103, 0.6141);
188         @{"0.3"} = (0.6179, 0.6217, 0.6255, 0.6293, 0.6331, 0.6368, 0.6406, 0.6443, 0.6480, 0.6517);
189         @{"0.4"} = (0.6554, 0.6591, 0.6628, 0.6664, 0.6700, 0.6736, 0.6772, 0.6808, 0.6844, 0.6879);
190         @{"0.5"} = (0.6915, 0.6950, 0.6985, 0.7019, 0.7054, 0.7088, 0.7123, 0.7157, 0.7190, 0.7224);
191         @{"0.6"} = (0.7257, 0.7291, 0.7324, 0.7357, 0.7389, 0.7422, 0.7454, 0.7486, 0.7517, 0.7549);
192         @{"0.7"} = (0.7580, 0.7611, 0.7642, 0.7673, 0.7704, 0.7734, 0.7764, 0.7794, 0.7823, 0.7852);
193         @{"0.8"} = (0.7881, 0.7910, 0.7939, 0.7967, 0.7995, 0.8023, 0.8051, 0.8078, 0.8106, 0.8133);
194         @{"0.9"} = (0.8159, 0.8186, 0.8212, 0.8238, 0.8264, 0.8289, 0.8315, 0.8340, 0.8365, 0.8389);
195         @{"1.0"} = (0.8413, 0.8438, 0.8461, 0.8485, 0.8508, 0.8531, 0.8554, 0.8577, 0.8599, 0.8621);
196         @{"1.1"} = (0.8643, 0.8665, 0.8686, 0.8708, 0.8729, 0.8749, 0.8770, 0.8790, 0.8810, 0.8830);
197         @{"1.2"} = (0.8849, 0.8869, 0.8888, 0.8907, 0.8925, 0.8944, 0.8962, 0.8980, 0.8997, 0.9015);
198         @{"1.3"} = (0.9032, 0.9049, 0.9066, 0.9082, 0.9099, 0.9115, 0.9131, 0.9147, 0.9162, 0.9177);
199         @{"1.4"} = (0.9192, 0.9207, 0.9222, 0.9236, 0.9251, 0.9265, 0.9279, 0.9292, 0.9306, 0.9319);
200         @{"1.5"} = (0.9332, 0.9345, 0.9357, 0.9370, 0.9382, 0.9394, 0.9406, 0.9418, 0.9429, 0.9441);
201         @{"1.6"} = (0.9452, 0.9463, 0.9474, 0.9484, 0.9495, 0.9505, 0.9515, 0.9525, 0.9535, 0.9545);
202         @{"1.7"} = (0.9554, 0.9564, 0.9573, 0.9582, 0.9591, 0.9599, 0.9608, 0.9616, 0.9625, 0.9633);
203         @{"1.8"} = (0.9641, 0.9649, 0.9656, 0.9664, 0.9671, 0.9678, 0.9686, 0.9693, 0.9699, 0.9706);
204         @{"1.9"} = (0.9713, 0.9719, 0.9726, 0.9732, 0.9738, 0.9744, 0.9750, 0.9756, 0.9761, 0.9767);
205         @{"2.0"} = (0.9772, 0.9778, 0.9783, 0.9788, 0.9793, 0.9798, 0.9803, 0.9808, 0.9812, 0.9817);
206         @{"2.1"} = (0.9821, 0.9826, 0.9830, 0.9834, 0.9838, 0.9842, 0.9846, 0.9850, 0.9854, 0.9857);
207         @{"2.2"} = (0.9861, 0.9864, 0.9868, 0.9871, 0.9875, 0.9878, 0.9881, 0.9884, 0.9887, 0.9890);
208         @{"2.3"} = (0.9893, 0.9896, 0.9898, 0.9901, 0.9904, 0.9906, 0.9909, 0.9911, 0.9913, 0.9916);
209         @{"2.4"} = (0.9918, 0.9920, 0.9922, 0.9925, 0.9927, 0.9929, 0.9931, 0.9932, 0.9934, 0.9936);
210         @{"2.5"} = (0.9938, 0.9940, 0.9941, 0.9943, 0.9945, 0.9946, 0.9948, 0.9949, 0.9951, 0.9952);
211         @{"2.6"} = (0.9953, 0.9955, 0.9956, 0.9957, 0.9959, 0.9960, 0.9961, 0.9962, 0.9963, 0.9964);
212         @{"2.7"} = (0.9965, 0.9966, 0.9967, 0.9968, 0.9969, 0.9970, 0.9971, 0.9972, 0.9973, 0.9974);
213         @{"2.8"} = (0.9974, 0.9975, 0.9976, 0.9977, 0.9977, 0.9978, 0.9979, 0.9979, 0.9980, 0.9981);
214         @{"2.9"} = (0.9981, 0.9982, 0.9982, 0.9983, 0.9984, 0.9984, 0.9985, 0.9985, 0.9986, 0.9986);
215         @{"3.0"} = (0.9987, 0.9987, 0.9987, 0.9988, 0.9988, 0.9988, 0.9989, 0.9989, 0.9990, 0.9990);
216         @{"3.1"} = (0.9990, 0.9991, 0.9991, 0.9991, 0.9992, 0.9992, 0.9992, 0.9992, 0.9993, 0.9993);
217         @{"3.2"} = (0.9993, 0.9993, 0.9994, 0.9994, 0.9994, 0.9994, 0.9995, 0.9995, 0.9995, 0.9995);
218         @{"3.3"} = (0.9995, 0.9995, 0.9995, 0.9996, 0.9996, 0.9996, 0.9996, 0.9996, 0.9996, 0.9997);
219         @{"3.4"} = (0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9997, 0.9998);
220
221         $x = $_[0];
222         $y = $_[1];
223         $v = $_[2];
224         $u = (($x+0.5) - $y)/$v;
225         if ($u < 0.0)
226         {
227             $u *= -1;
228             $negativ = 1;
229         }
230         if ($u > 3.5)
231         {
232             $u = 3.39;

```

```

233 }
234 $row = sprintf("%.1f", $u);
235 $column = substr($u, 3, 1);
236 if ($negativ == 0)
237 {
238     $p1 = ${"$row"}[$column];
239 }
240 else
241 {
242     $p1 = 1 - ${"$row"}[$column];
243     $negativ = 0;
244 }
245
246 ##### zweiten teil berechnen
247 #print "x2: $x\n";
248 #print "e2: $e\n";
249 #print "v2: $v\n";
250
251 $u = (($x - 0.5) - $e) / $v;
252 if ($u < 0.0)
253 {
254     $u *= -1;
255     $negativ = 1;
256 }
257 if ($u > 3.5)
258 {
259     $u = 3.39;
260 }
261
262 $row = sprintf("%.1f", $u);
263 $column = substr($u, 3, 1);
264 if ($negativ == 0)
265 {
266     $p2 = ${"$row"}[$column];
267 }
268 else
269 {
270     $p2 = 1 - ${"$row"}[$column];
271     $negativ = 0;
272 }
273 $p = ($p1 - $p2);
274 return $p;
275 }

```

Listing A.5: auswertung3.pl

A.6 Methode 5: Trilateration mit Abstandsberechnung

```

1  #!/usr/bin/perl -w
2
3
4  # Positionsbestimmung mittels Abstandsberechnung und Trilateration
5
6  use IO::Select;
7  use IO::Socket;
8  use Socket;
9  use Fcntl;
10 use Carp;
11 use POSIX;
12 use warnings;
13 use DBI;
14
15 ###
16 # Verbindung mit der Datenbank herstellen
17 ###
18
19 $dbname="lbs ";
20 $username="lbs ";
21 $password="lbs ";
22 $dbh = DBI->connect("dbi:Pg:database=$dbname;host=127.0.0.1;port=5432","$username","$password") or
    die "datenbank connection error";
23
24
25 ###
26 # Aktuelle Feldstärke des Clients aus der Datenbank auslesen
27 # In den Variablen clap1 bis clap3 stehen die Messwerte des gesuchten Clients
28 ##
29
30 $query = "SELECT signal1 , signal2 , signal3 , ap FROM sniffer";
31 $sth = $dbh->prepare($query);
32 $sth->execute();
33 $sth->bind_columns(undef, \ $signal1 , \ $signal2 , \ $signal3 , \ $sniffer);
34 while ($sth->fetch())

```



```

35 {
36     if ($sniffer == 1)
37     {
38         $clap1 = ($signal1 + $signal2 + $signal3) / 3;
39         $clap1 = sprintf("%.2f", $clap1);
40         #print "feldstärke1: $clap1\n";
41     }
42     if ($sniffer == 2)
43     {
44         $clap2 = ($signal1 + $signal2 + $signal3) / 3;
45         $clap2 = sprintf("%.2f", $clap2);
46         #print "feldstärke2: $clap2\n";
47     }
48     if ($sniffer == 3)
49     {
50         $clap3 = ($signal1 + $signal2 + $signal3) / 3;
51         $clap3 = sprintf("%.2f", $clap3);
52         #print "feldstärke3: $clap3\n";
53     }
54 }
55 ###
56 # Abstandsrechnung durchführen, Formel dazu wurde mit Matlab und linearer Regression bestimmt
57 # Dabei wird der Abstand von jedem AP zum Klient berechnet.
58 # Für jeden Klient ist die Formel anders, hier ist die Formel für den Acer N50 PDA!!
59 ###
60
61 $a0 = 44.0588;
62 $a1 = 32.4025;
63 $a2 = -0.0998;
64 $a3 = 0.0033;
65 $a4 = -19.5541;
66
67 ### Array anlegen in dem die dB Werte in 10cm Abständen gespeichert werden.
68 ### Die dB Werte werden bis 20m berechnet. i=20!!
69 for ($i=0; $i <= 20; $i = $i+0.1)
70 {
71     $apcm = $a0 + $a1*exp(-$i) + $a2*($i*$i) + $a3*($i*$i*$i) + $a4*$i*exp(-$i);
72     push (@apabstand, $apcm);
73 }
74
75
76
77 ### Abstand von AP1
78
79 $pos1 = 0;
80 for ($ap1=0; $ap1<=180; $ap1++)
81 {
82     $ap1plus = $ap1 + 1;
83     if ($clap1 < $apabstand[$ap1] && $clap1 > $apabstand[$ap1plus])
84     {
85         #print "ap1 gefunden $apabstand[$ap1] und $apabstand[$ap1plus] — index[$ap1]\n";
86         $pos1 = $ap1;
87     }
88 }
89
90 ### Abstand von AP2
91
92 $pos2 = 0;
93 for ($ap2=0; $ap2<=180; $ap2++)
94 {
95     $ap2plus = $ap2 + 1;
96     if ($clap2 < $apabstand[$ap2] && $clap2 > $apabstand[$ap2plus])
97     {
98         #print "ap2 gefunden $apabstand[$ap2] und $apabstand[$ap2plus] — index [$ap2]\n";
99         $pos2 = $ap2;
100     }
101 }
102
103 ### Abstand von AP3
104
105 $pos3 = 0;
106 for ($ap3=0; $ap3<=180; $ap3++)
107 {
108     $ap3plus = $ap3 + 1;
109     if ($clap3 < $apabstand[$ap3] && $clap3 > $apabstand[$ap3plus])
110     {
111         #print "ap3 gefunden $apabstand[$ap3] und $apabstand[$ap3plus] — index [$ap3]\n";
112         $pos3 = $ap3;
113     }
114 }
115
116 ###
117 # Trilateration mit den Messpunkten durchführen
118 ###
119
120 ###
121 # Abstände zu den APs bevor Transformiert wird
122 ###
123
124 $r1 = $pos1 * 10;

```

```

125 $r2 = $pos2 * 10;
126 $r3 = $pos3 * 10;
127
128
129
130 ###
131 # Transformierten
132 # r1 => R1
133 # r2 => R3
134 # r3 => R2
135 ###
136
137 $R1 = $r1;
138 $R2 = $r3;
139 $R3 = $r2;
140
141 ###
142 # Trilateration
143 ###
144
145 $d = 1730;
146 $i = 785;
147 $j = 825;
148 $x = ((($R1*$R1)-($R2*$R2)+($d*$d)) / (2*$d));
149 $y = (((($R1*$R1)-($R3*$R3)+($i*$i)+($j*$j)) / (2*$j))-((($i/$j)*$x));
150 print "x:$x y:$y\n";
151
152 exit(0);

```

Listing A.6: auswertung4.pl

A.7 Methode 6: Schnittpunkt zweier Kreise mit Mittelwerten

```

1  #!/usr/bin/perl -w
2
3  # Positionsbestimmung mit Durchschnittswert, Schnittpunkt zweier Kreise und auswahl des Punktes
4  # durch den dritten kreis by wess
5
6
7  use IO::Select;
8  use IO::Socket;
9  use Socket;
10 use Fcntl;
11 use Carp;
12 use POSIX;
13 use warnings;
14 use DBI;
15
16 ###
17 # Verbindung mti der Datenbank herstellen
18 ###
19
20 $dbname="lbs";
21 $username="lbs";
22 $password="lbs";
23 $dbh = DBI->connect("dbi:Pg:database=$dbname;host=127.0.0.1;port=5432","$username","$password") or
    die "datenbank connection error";
24
25 ###
26 # Referenzpunkte aus der Datenbank auslesen
27 # In den Arrays @ap1 bis @ap3 stehen die gemessen Feldstärken bei den einzelnen
28 # Positionen. Array Index 0 ist gleich erster Messpunkt, Array Index 1 ist
29 # gleich zweiter Messpunkt usw
30 #
31 # In den Variablen clap1 bis clap3 stehen die Messwerte des gesuchten Clients
32 ###
33
34 $query = "SELECT ap1,ap2,ap3 FROM data";
35 $sth = $dbh->prepare($query);
36 $sth->execute();
37 $sth->bind_columns(undef,\$ap1data,\$ap2data,\$ap3data);
38 while($sth->fetch())
39 {
40     push (@ap1,$ap1data);
41     push (@ap2,$ap2data);
42     push (@ap3,$ap3data);
43 }
44
45 ###
46 # Aktuelle Feldstärke des Clients aus der Datenbank auslesen
47 ###

```

```

48
49 $query = "SELECT signal1 , signal2 , signal3 , ap FROM sniffer ";
50 $sth = $dbh->prepare($query);
51 $sth->execute();
52 $sth->bind_columns(undef,\ $signal1 ,\ $signal2 ,\ $signal3 ,\ $sniffer);
53 while ( $sth->fetch() )
54 {
55     if ( $sniffer == 2 )
56     {
57         $clap1 = ( $signal1 + $signal2 + $signal3 ) / 3;
58     }
59     if ( $sniffer == 3 )
60     {
61         $clap2 = ( $signal1 + $signal2 + $signal3 ) / 3;
62     }
63     if ( $sniffer == 1 )
64     {
65         $clap3 = ( $signal1 + $signal2 + $signal3 ) / 3;
66     }
67 }
68 }
69
70 ###
71 # Diff AP1
72 ###
73
74 foreach (@ap1)
75 {
76     $diff1=$clap1 - $_;
77     push (@diffap1 , $diff1)
78 }
79
80 ###
81 # Diff AP2
82 ###
83
84 foreach (@ap2)
85 {
86     $diff2=$clap2 - $_;
87     push (@diffap2 , $diff2)
88 }
89
90 ###
91 # Diff AP3
92 ###
93
94 foreach (@ap3)
95 {
96     $diff3=$clap3 - $_;
97     push (@diffap3 , $diff3)
98 }
99
100 ###
101 # Abstand von AP1
102 ###
103
104 $x1 = 0;
105 $akt1 = $diffap1[0];
106 $pos1 = 0;
107
108 foreach (@diffap1)
109 {
110     if (abs($akt1) < abs($_))
111     {
112         $x1++;
113     }
114     elsif (abs($akt1) > abs($_))
115     {
116         $akt1 = $_;
117         $pos1 = $x1;
118         $x1++;
119     }
120     else
121     {
122         $x1++;
123         # sollte die kleinste zahl und die aktuelle gleich sein tue nichts
124     }
125 }
126
127 ###
128 # Abstand von AP2
129 ###
130 $x2 = 0;
131 $akt2 = $diffap2[0];
132 $pos2 = 0;
133
134 foreach (@diffap2)
135 {
136     if (abs($akt2) < abs($_))
137     {

```

```

138         $x2++;
139     }
140     elseif(abs($akt2) > abs($_))
141     {
142         $akt2 = $_;
143         $pos2 = $x2;
144         $x2++;
145     }
146     else
147     {
148         $x2++;
149         # sollte die kleinste zahl und die aktuelle gleich sein tue nichts
150     }
151 }
152
153 ###
154 # Abstand von AP3
155 ###
156 $x3 = 0;
157 $akt3 = $diffap3[0];
158 $pos3 = 0;
159
160 foreach(@diffap3)
161 {
162     if(abs($akt3) < abs($_))
163     {
164         $x3++;
165     }
166     elseif(abs($akt3) > abs($_))
167     {
168         $akt3 = $_;
169         $pos3 = $x3;
170         $x3++;
171     }
172     else
173     {
174         $x3++;
175         # sollte die kleinste zahl und die aktuelle gleich sein tue nichts
176     }
177 }
178
179 print "position ap1: $pos1\n";
180 print "position ap2: $pos2\n";
181 print "position ap3: $pos3\n";
182
183 ###
184 # Abstände zu den APs
185 ###
186
187 @ap1meter = (1533.3,965.2,246.6,397.1,1009.6,1561.6,1693.1,1203,765.3,1409.2,782.9,1276.2,1732.5);
188 @ap2meter = (1187.7,941.4,1071.1,744.5,541.6,904.3,751.7,201.6,549.2,1428,1337.5,351.1,611.2);
189 @ap3meter = (180.3,726.9,1438.5,1468.6,784.9,346.6,740.4,1021.8,1607.7,617.6,1187.9,1557.8,1181.9)
190 ;
191 $r1 = $ap1meter[$pos1];
192 $r2 = $ap2meter[$pos2];
193 $r3 = $ap3meter[$pos3];
194
195 print "r1: $r1; r3:$r3\n";
196 ###
197 # Schnittpunkt von den Kreisen r1 und r3 berechnen
198 ###
199
200 $x1 = 0;
201 $y1 = 0;
202 $x3 = 1680;
203 $y3 = 0;
204
205 $dx = $x3 - $x1;
206 $dy = $y3 - $y1;
207 $d = sqrt(($dx*$dx)+($dy*$dy));
208 $a = ((($r1*$r1)-($r3*$r3)+($d*$d)) / (2*$d));
209 $h2 = (($r1*$r1)-($a*$a));
210 $h = sqrt($h2);
211 $xs1 = $x1 + ($a/$d)*$dx - ($h/$d)*$dy;
212 $ys1 = $y1 + ($a/$d)*$dy + ($h/$d)*$dx;
213 $xs2 = $x1 + ($a/$d)*$dx + ($h/$d)*$dy;
214 $ys2 = $y1 + ($a/$d)*$dy - ($h/$d)*$dx;
215
216 $xap2 = 825;
217 $yap2 = 785;
218 $deltax1 = $xs1 - $xap2;
219 $deltay1 = $ys1 - $yap2;
220 $deltax2 = $xs2 - $xap2;
221 $deltay2 = $ys2 - $yap2;
222 $abstand1 = sqrt(($deltax1*$deltax1) + ($deltay1*$deltay1));
223 $abstand2 = sqrt(($deltax2*$deltax2) + ($deltay2*$deltay2));
224
225 $test1 = $abstand1 - $r2;
226 $test2 = $abstand2 - $r2;

```

```

227
228 if ($test1 < $test2)
229 {
230     print "abstand1 x:$xs1 y:$ys1\n";
231 }
232 elsif($test1 > $test2)
233 {
234     print "abstand2 x:$xs2 y:$ys2\n";
235 }
236 else
237 {
238     print "abstand1 = abstand2; Lösung=Abstand1 x:$xs1 y:$ys1\n";
239 }
240
241 exit(0);

```

Listing A.7: auswerutng5.pl

A.8 GnuPlot

```

1  #!/usr/bin/perl -w
2
3  #Visualisierungsprogramm by wess
4  # Das Programm gibt einen Punkt grafisch aus
5  # Es ben tigt eine Gnuplot installation sowie das Perl Modul GnuplotIF
6  # Beim Aufruf muss dem Programm der X und Y Wert   bergeben werden
7  # BEISPIEL: perl perlgnuplot.pl 100 100  zeichnet net Punkt 100 / 100 auf
8  # die Karte
9
10 use POSIX;
11 use warnings;
12 use Graphics::GnuplotIF qw(GnuplotIF);
13
14 my $plot1 = Graphics::GnuplotIF->new(persist => 1, style =>"linespoints");
15 $x1 = $ARGV[0];
16 $y1 = $ARGV[1];
17 @x = ($x1);
18 @y = ($y1);
19
20 $plot1->gnuplot_set_title(" Aufenthaltsort");
21 $plot1->gnuplot_set_xrange(0,2000);
22 $plot1->gnuplot_set_yrange(-500,1500);
23 $plot1->gnuplot_set_xlabel("x-Achse");
24 $plot1->gnuplot_set_ylabel("y-Achse");
25
26 $plot1->gnuplot_plot_xy(\@x, \@y);
27 exit(0);

```

Listing A.8: perlgnuplot.pl

Anhang B

Installationsanleitung

B.1 Grundsystem

Vor dem Start der Installation muss in der Datei „/etc/apt/sources.list“ jede Zeile die mit CDROM beginnt auskommentiert werden.

```
1 #aptitude install openssh-server
```

Listing B.1: SSH Installation

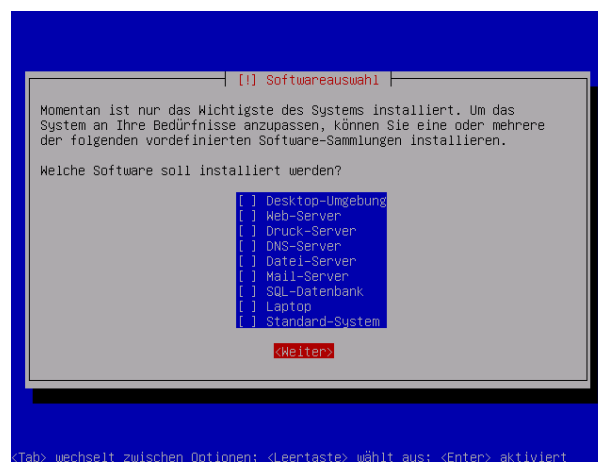


Abbildung B.1: Debian Installation - Softwareauswahl

B.2 Datenbank

```
1 #aptitude install postgresql-8.1
```

Listing B.2: PostgreSQL Installation

```
1 #psql -h localhost -U lbs -d lbs
```

Listing B.3: PostgreSQL administrieren

Die Parameter haben folgende Bedeutung:

- -h gibt den Host an
- -U gibt den Benutzer an
- -d gibt die Datenbank an

Das grafische Administrationstool pgadmin3 kann man wie folgt auf einem Linux System installieren.

```
1 #aptitude install pgadmin3
```

Listing B.4: PostgreSQL pgadmin3

Unter Windows gibt es ein Installationsdatei unter ¹.

B.3 Perl

```
1 #aptitude install libclass-dbi-perl libclass-dbi-pg-perl
```

Listing B.5: Perl Datenbankmodule

B.4 Madwifi

```
1 #wget http://downloads.sourceforge.net/madwifi/madwifi-0.9.4.tar.gz
2 #tar xvfz madwifi-0.9.4.tar.gz
3 #cd madwifi-0.9.4
4 #aptitude install make build-essential automake linux-headers-2.6-686 wireless-tools
5 #make
6 #make install
7 #modprobe ath_pci
```

Listing B.6: Madwifi Installation

B.5 Kismet

```
1 #aptitude install kismet
```

Listing B.7: Kismet Installation

B.6 Lesen der Feldstärke

```
1 use IO::Select;
2 use IO::Socket;
3 use Socket;
4 use Fcntl;
5 use Carp;
6 use POSIX;
7 use warnings;
8 use DBI;
```

Listing B.8: Sniffer Module laden

¹<http://www.postgresql.org/ftp/pgadmin3/release/v1.8.2/win32/>

```

1 $dbname="lbs ";
2 $username="lbs ";
3 $password="lbs ";
4 $dbh = DBI->connect("dbi:Pg:database=$dbname;host=127.0.0.1;port=5432","$username","$password")
5     or die "Fehler bei der Datenbankverbindung";

```

Listing B.9: Verbindung mit der Datenbank

B.7 DHCP

```

1 aptitude install dhcp3-server

```

Listing B.10: DHCP Installation

B.8 PIX

Hier werden zuerst die entsprechenden Access Listen angelegt, wobei darauf zu achten ist den Eintrag remote networkrichtig zu setzten. Die Zeilen 7-11 definieren die Phase 2. Des weiteren definieren die Zeilen 14-20 die Phase 1 von IPSEC. Danach muss IPSEC noch aktiviert werden indem es, wie in Zeile 13 zu sehen, an das Outside Interface gebunden wird.

```

1 access-list 101 extended permit ip any "remote network"
2 access-list nicht_naten extended permit ip any "remote network"
3 access-list nicht_naten extended permit ip "remote network" any
4 !
5 nat (inside) 0 access-list nicht_naten
6 !
7 crypto ipsec transform-set toyota esp-des esp-md5-hmac
8 crypto map bmw 1 match address 101
9 crypto map bmw 1 set peer "remote address"
10 crypto map bmw 1 set transform-set toyota
11 crypto map bmw 1 set security-association lifetime seconds 86400
12 !
13 crypto isakmp enable outside
14 isakmp key ***** address "remote address" netmask 255.255.255.0
15 crypto isakmp policy 1
16 authentication pre-share
17 encryption des
18 hash md5
19 group 1
20 lifetime 86400

```

Listing B.11: PIX 515 IPSEC Konfiguration

B.9 Kismet Server starten

```

1 #!/bin/sh
2
3 # Kismet startup by wess
4 # Dieses Skript zerstört zuerst vorhande Wlan Interfaces (ath0 bzw ath1)
5 # und legt ein neues Wlan Interfaces im Monitor Modus an
6 # Danach faehrt es das Interface hoch und startet den Kismet Server
7
8 wlanconfig ath0 destroy
9 wlanconfig ath1 destroy
10 wlanconfig ath0 create wlandev wifi0 wlanmode monitor
11 ifconfig ath0 up
12 kismet_server

```

Listing B.12: Kismet Server Starten

B.10 GnuPlot

```
1 #tar xvzf Graphics-GnuplotIF-1.4.tar.gz
2 #perl Makefile.PL
3 #make
4 #make test
5 #make install
```

Listing B.13: Perl GnuPlot Modul installieren

Abkürzungsverzeichnis

AoA	Angel of Arrival
AP	Access Point
ATM	Asynchronous Transfer Mode
BSS	basic service set
BSSID	Basic Service Set Identifier
CCA	clear channel assessment function
CoO	Cell of Origin
CRC	Cyclic Redundancy Code
DSSS	Direct Sequence Spread Spectrum
EIRP	Equivalent isotropically radiated power
FCS	frame check sequence
FHSS	Frequency Hopping Spread Spectrum
GMSK	Gaussian Minimum Shift Keying
HiperLAN	High Performance Radio Local Area Network
HomeRF	Home Radio Frequency
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISP	Internet Service Provider
LBS	Location-based services
MPDU	MAC Protocol Data Unit
PDA	Personal Digital Assistant
PLCP	Physical Layer Convergence Procedure
PMD	Physical Medium Dependent
PPP	Point-to-Point Protocol
PSP	PlayStation Portable
QoS	Quality of Service
RTS	request to send
SCSI	Small Computer System Interface
SFD	Start Frame Delimiter

SNR	Signal-Rauschabstand
TDoA	Time Differenz of Arrival
ToA	Time of Arrival
WAF	Wall Attenuation Factor
WDS	Wireless Distribution System
WiMax	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network

Literaturverzeichnis

- [AL05] Sunny Consolvo Anthony LaMarca, Yatin Chawathe, *Place lab: Device positioning using radio beacons in the wild*, <http://www.placelab.org/publications/pubs/pervasive-placelab-2005-final.pdf>, 2005.
- [Alk04] Eyad Alkassar, *Positionsbestimmung durch wlan*, <http://w5.cs.uni-sb.de/~stahl/proseminar-lbs/pdf/Radar.pdf>, 2004.
- [Ber04] Humboldt Universität Berlin, *Magic map - system*, http://www2.informatik.hu-berlin.de/rok/MagicMap/_images/screenshots/client_20-12-2004_3.png, 2004.
- [Ber07] ———, *Magic map - verfahren*, http://wiki.informatik.hu-berlin.de/nomads/index.php/MagicMap_Verfahren, 2007.
- [Com07] Computerpartner, *Integration von voice over ip in wireless lans*, <http://www.computerpartner.at/articles/20071127064443002.html>, 2007.
- [gft05] gftp, *gftp v2.0.18*, <http://gftp.seul.org/>, 2005.
- [Glo05] Thomas Gloor, *Ortsbestimmung mit placelab*, www.vs.inf.ethz.ch/edu/SS2005/DS/reports/08.1-localisation-report.pdf, 2005.
- [Jeh07] Gunnar Jehl, *Deterministic wlan position determination*, http://www.ks.uni-freiburg.de/download/papers/lbsSS07/DeterministicPosition/Gunnar%20Jehl_Deterministic%20WLAN%20Position%20Determination.pdf, 2007.
- [Kap07] Reiko Kaps, *Ipv6 für kleine netze*, <http://www.heise.de/netze/artikel/98759/0>, 2007.

- [Kow08] Martin Kowalski, *Schnittpunkte zweier kreise*, <http://kowalski.s4f.eu/addt/2kreise.html>, 2008.
- [Neu05] Andreas Neubacher, *Luftschnittstelle - air interface*, <http://ecampus.fh-stpoelten.ac.at/mod/resource/view.php?id=1468>, 2005.
- [Neu06] ———, *Das cdma zugriffsverfahren*, http://ecampus.fh-stpoelten.ac.at/file.php/974/060522_CDMA_Air_Interface_UMTS_2_Neubacher/UMTS_CDMA_Air-Interface_2_NEUBACHER.pdf, 2006.
- [Pap01] Lothar Papula, *Mathematik für ingenieure und naturwissenschaftler band 3*, Vieweg Verlagsgesellschaft mbH, 2001.
- [PAT04] Gene Mosca Paul A. Tipler, *Physik für wissenschaftler und ingenieure 2. auflage*, Elsevier, Spektrum Akademischer Verlag, 2004.
- [PB00] Venkata N. Padmanabhan Paramvir Bahl, *Radar: An in-building rf-based user location and tracking system*, <http://research.microsoft.com/~padmanab/papers/infocom2000.pdf>, 2000.
- [PI04] Martin Schweigert Peter Ibach, Tobias Hübner, *Magicmap kooperative positionsbestimmung über wlan*, www.ccc.de/congress/2004/fahrplan/files/48-ad-hoc-collaboration-paper.pdf, 2004.
- [Rec06] Jörg Rech, *Wireless lans*, Heise, 2006.
- [Sar03] Damir Sarac, *Projekt lokationsinformationen*, <http://www.sarac.de/frameset/content/CV/master%20program/projects/lokationinformation/Lokationsinfo.htm>, 2003.
- [Sch07] Alexander Schätzle, *Mobile and location based service*, Lehrstule für Kommunikationssysteme, Albert-Ludwigs-Universität Freiburg, 2007.
- [Soc03] IEEE Computer Society, *Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications (r2003)*, <http://ieeexplore.ieee.org>, 2003.
- [Soc07] ———, *Draft standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements. part 11: Wireless lan medi-*

- um access control (mac) and physical layer (phy) specifications*, <http://ieeexplore.ieee.org>, 2007.
- [SS06] Moritz Neun und Alistair Edwardes Stefan Steiniger, *Foundations of location based services*, www.geo.unizh.ch/publications/cartouche/lbs_lecturenotes_steinigeretal2006.pdf, 2006.
- [Sza07] Artur Szankowski, *Location based services*, HTL St.Pölten, 2007.
- [Tri06] Javier Triviño, *Ieee 802.11 long reach link*, http://www.wilac.net/descargas/documentos/EnlaceAguila_Baul_EN.pdf, 2006.
- [Wik08] Wikipedia, *Trilateration*, <http://en.wikipedia.org/wiki/Trilateration>, 2008.
- [W.K06] Krzysztof W.Kolodziej, *Local positioning systems*, CRC Press Inc, 2006.
- [xli07] xlightftpd, *Xlight ftp server v2.8*, <http://www.xlightftpd.com/>, 2007.
- [Zür05] Universität Zürich, *Elektromagnetische welle*, <http://www.physik.unizh.ch/teaching/physik-b/WS0506/woche12/Slide4.png>, 2005.