



Otto-von-Guericke-Universität Magdeburg

Fakultät für Informatik

Institut für Verteilte Systeme

Diplomarbeit

Evaluierungsumgebung für Lokalisierungsverfahren in drahtlosen Netzen

Hagen Steinicke

30. August 2006

Betreuer:

Prof. Dr. Edgar Nett

Universität Magdeburg

Fakultät für Informatik

Institut für Verteilte Systeme

Arbeitsgruppe Echtzeitsysteme und Kommunikation

M. Sc. Svilen Ivanov

Universität Magdeburg

Fakultät für Informatik

Institut für Verteilte Systeme

Arbeitsgruppe Echtzeitsysteme und Kommunikation

Kurzfassung

Das Gebiet der Ortung von Nutzern in drahtlosen Netzen wird stetig durch neue Lokalisierungsverfahren erweitert. Die Evaluierung dieser Verfahren ist wichtig, um die Genauigkeit der Lokalisierung zu ermitteln. Es existieren jedoch keine verbreiteten Standards, wie die Evaluierung eines Lokalisierungsverfahrens durchgeführt werden muss. Dadurch sind die Ergebnisse der Evaluierungen nicht vergleichbar. Diese Tatsache motiviert das Ziel dieser Arbeit, eine Möglichkeit zu finden, mit der man die Leistungsfähigkeit unterschiedlicher Verfahren testen und bewerten kann.

Um ein Verfahren evaluieren zu können, müssen alle nötigen Eingangsdaten vorhanden und alle erforderlichen Features implementiert sein, die der Algorithmus für die Lokalisierung benötigt. Aus diesem Grund wurden mehrere Verfahren auf diese Kriterien hin untersucht. Die Gesamtheit dieser Kriterien floss in die Entwicklung der Evaluierungsumgebung mit ein, welche als Ergebnis dieser Arbeit entstanden ist. Sie erleichtert das Testen von Lokalisierungsverfahren und ermöglicht den Vergleich verschiedener Verfahren hinsichtlich ihrer Ortungsgenauigkeit.

Inhaltsverzeichnis

Kurzfassung	III
Inhaltsverzeichnis	V
Abbildungsverzeichnis	IX
Tabellenverzeichnis	XI
Abkürzungsverzeichnis	XIII
1 Einleitung	1
1.1 Motivation	1
1.2 Ziel und Aufgabenstellung	2
1.3 Ergebnis	3
1.4 Gliederung	3
2 Grundlagen	5
2.1 Drahtlose Netze	5
2.1.1 WLAN	6
2.1.2 Frequenzen	6
2.1.3 Arbeitsmodi	7
2.1.4 Signalausbreitung	7
2.2 Lokalisierung im WLAN	8
2.2.1 Signalinformationen	8
2.2.2 Variation der Signalparameter	10
2.2.3 Offline-Phase	13
2.2.4 Online-Phase	14
2.3 Ausgewählte Lokalisierungsalgorithmen	15

2.3.1	RADAR	15
2.3.2	WCL	17
2.4	Ähnliche Arbeiten	17
2.4.1	Ekahau	18
2.4.2	MagicMap	19
2.5	Einflüsse auf das Ergebnis einer Evaluierung	19
2.5.1	Einflüsse der Testumgebung	20
2.5.2	Einflüsse der Testparameter	21
2.5.3	Notwendigkeit einer Evaluierungsumgebung	21
3	Konzepte	23
3.1	Universalitätsanforderungen	23
3.1.1	Geometrische und kartenbasierte Verfahren	24
3.1.2	Besondere Daten im Vorwissen	24
3.1.3	Unterschiedliche Signalinformationen	25
3.1.4	Statisches und mobiles Nutzerprofil	25
3.1.5	Deterministische und probabilistische Verfahren	25
3.1.6	Besondere Features der Evaluierungsumgebung	26
3.2	Aufbau der Evaluierungsumgebung	26
3.2.1	Logische Trennung der Komponenten	26
3.2.2	Kommunikation zwischen den Komponenten	31
3.2.3	Austauschbarkeit der Algorithmen	34
3.3	Evaluierung von Lokalisierungsalgorithmen	34
3.3.1	Festlegung und Variation der Testparameter	35
3.3.2	Vergleich mehrerer Algorithmen	36
3.3.3	Ermittlung der Offline-Evaluierungspunkte	39
4	Implementierung	43
4.1	Umsetzung der Universalitätsanforderungen	43
4.1.1	Daten des Vorwissens	43
4.1.2	Behandlung unterschiedlicher Signalinformationen	48
4.1.3	Benutzer-Korrektur	48
4.2	Umsetzung des Komponentenaufbaus	49
4.2.1	Kommunikation	49

4.2.2	Algorithmen als Module	54
4.3	Umsetzung der Evaluierung	57
4.3.1	Variable Testparameter	57
4.3.2	Aufnahme von Offline-Evaluierungspunkten	58
4.3.3	Berechnung der Evaluierung	61
4.4	Sonstige Informationen	63
5	Experimentelle Bewertung der Evaluierungsumgebung	65
5.1	Testumgebung und verwendete Hardware	65
5.2	Evaluierung von RADAR	66
5.2.1	Versuchsaufbau	66
5.2.2	Durchführung	67
5.2.3	Ergebnisse	68
5.2.4	Schlussfolgerung	74
5.3	Optimierung von RADAR	74
5.3.1	Versuchsaufbau	74
5.3.2	Durchführung	75
5.3.3	Ergebnisse	75
5.3.4	Schlussfolgerung	79
5.4	Evaluierung von WCL	79
5.4.1	Versuchsaufbau	80
5.4.2	Durchführung	80
5.4.3	Ergebnisse	80
5.4.4	Schlussfolgerung	83
5.5	Vergleich zwischen RADAR und WCL	83
5.5.1	Versuchsaufbau	84
5.5.2	Durchführung	84
5.5.3	Ergebnisse	84
5.5.4	Schlussfolgerung	85
5.6	Zusammenfassung der Experimente	86
6	Zusammenfassung und Ausblick	87
6.1	Zusammenfassung	87
6.2	Ausblick	89

Inhaltsverzeichnis

A Umgebungskarten	91
Literaturverzeichnis	97
Selbstständigkeitserklärung	101

Abbildungsverzeichnis

2.1	Schwankungen der Signalstärke an einem konstanten Ort (aus [12])	11
2.2	Kleinräumige RSSI-Variationen (aus [5])	12
3.1	Aufgaben der Komponenten im Client-Server-Modell	28
3.2	Aufgaben der Komponenten im minimaler Client-Server-Modell	29
3.3	Komponentenaufgaben im minimaler Client-Manager-Server-Modell . . .	30
3.4	gekapselter Aufbau des Servers	35
4.1	Struktur der Datei „training.txt“ am Beispiel	48
4.2	Struktur der Datei „evalPoints.txt“ am Beispiel (statisches Nutzerprofil) .	59
4.3	Struktur der Datei „evalPoints.txt“ am Beispiel (mobiles Nutzerprofil) . .	60
4.4	Struktur einer Evaluierungsdatei am Beispiel	61
5.1	RADAR: Vergleich der Messergebnisse bei maximaler Signalstärke und variabler Anzahl von Nachbarn	72
A.1	Versuchsaufbau der aktuellen Tests von RADAR (APs und Trainingspunkte)	91
A.2	Versuchsaufbau der originalen Tests von RADAR	92
A.3	Versuchsaufbau der aktuellen Tests (statische Offline-Evaluierungspunkte)	93
A.4	Versuchsaufbau der aktuellen Tests (mobile Offline-Evaluierungspunkte) .	94
A.5	Versuchsaufbau der aktuellen Tests für WCL (APs)	95

Tabellenverzeichnis

2.1	Skaleneinteilung der RSSI-Werte verschiedener Hersteller	11
3.1	Bewertung der einzelnen Varianten der Komponententrennung	31
3.2	Eignung der Transportprotokolle für Client-Server-Kommunikation	33
3.3	Leistungsmerkmale der Evaluierungstechniken	38
3.4	Bewertung der Ermittlungsmethoden der Offline-Evaluierungspunkte	41
4.1	Umgebungskarten-Daten in der Topologie	44
4.2	AP-Daten in der Topologie	46
4.3	Kommunikation für das Absolvieren eines Trainings	50
4.4	Kommunikation für den Austausch ortungsrelevanter Informationen	51
4.5	Kommunikation für die Lokalisierung von Clients	52
4.6	Kommunikation für die Durchführung einer Evaluierung	52
4.7	rein virtuelle Methoden des <i>EngineHolders</i> , Befehle vom Manager	55
4.8	rein virtuelle Methoden des <i>EngineHolders</i> , Daten vom Manager	55
4.9	rein virtuelle Methoden des <i>EngineHolders</i> , Veränderung wichtiger Variablen	56
4.10	rein virtuelle Methoden des <i>EngineHolders</i> , Client-bezogene Informationen	57
5.1	RADAR: Vergleich der Testergebnisse mit einem Nachbar	69
5.2	RADAR: Messergebnisse in Abhängigkeit von der Anzahl der Nachbarn	69
5.3	RADAR: Vergleich der Testergebnisse mit variabler Anzahl von Nachbarn	70
5.4	RADAR: Vergleich der Testergebnisse bei Verwendung der maximalen Signalstärke	71
5.5	RADAR: Vergleich der Testergebnisse, mobiles Nutzerprofil	73
5.6	Optimierung von RADAR: Messergebnisse in Abhängigkeit von der Anzahl der APs	76

5.7	Optimierung von RADAR: Messergebnisse in Abhängigkeit von der Anzahl der Trainingspunkte	76
5.8	Optimierung von RADAR: Messergebnisse in Abhängigkeit von der Menge der Messwerte je Trainingspunkt	77
5.9	Optimierung von RADAR: Messergebnisse in Abhängigkeit von der Menge der Messwerte je Evaluierungspunkt	78
5.10	Optimierung von RADAR: Vergleich der Messergebnisse zwischen verbesserter Konstellation und Standard-Konstellation	78
5.11	WCL: Messergebnisse in Abhängigkeit von der Lokalisierungszeit bei statischem Nutzerprofil (9 APs)	81
5.12	WCL: Messergebnisse in Abhängigkeit von der Lokalisierungszeit bei mobilem Nutzerprofil (9 APs)	82
5.13	WCL: Vergleich der Messergebnisse, Online- und Offline-Evaluierung bei statischem Nutzerprofil (9 APs)	82
5.14	WCL: Messergebnisse in Abhängigkeit von der Lokalisierungszeit bei statischem Nutzerprofil (7 APs)	83
5.15	RADAR und WCL: Vergleich der Messergebnisse, unveränderte Versionen	85
5.16	RADAR und WCL: Vergleich der besten Messergebnisse	85

Abkürzungsverzeichnis

AP	Access Point
CGLCD	Coarse Grained Localization with Center Determination
DSL	Digital Subscriber Line
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
MAC	Media Access Control
NNSS	Nearest Neighbor(s) in Signal Space
PDA	Personal Digital Assistant
PCMCIA	Personal Computer Memory Card International Association
RSSI	Received Signal Strength Indication
SSID	Service Set Identifier
SVG	Scalable Vector Graphics
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WCL	Weighted Centroid Localization
WLAN	Wireless Local Area Network
XML	eXtensible Markup Language

1 Einleitung

1.1 Motivation

Drahtlose Netze sind wegen ihrer vielen positiven Aspekte inzwischen weit verbreitet. Es gibt unzählige private, kommerzielle und wissenschaftliche Anwendungsfelder. Sie dienen zur drahtlosen Vernetzung in Gebäuden und größeren Flächen, wie zum Beispiel einem Universitäts-Campus oder einem Bahnhof. In einem WLAN (Wireless Local Area Network) ist es möglich, drahtlos zu kommunizieren und mittels eines APs (Access Points) ortsungebunden auf die Angebote des drahtgebundenen Netzes zuzugreifen.

Diese Ortsungebundenheit verhindert jedoch die nützliche Lieferung ortsabhängiger Inhalte und Dienste. Ein Beispiel ist die Positionsbestimmung durch GPS (Global Positioning System), die für ein Navigationssystem im Auto unerlässlich ist. Jedoch benötigt GPS spezielle Hardware und funktioniert nur im Freien. In einem Gebäude mit einem drahtlosen Netz ist eine Ortung ohne spezielle, teilweise teure, Hardware möglich. Dazu sind Lokalisierungsverfahren notwendig, welche die Position eines mobilen Gerätes innerhalb eines drahtlosen Netzes ermitteln können. Es ergeben sich viele Anwendungsmöglichkeiten, angefangen mit der Ortung eines mobilen Gerätes im Notfall, über die Möglichkeit der Navigation in einem Gebiet mit gleichzeitiger Lieferung von Daten über spezielle Sehenswürdigkeiten, bis hin zur Zugangskontrolle eines WLANs, abhängig vom Ort des Benutzers. Heutige Verfahren ermöglichen eine Positionsbestimmung auf wenige Meter genau.

Inzwischen existieren viele verschiedene Lokalisierungsalgorithmen mit unterschiedlichen technischen, konzeptionellen und statistischen Ansätzen. Neben Zielen, wie einem geringen Berechnungsaufwand, haben diese Verfahren ein gemeinsames Hauptziel: Die Fehlerdistanz minimieren! Das ist die Entfernung zwischen der realen und der errechneten Position des mobilen Gerätes. Um diese Fehlerdistanz und andere Parameter zu ermitteln, ist eine Evaluierung des Verfahrens notwendig.

Die Entwickler testen ihr Lokalisierungsverfahren in einer selbst gewählten Testumgebung, mit selbst gewählten Testparametern und geben, darauf aufbauend, die Genauigkeit ihres Verfahrens an. Ein ausführlicher Vergleich zwischen der Leistungsfähigkeit mehrerer Verfahren ist aufgrund dieser unterschiedlichen Voraussetzungen nicht möglich. Die Genauigkeit eines Lokalisationsverfahrens ist nicht nur vom Verfahren selbst, sondern auch von der Testumgebung und den Testparametern abhängig. So fällt die Bewertung der Qualität einzelner Lokalisationsverfahren schwer.

1.2 Ziel und Aufgabenstellung

Es soll Aufgabe dieser Diplomarbeit sein, eine Evaluierungsumgebung zu schaffen, die es erlaubt mehrere Lokalisierungsverfahren, unabhängig von ihren unterschiedlichen Grundvoraussetzungen, zu testen. Sie soll die Evaluierung erleichtern und einen aussagekräftigen Vergleich zwischen den einzelnen Verfahren zulassen. Um diesen Vergleich anstellen zu können, ist es notwendig, jedes einzelne Verfahren unter gleichen Bedingungen zu testen. Dies soll durch die Evaluierungsumgebung ermöglicht werden.

Die Evaluierungsumgebung muss in der Lage sein, den Aufbau und die Eigenschaften eines drahtlosen Netzes aufzunehmen und zu verarbeiten, da diese für die Berechnung der Position zwingend erforderlich sind. Dies geschieht, abhängig vom jeweiligen Lokalisierungsalgorithmus, entweder durch die Angabe der Topologie des Netzwerkes oder durch ein Training. Die Art der Speicherung der Topologie ist dabei so zu wählen, dass sie leicht maschinell und menschlich lesbar ist, damit sie auch außerhalb der Evaluierungsumgebung erstellt und genutzt werden kann.

Es ist notwendig, dass Umgebungskarten (beispielsweise Gebäudepläne) dargestellt werden, auf deren Grundlage eine Orientierung und eine Angabe von Positionen möglich sind. Die Darstellungsart sollte so gewählt sein, dass es möglich ist, überaus große Karten anzuzeigen, ohne dabei zu viele Ressourcen auf dem darstellenden Gerät zu verbrauchen. Auch die zu ortenden mobilen Geräte sollen mit möglichst wenig Ressourcen auskommen.

Ein wichtiges Kriterium ist die Universalität. Darum ist darauf zu achten, dass eine möglichst große Zahl verschiedenartiger Algorithmen unterstützt wird. Dies wird exemplarisch an zwei Algorithmen gezeigt, wobei beide Arten der Speicherung des Aufbaus des drahtlosen Netzes (Angabe der Topologie, Training) vertreten sind.

1.3 Ergebnis

Im Rahmen dieser Diplomarbeit wurde eine Evaluierungsumgebung für Lokalisierungsverfahren in drahtlosen Netzen entwickelt, mit der man in der Lage ist, mehrere verschiedene Verfahren zu vergleichen. Dadurch wird eine leichtere und aussagekräftigere Analyse ermöglicht, was auch für die Entwicklung neuer Lokalisierungsalgorithmen hilfreich ist. Dazu wurden zuerst bekannte Lokalisierungsalgorithmen auf ihre Vorbedingungen hin untersucht. Das Wissen aus dieser Untersuchung bildete die Liste der notwendigen Eigenschaften der Evaluierungsumgebung.

Es wurden auch bereits verfügbare Lokalisierungsprogramme analysiert, die zwar nur jeweils einen einzigen Algorithmus unterstützen und deren Hauptaugenmerk nicht auf der Evaluierung sondern auf der Benutzung liegt. Jedoch dienten sie als Beispiele für die visuelle Darstellung und Trennung der Komponenten für diese Evaluierungsumgebung. Aufgrund dieser Erkenntnisse wurde die Evaluierungsumgebung in drei logische Teile getrennt: ein Manager, der die Vorbedingungen für den jeweiligen Lokalisierungsalgorithmus bereitstellt und die anderen Komponenten steuert. Ein lokalisierbares mobiles Gerät, das ortsspezifische Signaldaten sammelt. Ein Server, der mithilfe des universell einsetzbaren Algorithmus und den vom Manager erfüllten Vorbedingungen, den Ort mehrerer mobiler Geräte ermitteln kann.

Abschließend wurde die Universalität der Evaluierungsumgebung durch die Implementierung, Benutzung und Evaluierung von zwei Lokalisierungsalgorithmen belegt. Dies waren RADAR [1] und WCL [2, 3]. Es wurde gezeigt, dass die Evaluierungsumgebung die Anforderungen erfüllt und die Evaluierung von Lokalisationsverfahren erleichtert.

1.4 Gliederung

In Kapitel 2 werden die Grundlagen behandelt, die für das Verständnis der Arbeit notwendig sind. Der Aufbau und die besonderen Eigenschaften von drahtlosen Netzen werden eingangs erklärt. Anschließend werden der Vorgang der Lokalisierung und einige Lokalisierungsalgorithmen näher beschrieben. Am Ende folgen eine Vorstellung von einigen Lokalisierungsprogrammen und die Erläuterung von Einflüssen auf das Ergebnis einer Evaluierung.

Kapitel 3 beschäftigt sich mit den Konzepten dieser Evaluierungsumgebung. Es wird anfangs auf das Erreichen der geforderten Universalität eingegangen und später auf den logi-

schen Aufbau der Evaluierungsumgebung. Abschließend geht es um die Evaluierung von Lokalisierungsalgorithmen.

In Kapitel 4 wird die Implementierung der Evaluierungsumgebung beschrieben. Zu Beginn wird die Umsetzung der Universalitätsanforderungen und danach die Verwirklichung des Komponentenaufbaus erklärt. Die Erläuterungen zur Implementierung der Evaluierung folgen am Schluss.

Im Anschluss daran folgt in Kapitel 5 die experimentelle Bewertung der Evaluierungsumgebung. Die beiden Algorithmen RADAR und WCL werden evaluiert und verglichen, um die Funktionalität der Evaluierungsumgebung zu beweisen.

Das Ende bilden eine Zusammenfassung und ein Ausblick in Kapitel 6.

2 Grundlagen

Die Entwicklung einer Evaluierungsumgebung für Lokalisierungsverfahren in drahtlosen Netzen setzt Grundwissen voraus, welches in diesem Kapitel vermittelt werden soll.

Zuerst wird erklärt, was drahtlose Netze sind und welche wichtigen Eigenschaften sie haben. Dabei wird besonders auf WLAN (Wireless Local Area Network) eingegangen. Als nächstes werden die generelle Grundstruktur und die Arbeitsweisen von Lokalisierungsverfahren näher betrachtet. Danach folgt die Vorstellung einiger spezieller Lokalisierungsalgorithmen. Anschließend dienen zwei bekannte Lokalisierungsprogramme als Anschauungsobjekte, anhand derer der prinzipielle und logische Aufbau einer Evaluierungsumgebung abgeleitet werden kann. Abschließend wird beschrieben wie das Ergebnis einer Evaluierung durch die Testumgebung und der Testparameter verändert werden kann.

2.1 Drahtlose Netze

Die Telekommunikation machte mit der Entdeckung der elektromagnetischen Wellen einen großen Schritt vorwärts. Diese Wellen setzen sich ohne spezielle Träger fort und erlauben so eine Kommunikation ohne vorherige Verbindung der Teilnehmer mit einem Kabel. Die zu übermittelnden Daten werden mittels Modulation der Phase, Frequenz und/oder Amplitude kodiert. Abhängig von der Frequenz, der Modulationsart, den Sende- und Empfangsantennen und der Sendeleistung liegt die Reichweite drahtloser Netze zwischen wenigen Zentimetern und tausenden Kilometern. Seit Beginn des 20. Jahrhunderts entwickelte sich die Technik der drahtlosen Kommunikation ständig weiter und ermöglicht so heute eine Vielzahl von Anwendungen, wie Radio, Funktelefon, Fernsehen, GPS und drahtlose Computernetzwerke, dessen Vertreter zum Beispiel Bluetooth, IrDA und WLAN sind. WLAN wird nun folgend näher erläutert. Dabei wird auf die verwendeten Frequenzen und die möglichen Arbeitsmodi eingegangen. Am Ende wird die Signalausbreitung diskutiert.

2.1.1 WLAN

WLAN steht für „Wireless Local Area Network“ und beschreibt somit die drahtlose Variante eines LANs (Local Area Network), welches ein örtlich begrenztes Computernetzwerk ist. WLAN ist ein vom IEEE (Institute of Electrical and Electronics Engineers) herausgegebener Industriestandard mit der Bezeichnung IEEE 802.11. Daneben gibt es noch andere Kurzstreckenfunksysteme, die zum Aufbau eines Computernetzwerkes geeignet sind, beispielsweise Bluetooth (IEEE 802.15). WLAN als Oberbegriff vereint mehrere Protokolle unter sich, wie zum Beispiel IEEE 802.11b, welcher eine Übertragungsrate von 11 MBit/s hat oder IEEE 802.11a mit einer Übertragungsrate von 54 MBit/s. Aufgrund seines geringen Installationsaufwandes hat WLAN inzwischen in vielen Bereichen des Lebens Einzug gehalten. In privater Umgebung dient es als drahtlose Anbindung an einen DSL-Router (Digital Subscriber Line) oder zum Aufbau eines kleinen Privatnetzes. Auch in vielen Büros ist drahtlose Kommunikation anstelle des unhandlichen Kabels getreten.

2.1.2 Frequenzen

In Deutschland existieren zwei frei nutzbare Frequenzbänder, auf denen handelsübliche WLAN-Geräte arbeiten. Dies sind zum einen das ISM-Band (Industrial, Scientific and Medical) bei 2,4 GHz - 2,4835 GHz, auf dem Geräte nach dem IEEE 802.11b- und IEEE 802.11g-Standard arbeiten und zum anderen das erst seit dem 13.11.2002 frei verfügbare 5-GHz-Band (5,15 GHz - 5,725 GHz), welches vom Geräten des Standards IEEE 802.11a und IEEE 802.11h benutzt wird.

Aufgrund der Notwendigkeit des Einsatzes mehrerer getrennter WLAN-Systeme in einem Gebiet, benutzt eine Station nicht das gesamte Frequenzband, sondern nur einen kleinen Frequenzbereich. Die Frequenzbänder sind in mehrere dieser Frequenzbereiche, Kanäle genannt, unterteilt. Beim ISM-Band sind in Deutschland 13 Kanäle verfügbar, beim 5-GHz-Band 19. Eine Station benutzt einen Kanal um mit anderen Stationen auf dem gleichen Kanal zu kommunizieren. Es ist ihr nicht möglich, auf mehreren Frequenzbereichen gleichzeitig zu senden oder zu empfangen. Allerdings kann sie die Frequenz jederzeit wechseln. Das dauert jedoch einige Millisekunden, was dazu führt, dass die Station während dieser Zeit nicht kommunizieren kann und eventuell an sie geschickte Pakete verloren gehen. Das Wechseln des Kanals ist ein elementarer Prozess bei der Messung der Signaleigenschaften.

2.1.3 Arbeitsmodi

Mit dem Ad-hoc-Modus und dem Infrastrukturmodus bietet WLAN zwei verschiedene Arbeitsmodi an. Im Ad-hoc-Modus können alle Stationen innerhalb ihrer Reichweite miteinander kommunizieren und so autonome, spontane Zellen bilden. Dabei wird keine vorhandene Infrastruktur benötigt und es gibt hier keine feste Hierarchie unter den Teilnehmern. Eine Kommunikation ist nur innerhalb der Zelle möglich. Es sind weitere Dienste nötig, um einen Datentransfer zu weit entfernten Stationen oder anderen Netzwerken zu gewährleisten.

Anders ist dies im Infrastrukturmodus, wo die Kommunikation durch APs (Access Points) geregelt wird. Die einzelnen Endgeräte bilden hier kein loses Netz, sondern sie melden sich bei einem AP an. Das sind meistens unbewegliche Basisstationen, die eine Verbindung zu einem drahtgebundenen Medium haben, wodurch die bei ihnen angemeldeten Stationen die Möglichkeit der Kommunikation zu anderen Teilnehmern und Netzwerken bekommen. Die mobilen Stationen kommunizieren dabei ausschließlich mit dem AP, wobei dieser die Sendereihenfolge festlegt. Der AP als zentraler Knotenpunkt definiert die Zelle und legt alle Sendeparameter, wie zum Beispiel den Kanal und die Verschlüsselung, fest. Um es Stationen zu erleichtern eine Zelle zu finden, sendet der AP in regelmäßigen Abständen (meist alle 100 Millisekunden) ein Beacon aus. Ein Beacon ist ein Paket, das alle nötigen Informationen enthält, die erforderlich sind, um sich bei diesem AP anzumelden und so dieser Zelle beizutreten. Viele Lokalisierungsverfahren nutzen die aus einem Beacon gewonnenen Informationen zur Berechnung des Ortes.

2.1.4 Signalausbreitung

Radiowellen werden beim Durchdringen von Materie abgeschwächt. Besonders eisenhaltige Materialien, wie Stahlbetonwände und Maschinen, dämpfen das Signal sehr stark. Auch Wasser, vorkommend in Menschen und als Feuchtigkeit in der Luft, ist ein starker Dämpfer. Der Indikator für die Empfangsfeldstärke, also die Qualität des Signals, wird Signalstärke oder RSSI (Received Signal Strength Indication) genannt und in dBm angegeben. Je mehr ein Signal gedämpft wird, desto geringer ist seine Signalstärke. Unterschreitet die Signalstärke, bedingt durch Dämpfung, einen bestimmten Wert, ist es für einen Empfänger nicht mehr dekodierbar. Die Erkenntnis, dass die Signalstärke durch steigende Entfernung vom Sender und Hindernisse abnimmt, ist eine Grundvoraussetzung für

viele Lokalisierungsalgorithmen.

Unter idealen Bedingungen ist eine maximale Reichweite eines WLANs von mehreren hundert Metern möglich. Jedoch ist sie im Normalfall aufgrund der Dämpfung deutlich geringer, da WLANs meistens in Gebäuden und seltener im Freien eingesetzt werden. Die maximale Reichweite und die Ausprägung der Dämpfung sind von der Sendeleistung, dem Modulationsverfahren und der verwendeten Frequenz abhängig. Eine geringere Frequenz und eine höhere Sendeleistung vergrößern die Reichweite. Der Faktor der Dämpfung, also wie stark ein Signal durch ein Hindernis abgeschwächt wird, ist ebenfalls von der Frequenz abhängig. Wird ein Signal durch Amplitudenmodulation kodiert, wie es zum Beispiel im Standard IEEE 802.11a und IEEE 802.11h festgelegt ist, ist es deutlich anfälliger gegen Dämpfung, als wenn es nur durch Phasen- oder Frequenzmodulation erzeugt wurde. Der Grund ist, dass beim stetigen Energieverlust der Welle eine Verringerung der Amplitude zu verzeichnen ist. Diese Unterschiede fließen auch in die Ortsberechnung einiger Lokalisationsverfahren mit ein.

2.2 Lokalisierung im WLAN

Lokalisierungsverfahren für drahtlose Netze ermöglichen die Ortung eines mobilen Gerätes auf wenige Meter genau, meistens ohne dafür zusätzliche Hardware zu benötigen.

Lokalisierungsalgorithmen arbeiten normalerweise in zwei Phasen. Die erste Phase (Offline-Phase) dient der Erstellung des Vorwissens, welches für die Berechnung des Ortes unbedingt erforderlich ist. In der zweiten Phase (Online-Phase) geschieht dann die eigentliche Lokalisierung. Die vom zu ortenden mobilen Gerät, auch Client genannt, gesendeten ortsspezifischen Signalinformationen werden dazu benutzt.

Welche Arten von ortsspezifischen Informationen es gibt und welchen Schwankungen sie unterliegen wird nacheinander in den folgenden beiden Kapiteln erklärt. Im Anschluss daran werden die Offline- und die Online-Phase näher erläutert.

2.2.1 Signalinformationen

Lokalisierungsalgorithmen benutzen physikalische Veränderungen des Signals oder Übertragungseigenschaften als Parameter für ihre Berechnungen. Mögliche Parameter sind die Signalstärke, das Verhältnis der empfangen und maximal empfangbaren Beacons, der Winkel zwischen den Kommunikationspartnern und die Zeitdifferenz zwischen Versen-

den und Empfangen eines Paketes. Dabei setzen die Algorithmen meistens einige feste Sendestationen als Fixpunkte voraus.

Signalstärke: Aufgrund der Tatsache, dass elektromagnetische Wellen durch Luft und Hindernisse abgeschwächt werden (siehe Kapitel 2.1.4), kann man vermuten, dass mit steigender Entfernung vom Sender die Signalstärke immer geringer wird und dass sich diese Beziehung durch eine Gleichung ausdrücken lässt. Mit dieser idealisierten Annahme lässt sich von der empfangenen Signalstärke auf die Entfernung schließen. Ebenfalls kann man dadurch annehmen, dass die Signalstärke an einer bestimmten Position immer den gleichen Wert hat. Diese Annahmen entsprechen nicht den wirklichen Verhältnissen in einem drahtlosen Netz (siehe Kapitel 2.2.2). Sie kommen jedoch der Realität zumindest nah und ermöglichen so die ungefähre Bestimmung des Aufenthaltsortes eines mobilen Gerätes. Um die Signalstärkewerte aller im Empfangsbereich liegenden APs messen zu können, muss das mobile Gerät alle vorhandenen Kanäle durchlaufen und ein Paket von dem AP empfangen, der in diesem Kanal sendet. Dieses erwähnte Paket ist meistens ein Beacon, da dieser verlässlich und regelmäßig vom AP versendet wird. Ein Großteil der Lokisierungsalgorithmen benutzt die Signalstärke, in Form des RSSI-Wertes, als Parameter.

Anzahl empfangener Beacons: Ein ähnliches Prinzip wird angewandt, wenn das Verhältnis der empfangenen und maximal empfangbaren Beacons zur Berechnung der Position benutzt wird. Anhand dieses Prozentsatzes ist, wenn auch sehr grobgranulös, eine Vermutung über die Qualität des Signals möglich. Hier wird angenommen, dass nur ein Teil, der vom Sender in regelmäßigen Zeitabständen geschickten Beacons, beim Empfänger ankommt. Da mit steigender Entfernung die Signaleigenschaften schlechter werden, erreichen dadurch weniger Pakete ihr Ziel.

Winkel: Ein von der Signalstärke unabhängiges Verfahren ist das Messen des Winkels von empfangenen Paketen. Speziell dafür entwickelte Antennen sind in der Lage die Herkunftsrichtung eines Signals zu ermitteln. Sind Winkel zu mehreren APs bekannt, ist eine Triangulation und somit eine Lokalisierung möglich.

Zeitdifferenz: Eine andere Möglichkeit, die Distanz zwischen zwei Punkten zu ermitteln, ist die Messung der Zeit, die ein Signal von einem zum anderen Punkt benötigt [4].

Die zurückgelegte Strecke ergibt sich aus dem Produkt der Signalausbreitungsgeschwindigkeit und der verstrichenen Zeit. Radiowellen bewegen sich, wie alle anderen elektromagnetischen Wellen auch, im Vakuum mit Lichtgeschwindigkeit fort. Bei dieser Geschwindigkeit legt das Signal 300 Meter in einer Mikrosekunde (μs) zurück. Aufgrund dieser hohen Geschwindigkeit hängt die Exaktheit der Ortung maßgeblich von der Synchronisation der Kommunikationspartner und der Messgenauigkeit der Zeit ab.

2.2.2 Variation der Signalparameter

Leider ergeben mehrere Messungen der Signaleigenschaften an einem Ort teils geringfügig, teils beträchtlich voneinander abweichende Werte, wodurch die Genauigkeit der Lokalisierungsalgorithmen sinkt. Verantwortlich dafür sind die natürlichen Schwankungen der Signaleigenschaften, die nun näher beschrieben werden.

Signalstärke: Wäre es möglich, die Beziehung zwischen der Signalstärke und der zurückgelegten Distanz durch eine exakte Gleichung auszudrücken, könnte man eine Ortung auf den Zentimeter genau ausführen. Dies ist jedoch aufgrund mehrerer Faktoren nicht möglich.

Jeder Gegenstand hat aufgrund seines Materials andere physikalische Eigenschaften, was dazu führt, dass Signale durch ihn unterschiedlich gedämpft, gestreut und reflektiert werden. Einige dieser Hindernisse, wie zum Beispiel Wände, Möbel und andere stationäre Gegenstände, könnten in die Berechnung der Gleichung mit einbezogen werden. Jedoch bilden ein Großteil der Hindernisse, wie Menschen, Türen, veränderte Luftfeuchtigkeit, Vegetation und andere bewegliche Gegenstände sowie die negativen Auswirkungen mancher Elektrogeräte, eine ständige Zufallskonstante. So kommt es, dass trotz feststehender Kommunikationspartner, die Signalstärke über einen längeren Zeitraum um 10-15 dBm schwankt (siehe Abbildung 2.1).

Fest installierte APs haben omnidirektionale Antennen und sind meistens so montiert, dass die erste Fresnel-Zone¹ nicht verletzt ist. Das sieht bei mobilen Geräten, wie Laptops meist anders aus. Sie haben teilweise lediglich eine PCMCIA-Karte² (Personal Computer Memory Card International Association) als Adapter. Hier ist die Antenne relativ klein

¹Die erste Fresnel-Zone ist der Bereich zwischen Sender und Empfänger, in dem die meiste Energie übertragen wird. Hindernisse in diesem Bereich stören die Übertragung stark.

²PCMCIA ist ein Standard für einsteckbare Erweiterungskarten mobiler Computer. PCMCIA-Karten haben eine Größe von 85,6mm x 54,0mm und sind Hot-Plug- und Plug&Play-fähig.

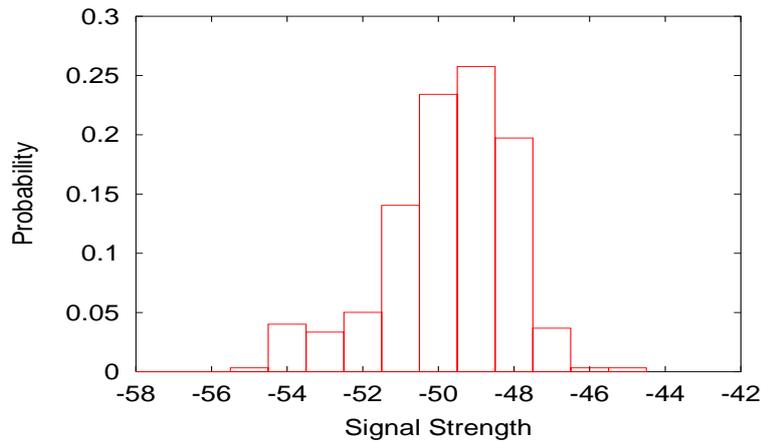


Abbildung 2.1: Schwankungen der Signalstärke an einem konstanten Ort (aus [12])

und zeigt, bedingt durch die Lage des Laptops, in eine bestimmte Richtung, im Folgenden Orientierung genannt. Aus diesem Grund und der Tatsache, dass der mit dem Laptop arbeitende Mensch ein orientierungsabhängiges Hindernis bildet, ergeben sich unterschiedliche Signalstärkewerte an ein und demselben Ort.

Eine weitere Ursache sind „kleinräumige RSSI-Variationen“ [5, 6] (siehe Abbildung 2.2). Dieses Phänomen tritt bei einer Verschiebung eines Kommunikationspartners im Bereich der Wellenlänge auf. Im ISM-Band beträgt diese ca. 12,5cm. Durch eine so geringe Veränderung des Ortes kann der RSSI-Wert um bis zu 10 dBm variieren.

Die unterschiedlichen für die Ortung eingesetzten Netzwerkadapter führen zu einer weiteren Ungenauigkeit. Jeder Hersteller verwendet für die Produktion andere Bauteile und andere Software. Darüber hinaus benutzen sie teilweise eine andere Skaleneinteilung für die RSSI-Werte [7] (siehe Tabelle 2.1). Selbst baugleiche Adapter unterscheiden sich durch geringfügige Fabrikationsabweichungen. Dies führt dazu, dass verschiedene Adapter unter den gleichen Voraussetzungen geringfügig andere Werte für die Signalstärke liefern.

Hersteller	RSSI-Bereich	Energiebereich (dBm)	Schrittweite (dBm)
Cisco	0...100	-113...-10	1
Atheros	0...60	-95...-35	1
Symbol	0...31	-100...-50	10

Tabelle 2.1: Skaleneinteilung der RSSI-Werte verschiedener Hersteller

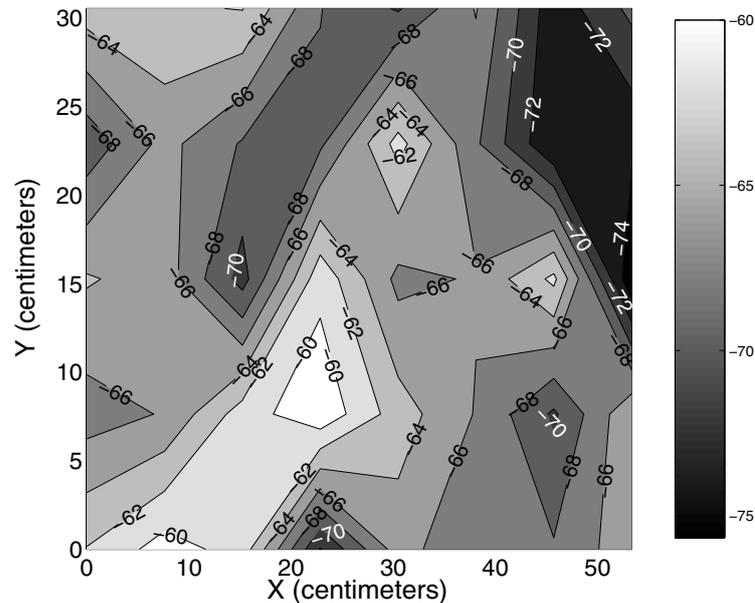


Abbildung 2.2: Kleinräumige RSSI-Variationen (aus [5]). Dargestellt sind die RSSI-Werte auf einer kleinen Fläche. In der Skala rechts sind die Farbwerte der Signalstärkewerte abgetragen.

Winkel: Die Methode der Winkelmessung ist zwar gegen die Schwankungen der Signalstärke immun, jedoch erweisen sich hier die Mehrfachausbreitungen des Signals als das eigentliche Problem. Wenn ein Signal, aufgrund einer Reflexion, das Ziel auf einem anderen Weg erreicht, ändert sich dadurch der am Ziel gemessene Winkel. Dieser Winkel ist dann nicht mehr auf die sendende Station gerichtet, sondern auf den Ort der Reflexion.

Zeitdifferenz: Aufgrund der hohen Geschwindigkeit des Signals, ist die größte Fehlerquelle die Messgenauigkeit. Weitere Fehlerquellen bei der Methode der Zeitmessung sind die unterschiedlichen Ausbreitungsgeschwindigkeiten innerhalb verschiedener Stoffe und Mehrfachausbreitungen des Signals. In Stahlbeton beträgt die Geschwindigkeit von elektromagnetischen Wellen nur $1/3$ der Lichtgeschwindigkeit. Die Signalsendezeit wird durch Hindernisse beeinflusst. Dieser Einfluss ist jedoch nicht so groß, wie bei der Methode der Signalstärke, da sich nach dem Durchdringen des Hindernisses das Signal wieder mit der normalen Geschwindigkeit fortbewegt. Ein Signal, das sein Ziel auf einem längeren Weg erreicht, bedingt durch Mehrfachausbreitungen, benötigt dafür auch mehr Zeit. Ein weiteres Problem bei der Zeitmessung ist die Verarbeitungszeit. Wenn eine Station die Zeit zwischen dem Versand eines Paketes und dem Erhalt des Bestätigungspaketes

(Acknowledge) misst, ist darin die Verarbeitungszeit der zweiten Station enthalten. Diese Zeit ist variabel und muss von der messenden Station geschätzt werden. Dies führt zu Ungenauigkeiten.

2.2.3 Offline-Phase

Bevor ein mobiles Gerät geortet werden kann, müssen dem Lokalisierungsalgorithmus Informationen über das drahtlose Netzwerk zur Verfügung gestellt werden, anhand derer er eine Entscheidung über den Aufenthaltsort treffen kann. Zum einen können diese Informationen die Topologie des Netzes sein. Zum anderen können sie aus einem Satz von Trainingspunkten bestehen.

Wenn ein Algorithmus die Topologie eines drahtlosen Netzes benutzt, so stehen ihm die Positionen und weitere Daten von festen Sendestationen (meistens APs) zur Verfügung. Es ist ebenfalls möglich, weitere Daten in die Topologie zu übernehmen, die für die Berechnung relevant sind (z. B. Hindernisse). Die Topologie muss an Veränderungen der Umwelt angepasst werden. Dies ist jedoch relativ einfach möglich, indem die Positionen und Daten der verzeichneten APs korrigiert werden. Wenn die Topologie für die Berechnung des Ortes zur Verfügung steht, werden solche Verfahren „geometrisch“ genannt.

Verwendet ein Algorithmus einen Satz von Trainingspunkten, so steht ihm eine so genannte „Radiomap“ zur Verfügung. Sie besteht aus einer Liste mehrerer Positionen, samt den dort beispielhaft gemessenen Signalparametern. Um ein möglichst gutes Resultat bei der Lokalisierung zu erzielen, müssen die Trainingspunkte gleichmäßig in der Umgebungskarte verteilt sein. Das Erstellen dieser Radiomap ist aufgrund der vielen Messpunkte sehr zeitaufwendig. Im Falle einer Änderung am Aufbau des WLANs oder bei größeren baulichen Maßnahmen, muss das Training wiederholt werden. Dies ist ebenfalls in einer dynamischen Umgebung der Fall, in der die Positionen von beweglichen Hindernissen stark variieren. Wenn ein Training für die Berechnung der Position absolviert werden muss, werden solche Verfahren als „kartenbasiert“ bezeichnet.

Es gibt hybride Verfahren [8, 9, 10], die sowohl Topologie-Daten, als auch absolvierte Messungen zur Berechnung benutzen und somit sowohl geometrisch, als auch kartenbasiert sind.

2.2.4 Online-Phase

In der Online-Phase wird die eigentliche Lokalisierung durchgeführt. Der Algorithmus verwendet sein Vorwissen und die vom Client geschickten Signalinformationen, um den Ort zu berechnen.

Das Benutzerverhalten des Clients hat Einfluss auf die Art der Berechnung. Man unterscheidet zwischen zwei Profilen: Zum einen sind das statische Nutzer (user location problem) und zum anderen mobile Nutzer (user tracking problem). Im ersten Fall wird angenommen, dass sich das mobile Gerät während der Ortung nicht bewegt und man somit davon ausgehen kann, dass alle im Berechnungszeitraum liegenden Signalinformationen für die Ortung gleichermaßen verwendet werden können. Im zweiten Fall nimmt man an, dass sich das mobile Gerät während der Ortung bewegen kann. Bei der Ermittlung des derzeitigen Aufenthaltsortes ist es daher nicht möglich alle zur Verfügung stehenden Signalinformationen gleichrangig zu benutzen, da sie an Positionen aufgenommen wurden, die nicht zwangsläufig der jetzigen Position des Clients entsprechen, sondern lediglich auf dem Weg dorthin liegen.

Sind dem Algorithmus die Positionen der sendenden APs bekannt, so kann er, aufgrund der übermittelten Parameterwerte, die Distanzen des Clients zu ihnen schätzen. Weiteres Wissen über die Umgebung ermöglicht dabei eine genauere Berechnung der Entfernung. Wenn mindestens drei Distanzen bekannt sind, kann er z. B. eine Trilateration durchführen und dadurch einen Punkt in der Ebene abzirkeln. Sind sogar die Entfernungen zu vier Basisstationen bekannt, ist so eine Ermittlung der Position im Raum möglich.

Ist ein absolviertes Training vorhanden, kann der Algorithmus die vom mobilen Gerät geschickten Signalparameter mit den in der Radiomap gespeicherten vergleichen und bestimmt dadurch die Position. Dieser Methode liegt die Annahme zugrunde, dass die Signaleigenschaften an einem bestimmten Punkt immer gleich sind und dass bei einer geringfügigen Entfernung von diesem Punkt, die Signaleigenschaften auch nur geringfügig anders sind.

Aufgrund der Art der Berechnung kann man Lokalisationsverfahren in zwei grundlegende Kategorien unterteilen. Dies sind zum einen die „deterministischen“, zum anderen die „probabilistischen“ Verfahren.

Bei deterministischen Verfahren werden die Signalparameter eines APs, an einem Punkt, durch skalare Werte ausgedrückt, zum Beispiel den Mittelwert aller dort empfangenen Signalstärkewerte. Danach werden deterministische Methoden verwendet, um den Ort eines

Clients zu berechnen. Triangulation, Trilateration und die Suche nach den nächsten Nachbarn sind Beispiele für solche Methoden. Probabilistische Verfahren hingegen verwenden die Wahrscheinlichkeitsverteilung der Signalparameter eines APs, an einem Punkt, und setzen probabilistische Techniken, wie z. B. bedingte Wahrscheinlichkeiten, zur Berechnung des Ortes ein.

In [11] wurde die generelle Arbeitsweise von deterministischen und probabilistischen Verfahren untersucht. Dabei wurde festgestellt, dass die erreichbare Genauigkeit von probabilistischen Verfahren höher ist.

2.3 Ausgewählte Lokalisierungsalgorithmen

Aufgrund der erforderlichen Universalität der zu entwickelnden Evaluierungsumgebung ist eine Recherche verschiedener Algorithmen und ihrer Anforderungen notwendig.

In Kapitel 5 wird die Evaluierungsumgebung getestet. Mit RADAR [1] und WCL [2, 3] werden nun die beiden zur Evaluierung der Umgebung verwendeten Algorithmen genauer vorgestellt.

2.3.1 RADAR

RADAR war eine der ersten Entwicklungen im Bereich der Indoor-Lokalisierung, die auf WLAN basierte. Der mit Signalstärkewerten arbeitende Algorithmus wurde im Jahr 2000 entwickelt. Es wurde sowohl ein kartenbasiertes, als auch ein geometrisches Lokalisierungsverfahren vorgestellt, wobei beide Ansätze in die Klasse der deterministischen Verfahren gehören. Die Entwickler testeten und bewerteten sie in ihren eigenen Räumen. Obwohl die erreichbare Genauigkeit von RADAR relativ gering ist, nehmen viele heute existierende Verfahren Bezug darauf.

Das kartenbasierte Verfahren benötigt in der Offline-Phase ein Training, welches von einer Person mit einem Laptop ausgeführt wird, die an mehreren Trainingspunkten Signalstärke-Messwerte von den APs aufnimmt. Den Entwicklern war bekannt, dass der menschliche Körper ein Hindernis für die Radiowellen darstellt und daher eine Abhängigkeit der Signalstärke von der Orientierung erkennbar ist. An jedem Trainingspunkt werden daher vier Messungen durchgeführt, wobei die Person bei jeder Messung in eine andere Himmelsrichtung blickt. Dies hat den Zweck, jeder Position vier verschiedene Werte zu geben und es so zu ermöglichen, einen an dieser Position befindlichen Client, unabhängig von

seiner Orientierung, zu orten. Die an einem Punkt mit spezieller Orientierung aufgenommenen Werte werden am Ende gemittelt, um die natürlichen Schwankungen der Signalstärke auszugleichen. Somit ist die Radiomap fertig erstellt.

Für das geometrische Verfahren benutzt man in der Offline-Phase die Topologie eines Gebäudes. Man misst die Signalstärkewerte nicht, sondern berechnet sie anhand der Formel 2.1. Dadurch erhält man eine Radiomap, mit gleichem Aufbau (abgesehen von der Orientierung) wie beim kartenbasierten Verfahren, wodurch dieselbe Behandlung in der Online-Phase möglich ist.

$$P(d)[dBm] = P(d_0)[dBm] - 10n \log \left(\frac{d}{d_0} \right) - \begin{cases} nW * WAF & nW < C \\ C * WAF & nW \geq C \end{cases} \quad (2.1)$$

In dieser Formel steht n für die Rate der Abnahme der Signalstärke abhängig von der Entfernung. $P(d_0)$ ist die Signalstärke an einer Referenzdistanz d_0 und d ist die Entfernung zwischen Sender und Empfänger. Der Einfluss der Wände wird durch den WAF (Wall Attenuation Factor) ausgedrückt. C ist die maximale Anzahl der Wände, bei der der WAF noch einen Einfluss hat, wobei nW die Anzahl der Wände zwischen Sender und Empfänger ist.

Zur Lokalisierung in der Online-Phase werden die vom Client geschickten Signalstärkewerte mit denen in der Radiomap verglichen. Dabei wird der ähnlichste Punkt, Nachbar genannt, ermittelt, welcher dann als errechnete Position ausgegeben wird. Für die Bestimmung der Ähnlichkeit wird die Euklidische Distanz (Summe der quadrierten Einzelabweichungen der Signalstärkewerte jedes APs) verwendet. Diese Technik wird NNSS (Nearest Neighbor(s) in Signal Space) genannt. Um die Genauigkeit zu erhöhen, experimentierten die Entwickler mit einigen Parametern. So wurden später mehrere Nachbarn ermittelt und der Mittelwert ihrer Positionen als Ergebnis ausgegeben. Eine weitere Verbesserung erschloss sich durch die Verwendung der maximalen Signalstärke je Messpunkt. Dazu wurden nur die größten Signalstärkewerte aller vier Orientierungen eines Punktes für die Lokalisierung verwendet.

Für die Evaluierung ihres Algorithmus verwendeten die Entwickler das bereits zur Verfügung stehende Training. Dazu entfernten sie einen Trainingspunkt und die dort empfangenen Signalstärkewerte aus der Radiomap und führten anschließend die Ortsberechnung durch. Als Eingabe für den Algorithmus verwendeten sie die Signalstärkewerte des ent-

fernten Trainingspunktes. Sie geben die Genauigkeit ihres Algorithmus durch einen Kreis um den realen Ort des mobilen Gerätes an, in denen die errechneten Positionen mit einer Wahrscheinlichkeit von 25% oder 50% liegen. Beim kartenbasierten Verfahren sind 25% der berechneten Positionen weniger als 1,00 Meter und 50% weniger als 2,13 Meter, von der realen Position entfernt. Beim geometrischen Verfahren befinden sich 25% der berechneten Positionen innerhalb von 1,86 Metern und 50% innerhalb von 4,3 Metern um die reale Position.

2.3.2 WCL

WCL (Weighted Centroid Localization) ist eine Weiterentwicklung von CGLCD (Coarse Grained Localization with Center Determination). CGLCD ist ein sehr simples, grobgranulöses, kartenbasiertes Verfahren für Sensornetzwerke mit geringem Berechnungsaufwand. Es errechnet den Ort des Clients als Mittelwert der Positionen aller APs, die der Client gerade in seinem Sendebereich empfängt. Bei WCL wird dieser Algorithmus um eine Wichtung der AP-Positionen erweitert. Je größer die empfangene Signalstärke ist, desto höher ist auch der Einfluss der Position dieses entsprechenden APs auf das Gesamtergebnis. Dadurch wird der Client näher zu einem AP mit höherer Signalstärke positioniert, was zu einer erhöhten Genauigkeit im Verhältnis zu CGLCD führt.

2.4 Ähnliche Arbeiten

Da es noch keine ähnlichen Evaluierungsumgebungen gibt, die man als Vergleichsobjekte nutzen könnte, dienen hier zwei bereits existierende Lokalisierungsprogramme als Anschauungsmaterial. Im Gegensatz zu der zu entwickelnden Evaluierungsumgebung, die mehrere verschiedene Lokalisierungsalgorithmen unterstützen soll, sind beide hier erwähnten Programme nur auf einen speziellen Algorithmus zugeschnitten. Darüber hinaus ist ihr Hauptziel nicht die Evaluierung, sondern die Lokalisierung. Daher können sie nur als Beispiele für die Benutzeroberfläche, besondere Features und den Aufbau der Komponenten dienen. Die beiden Programme sind zum einen die kommerzielle Ekahau Positioning Engine (kurz: Ekahau) [13] der Firma Ekahau Inc. und zum anderen das frei verfügbare MagicMap [8, 9] der Humboldt-Universität Berlin.

2.4.1 Ekahau

Ekahau ist ein vollständiges Programmpaket bestehend aus einem so genannten Manager mit grafischer Benutzeroberfläche, einem lokalisierbaren Client und einer Positioning Engine. Sie führt die Ortung des Clients durch und gibt die Position an den Manager weiter, der sie dann visualisiert. Alle drei Komponenten können sich auf verschiedenen Rechnern befinden, müssen sie jedoch nicht. Es ist möglich mehrere Clients gleichzeitig zu lokalisieren. Die Hersteller versprechen eine Fehlerdistanz von weniger als fünf Metern, was der Durchschnittsgröße eines Arbeitszimmers entspricht.

Ekahau arbeitet mit RSSI-Werten. Der weitgehend geheim gehaltene Algorithmus befindet sich in der Positioning Engine. Der Client ist minimal aufgebaut und seine einzige Aufgabe ist das periodische Versenden von RSSI-Werten, wobei jeder Client festlegt, welcher Manager ihn lokalisieren kann und welcher nicht. Der Benutzer selbst arbeitet fast ausschließlich nur mit dem Manager.

Ekahau muss in der Offline-Phase ein Training absolvieren. Man lädt zunächst eine Umgebungskarte, wobei die Bildformate bmp, jpg und png möglich sind. Man kann das Pixel-Meter-Verhältnis entweder direkt angeben oder abmessen. Diese Angabe ist für die korrekte metrische Wiedergabe der Positionen der lokalisierten Clients notwendig. Anschließend legt man so genannte „Tracking Rails“ fest, die den begehbaren Bereich auf der Karte festlegen. Danach trainiert man mehrere Punkte in einem Abstand von ca. 3-5 Metern auf diesen Tracking Rails. Das Training eines Punktes benötigt circa 20 Sekunden, wobei man sich während dieser Zeit um die eigene Achse drehen soll, um eine gewisse Unabhängigkeit von der Orientierung des mobilen Gerätes zu erreichen.

Nach dem absolvierten Training ist es nun möglich, Clients zu lokalisieren. Es besteht außerdem die Möglichkeit die Liste, der für die Lokalisierung verwendeten APs, zu bearbeiten. Dies ist insbesondere dann notwendig, wenn APs empfangen werden, die nicht zum eigenen Netzwerk gehören. Aufgrund der fehlenden Kontrolle über diese Basisstationen würde eine unbemerkte Abschaltung oder Umpositionierung zu einer verminderten Lokalisationsleistung des Programms führen.

Um das Problem der Messungenauigkeit durch WLAN-Adapter verschiedener Hersteller zu beheben, wurden mehr als 40 der meist verbreiteten WLAN-Adapter getestet und so ihr Einfluss auf die empfangene Signalstärke aufgezeichnet. Anhand dieser Aufzeichnungen können dann die RSSI-Werte verschiedener Adapter angeglichen werden.

2.4.2 MagicMap

Dieses Programm wird seit einigen Jahren an der Humboldt-Universität in Berlin am Institut für Informatik entwickelt. Aufgrund seines nichtkommerziellen Konzeptes ist das Programm zum Testen und Anwenden frei verfügbar und es sind Informationen über den verwendeten Lokalisierungsalgorithmus vorhanden. Es arbeitet mit RSSI-Werten und es wird eine mittlere Fehlerdistanz von 1-5 Metern angegeben.

Jeder Client berechnet seine eigene Position selbst. Dazu verarbeitet er die empfangenen Signalstärken von APs und anderen Clients sowie kooperativ übermittelte Referenzpunkte. MagicMap benutzt sowohl die Topologie des Netzwerkes, gegeben durch die Positionen von APs, als auch optional eine Radiomap, die durch die übermittelten Referenzpunkte erzeugt wird. Der verwendete Algorithmus ist probabilistisch.

In MagicMap können beliebige Karten vom Format tiff, gif oder jpg geladen werden. Auf diesen Karten ist dann eine Positionierung fester Kommunikationsteilnehmer, wie APs, möglich. Leider lässt sich eine Ortung im Moment nur 2-dimensional durchführen. Signale von Station, die einen gewissen Höhenunterschied zu den restlichen Stationen aufweisen, z. B. weil sie sich auf einer anderen Etage befinden, stören den Lokalisierungsvorgang beträchtlich.

Jeder Benutzer kann im Falle einer unkorrekten Positionierung einzelne Knoten (APs oder Clients) verschieben und so den Lokalisierungsalgorithmus unterstützen. Dieser wird kollektiv von allen im Netzwerk befindlichen Knoten ausgeführt. Er sucht mittels des JUNG-Frameworks [14] immer nach der größtmöglichen „Entspannung“ aller Knoten und ihrer, durch die empfangenen Signalstärkewerte geschätzten, Distanzen zu anderen Knoten. Dabei dienen Referenzpunkte als Hilfestellung. Diese Punkte werden nicht in einer separaten Offline-Phase erstellt, sondern kontinuierlich durch die Benutzer hinzugefügt. Da diese Punkte ausgetauscht werden, stehen jedem Knoten nicht nur die selbst angelegten, sondern alle Referenzpunkte zur Verfügung, sodass sie sogar in Gebieten lokalisierbar sind, an denen sie selbst vorher noch nie waren.

2.5 Einflüsse auf das Ergebnis einer Evaluierung

Die Qualität eines Lokalisierungsalgorithmus wird größtenteils durch seine Ortungsgenauigkeit angegeben. Weniger wichtige Kriterien sind der Rechenaufwand und die Menge des Vorwissens, die dem Algorithmus zur Verfügung gestellt werden muss. Die Ausprä-

gung der Ortungsgenauigkeit ist die Fehlerdistanz. Jeder Algorithmus wurde von den Entwicklern selbst evaluiert, um die Qualität zu ermitteln und einen Vergleich zu anderen Algorithmen zu ermöglichen. Ein aussagekräftiger Vergleich ist jedoch kaum möglich, da die für die Evaluierung nötigen Messungen bei anderen Testbedingungen und Testparametern ausgeführt wurden.

In den folgenden zwei Unterkapiteln wird erläutert, wie abweichende Bedingungen und Parameter die Ergebnisse der Ortsmessungen beeinflussen können. Am Ende folgt ein kurze Diskussion über die Notwendigkeit einer Evaluierungsumgebung.

2.5.1 Einflüsse der Testumgebung

Der von den Entwicklern angegebene Wert der Fehlerdistanz ist nur in der gleichen oder zumindest ähnlichen Testumgebung reproduzierbar. Meistens sinkt in andersartigen Umgebungen die Qualität der Ortung, da das Verfahren im Laufe der Entwicklung und Optimierung auf die Testumgebung zugeschnitten wurde und nur dort optimale Resultate erreicht.

Normalerweise wird in den Gebäuden und Räumen der Entwickler evaluiert, meistens in einer Universität oder einem Forschungslabor. Das sind größtenteils Bürogebäude mit relativ kleinen Räumen, vielen Wänden und einem langen Flur. Die Ergebnisse sind nicht ohne weiteres in andere Gebäudearten, wie z. B. große Lagerhallen, Fabriken oder Privathäuser, übertragbar. Dort sind die Größe und Aufteilung der Zimmer und die Dicke und Beschaffenheit der Wände anders.

Es herrschten während der Messungen besondere, meist undokumentierte Randbedingungen, wie eine bestimmte Luftfeuchtigkeit oder eine gewisse menschliche Aktivität im Versuchsterritorium. Dies ist teilweise abhängig von der Jahres- und Uhrzeit, bei der die Tests durchgeführt wurden.

Einen großen Einfluss auf die Ortungsgenauigkeit haben die örtliche Verteilung und die Anzahl der APs in der Testumgebung. Viele Algorithmen liefern ihre besten Resultate nur, wenn bei jeder möglichen Client-Position eine bestimmte Anzahl von APs empfangbar ist. Auch haben bestimmte Anordnungen der Basisstationen für die Leistung einiger Verfahren negative Auswirkungen. Stehen sie zu dicht zusammen oder befinden sich mehrere auf einer gerade Linie, könnte die Lokalisierung ungenauer werden.

2.5.2 Einflüsse der Testparameter

Jeder Test wird mit bestimmten Testparametern durchgeführt. Jedoch werden die unterschiedlichen Algorithmen nicht mit den gleichen Parametern geprüft, da die Entwickler diese selbst festlegen und diese daher nicht mit den gewählten Parametern anderer Entwickler übereinstimmen müssen. Ein Vergleich von Messungen mit unterschiedlichen Parametern ist schwierig.

Ein wichtiger Punkt in dieser Betrachtung sind die Trainingspunkte. Die Entwickler eines Algorithmus haben die optimale Verteilung, Lage und die Abstände zwischen den einzelnen Trainingspunkten durch Versuche ermittelt und verwenden diese Erkenntnisse bei der Evaluierung. Eine davon abweichende Verteilung kann das Resultat der Messungen verschlechtern. Generell kann man von der Annahme ausgehen, dass eine höhere Dichte von Trainingspunkten zu einem besseren Ergebnis führt.

Des Weiteren spielt die Anzahl der an jedem einzelnen Trainingspunkt aufgenommenen Messwerte eine große Rolle. Je mehr Werte von einem Punkt vorhanden sind, desto mehr Informationen können daraus gewonnen werden. Ist die Anzahl zu gering, leidet die Genauigkeit der Ortung. Dies ist ebenfalls bei der Lokalisierung und Evaluierung so.

Bei einigen Algorithmen gibt es Areale, in denen die berechnete Clientposition nie liegen kann. Genauso gibt es Areale, die mit geringerer oder höherer Wahrscheinlichkeit eine errechnete Clientpositionen enthalten. Darum ist die Lage der Evaluierungspunkte eine weitere nicht zu unterschätzende Tatsache. Eine Position und die dort gemessenen Signalparameter werden als „Evaluierungspunkt“ bezeichnet. Diese Punkte werden für die Messungen einer Evaluierung verwendet und ihre Positionen gehen somit als „reale Positionen“ in die Berechnung der Fehlerdistanz mit ein.

Obwohl eigentlich schon in Kapitel 2.5.1 aufgeführt, können die Anzahl und Verteilung der APs sowie die Uhrzeit und der Wochentag der Messungen, mit in der Kategorie der Testparameter aufgelistet werden. Eine speziell an den Algorithmus angepasste WLAN-Topologie kann das Ergebnis genauso positiv beeinflussen wie die Ausführung der Messungen bei Nacht in einem menschenleeren Gebäude.

2.5.3 Notwendigkeit einer Evaluierungsumgebung

Aufgrund der hier beschriebenen Einflüsse auf das Ergebnis einer Evaluierung ist eine Evaluierungsumgebung sinnvoll. Die Ergebnisse einzelner Tests können so verglichen

werden, da die Testumgebung und die Testparameter gleich sind. Es ist ebenfalls möglich den Algorithmus zu finden, der unter bestimmten Randbedingungen die besten Ergebnisse liefert. Somit kann das optimale Lokalisationsverfahren für einen bestimmten Aufgabenbereich gefunden werden.

3 Konzepte

In diesem Kapitel werden die Konzepte der hier entwickelten Arbeit vorgestellt.

Als Erstes wird erläutert, welche Universalitätsanforderungen die Evaluierungsumgebung erfüllen muss. Anschließend wird der Aufbau der Evaluierungsumgebung behandelt. Dabei wird auf die Trennung der logischen Komponenten, auf die Kommunikation zwischen ihnen und auf die Verwendbarkeit der Umgebung für verschiedene Algorithmen eingegangen. Am Ende folgt eine Beschreibung der angewandten Methoden der Evaluierung. Dabei stehen die Festlegung der Testparameter, der Vergleich zwischen mehreren Algorithmen und die Ermittlung der Evaluierungspunkte im Vordergrund.

3.1 Universalitätsanforderungen

Ein sehr wichtiges Kriterium dieser Evaluierungsumgebung ist, dass Algorithmen trotz ihrer unterschiedlichen Grundvoraussetzungen und Eigenschaften getestet werden können. Deshalb wurden verschiedene Algorithmen auf abweichende Grundvoraussetzungen und Arbeitsweisen hin untersucht. Sie unterscheiden sich durch mehrere Punkte, die nun vorgestellt werden. Die Umsetzung dieser Anforderungen erfolgt in Kapitel 4.1

Es gibt sowohl kartenbasierte, als auch geometrische Verfahren, die eine andere Art des Vorwissens für die Berechnung benötigen. Ebenso unterscheiden sie sich in Menge und Vielfalt der Informationen, die sie für die Berechnung brauchen und demzufolge im Vorwissen und in den übermittelten Signalinformationen enthalten sein müssen. Die Lokalisierung kann mittels mehrerer verschiedener Signalinformationen erfolgen. Des Weiteren muss die Umgebung in der Lage sein, statische und mobile Nutzer zu orten und die Berechnung von deterministischen und probabilistischen Verfahren zu unterstützen. Einige Verfahren benutzen außerdem besondere Features, welche die Evaluierungsumgebung ebenfalls zur Verfügung stellen muss.

3.1.1 Geometrische und kartenbasierte Verfahren

Wie in Kapitel 2.2.3 beschrieben, gibt es zwei verschiedene Arten des Vorwissens. Alle Algorithmen verwenden jeweils eine oder sogar beide Arten für ihre Ortsberechnung. Die Evaluierungsumgebung muss dem Algorithmus dieses Wissen in geeigneter Form zur Verfügung stellen. Es ist daher erforderlich, dass sie ein Training absolvieren und eine Topologie erstellen und verarbeiten kann.

3.1.2 Besondere Daten im Vorwissen

Es gibt mehrere Wege um die Genauigkeit der Lokalisierung zu erhöhen. Einer besteht darin, die vorhandenen Daten durch bessere statistische Methoden auszuwerten. Eine andere Möglichkeit ist, die Menge und Vielfalt der verwertbaren Daten zu erhöhen. Einige Lokalisierungsverfahren gehen dabei den zweiten Weg, indem sie zusätzliche Informationen benutzen, die von der Topologie oder dem Training zur Verfügung gestellt werden müssen.

Daten der Topologie: Neben den Positionen von APs können in der Topologie auch zusätzliche Daten, wie die Sendeleistung und das verwendete WLAN-Protokoll (siehe Kapitel 2.1.4) notiert sein. Beide Angaben haben einen Einfluss auf die maximale Reichweite des WLANs.

Die Signalausbreitung wird in Gebäuden maßgeblich durch Wände und andere große stationäre Hindernisse beeinflusst. Einige Algorithmen, wie z. B. RADAR [1] (Kapitel 2.3.1), beziehen Hindernisinformationen in die Berechnung mit ein, was ein Einfügen dieser Daten in die Topologie sinnvoll und notwendig macht.

Um das Ergebnis einer Positionierung auf seine logische Korrektheit hin zu überprüfen, ist eine Angabe des „begehbaren Bereiches“ nötig. Dieser Bereich kennzeichnet Gebiete, in denen sich mobile Geräte befinden können. Dadurch wird eine Anzeige des Clients in einer Wand oder an einem anderweitig blockierten Ort verhindert. Eine Möglichkeit dies zu tun, sind so genannte Tracks, wie sie bei Ekahau [13] (Kapitel 2.4.1) verwendet werden. Eine andere Möglichkeit ist die Auswertung der in der Umgebungskarte eingezeichneten Hindernisse. Ebenfalls können die Maße der angezeigten Umgebungskarte hilfreich sein, um so die Grenzen möglicher Clientpositionen festzulegen.

Daten des Trainings: Die im Training erstellte Radiomap kann, neben den immer gespeicherten Werten für Ort und Signaleigenschaften, auch eine Orientierung enthalten, um differenzierbare Werte für jeden Messpunkt bereitzustellen. Dies ist zum Beispiel bei RADAR und [15] der Fall. Andere Algorithmen [16] verfolgen den Ansatz, dass sich die Signaleigenschaften abhängig vom Wochentag und der Tageszeit verändern. Im Training und während Lokalisation ist daher eine Angabe des Datums und der Uhrzeit notwendig.

3.1.3 Unterschiedliche Signalinformationen

Wie in Kapitel 2.2.1 beschrieben, gibt es mehrere Arten der verwendbaren Signalinformationen. Um die Universalität zu gewährleisten, muss die Evaluierungsumgebung in der Lage sein, jede mögliche Art der Signalinformationen zu speichern und zu verarbeiten.

3.1.4 Statisches und mobiles Nutzerprofil

Die in Kapitel 2.2.4 beschriebenen Nutzerprofile haben Auswirkungen auf die Art und Weise, wie die Berechnung des derzeitigen Ortes erfolgt. Im Gegensatz zu statischen Nutzern, muss bei mobilen Nutzern eine Wichtung der empfangenen Signalinformationen erfolgen. Außerdem kann es nötig sein, eine Geschwindigkeit des mobilen Gerätes zu ermitteln, um damit die momentane Position schätzen zu können. Eine Wichtung der Signalinformationen und eine Geschwindigkeitsapproximation sind nur dann möglich, wenn alle empfangenen Daten mit einer Uhrzeit versehen sind. Weitere Anpassungen sind für den Lokalisierungsteil der Evaluierungsumgebung nicht nötig, da die Behandlung des Nutzerprofils Aufgabe des verwendeten Algorithmus ist. Bei dem Evaluierungsteil ist jedoch sicher zu stellen, dass beide Arten des Nutzerprofils evaluiert werden können.

3.1.5 Deterministische und probabilistische Verfahren

Die Darstellung und Auswertung der Daten, die vom mobilen Gerät geschickt werden, können auf verschiedene Arten erfolgen (siehe Kapitel 2.2.4). Es gibt deterministische Verfahren, die skalare Werte benutzen und probabilistische Verfahren, welche die empfangenen Werte in eine Wahrscheinlichkeitsverteilung umwandeln. Beide Arten benötigen die gleichen Daten, da sie diese nur anders verarbeiten. Daher werden durch diese Unterscheidung keine besonderen Anforderungen an die Evaluierungsumgebung gestellt.

3.1.6 Besondere Features der Evaluierungsumgebung

Um eine gleich bleibend hohe Genauigkeit der Ortung zu gewährleisten, arbeiten manche Verfahren [8, 9, 10] mit einer Benutzer-Korrektur. Benutzer können hier ihre falsche Position berichtigen, wobei der Algorithmus diese Korrektur in die weiteren Berechnungen mit einbezieht. Dabei werden für die Berechnung der Position eines Clients seine eigenen Korrekturpunkte bevorzugt verwendet. Die Evaluierungsumgebung muss eine Benutzer-Korrektur ermöglichen und alle Daten, nicht nur die Korrekturpunkte, mittels des Clientnamens identifizierbar machen.

3.2 Aufbau der Evaluierungsumgebung

Für die Lokalisierung eines mobilen Gerätes benötigt man eigentlich nur das mobile Gerät selbst. Es sind keine weiteren Komponenten - von der Infrastruktur des WLANs abgesehen - notwendig, da alle nötigen Schritte zur Lokalisierung auf einem einzigen Rechner ausgeführt werden können. Dieser Aufbau ist jedoch aus mehreren Gründen für eine Evaluierungsumgebung sehr unpraktisch. Darum wird nun folgend beschrieben, wie ein geeigneter Komponentenaufbau entwickelt wurde.

3.2.1 Logische Trennung der Komponenten

Um ein mobiles Gerät zu orten, müssen von einem Lokalisierungssystem mehrere Arbeitsschritte ausgeführt werden. Diese sind:

- Bereitstellung des Vorwissens durch Erstellen einer Topologie oder Absolvieren eines Trainings
- Speicherung des Vorwissens
- Messen der ortsspezifischen Signalinformationen
- Berechnung der Position
- Verarbeitung und Anzeige der Clientposition

Bei einem Evaluierungssystem ergeben sich darüber hinaus weitere Arbeitsschritte:

- Vergleich der realen und errechneten Position
- Speicherung der Ergebnisse zur späteren Auswertung

Für den effizienten Ablauf der Lokalisierung und Evaluierung ist die richtige Verteilung dieser Aufgaben, auf eine oder mehrere Komponenten, entscheidend. Dabei muss die durch die Aufteilung entstehende Struktur einigen Anforderungen genügen.

Wie der Name schon sagt, soll die hier entwickelte Umgebung die Evaluierung erleichtern. Darum soll es einfach sein, die Testparameter zu ändern und die Resultate eines Tests auszuwerten. Aus diesem Grund ist eine zentrale Speicherung aller für die Evaluierung wichtigen Daten notwendig. In einem realen Szenario ist meistens mehr als nur ein mobiles Gerät zu orten. Manche Verfahren, wie zum Beispiel MagicMap [8, 9] (Kapitel 2.4.2) benutzen die errechneten Orte anderer Clients für die Lokalisierung. Die Struktur soll es daher zulassen, dass mehrere mobile Geräte gleichzeitig lokalisiert und grafisch dargestellt werden können. Die Anforderungen an die mobilen Geräte sollen möglichst niedrig sein, damit auch die Möglichkeit besteht, Geräte mit geringen Systemressourcen zu verwenden, wie es in der Aufgabenstellung (Kapitel 1.2) gefordert wird. Dadurch lassen sich kostengünstige Tests mit einer hohen Anzahl von Geräten durchführen. Ebenfalls ist es vorteilhaft, wenn das mobile Gerät so wenig wie möglich Kommunikation durchführen muss. Diese könnte im Verlauf der Messung der Signaleigenschaften gestört werden. Das kann, z. B. beim Wechseln des Kanals (siehe Kapitel 2.1.2) passieren.

Es ergibt sich folgende Liste der gewünschten Eigenschaften (geordnet nach Wichtigkeit):

- Zentrale Speicherung aller wichtigen Daten: Topologie, Training, Ergebnisse der Evaluierung
- Gleichzeitige Lokalisierung und grafische Darstellung mehrerer mobiler Geräte
- Mobiles Gerät soll mit möglichst geringen Systemressourcen auskommen
- So wenig wie möglich Kommunikation mit dem mobilen Gerät, um Störungen zu vermeiden

Im Folgenden werden verschiedene Modelle für eine Lösung bezüglich dieser Anforderungen beschrieben. Eine kurze Auswertung der vorgestellten Modelle erfolgt in Tabelle 3.1.

Modell ohne logische Komponententrennung

Alle für die Lokalisierung notwendigen Arbeitsschritte werden auf einem Rechner erledigt. Das Vorwissen, die errechnete Position und die Ergebnisse der Evaluierung werden bei dem mobilen Gerät selbst gespeichert.

Weil dem mobilen Gerät ausschließlich seine eigenen Daten zur Verfügung stehen, kann er nur seine eigene Position berechnen und grafisch darstellen. Alle Berechnungen, darunter auch die teilweise anspruchsvolle Ermittlung der Clientposition, werden auf einem einzigen Gerät ausgeführt. Das ist nicht auf einem Gerät mit stark begrenzten Systemressourcen, wie z. B. einem PDA (Personal Digital Assistant), möglich. Außerdem ist die separate Speicherung sehr unpraktisch, weil die Ergebnisse der Berechnung danach einzeln zusammengesammelt werden müssen. Eine Veränderung der Trainingsdaten, der Topologie oder des verwendeten Algorithmus, zu Versuchszwecken, ist sehr zeitaufwendig, da dies auf jedem mobilen Gerät einzeln durchgeführt werden muss. Ein Vorteil dieser Variante ist, dass keine Kommunikation mit anderen Rechnern notwendig ist, um die Lokalisierung durchzuführen.

Es ist also eine bessere Lösung zu finden, die sich durch die Beseitigung der größten Mankos dieser Variante, die dezentrale Speicherung und die fehlende Behandlung mehrerer mobiler Geräte, auszeichnet.

Client-Server-Modell

Bei diesem Modell erfolgt eine Trennung der Aufgaben. Der Client erstellt das Vorwissen und schickt es zur zentralen Speicherung zum Server. Dieser führt auch die Berechnung des Ortes, anhand der vom Client gemessenen Signalparameter, aus und schickt die Position zur Darstellung zurück an den Client. Das Messen der Fehlerabweichung erfolgt beim Client, der das Resultat wiederum zum Server schickt (siehe Abbildung 3.1).

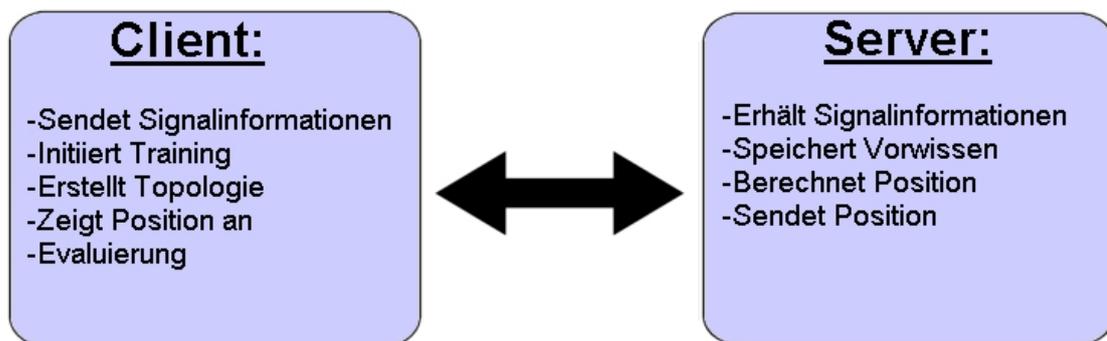


Abbildung 3.1: Aufgaben der Komponenten im Client-Server-Modell

Durch die Aufteilung der Arbeitsschritte auf zwei Komponenten werden viele der Nachteile des vorherigen Konzeptes beseitigt. Der größte Vorteil ist die zentrale Speicherung

der wichtigen Daten. Dadurch kann eine Veränderung des Datenbestandes und eine Auswertung der Ergebnisse leicht durchgeführt werden. Die Berechnung des Ortes erfolgt zentral beim Server, sodass es möglich ist, mehrere Clients gleichzeitig zu lokalisieren. Die Positionen können an andere Clients weiter gegeben werden, wodurch diese mehr als nur ihre eigene Position anzeigen können. Ein Nachteil ist, dass der Client neben der Messung und Übermittlung der Signalinformationen noch weitere essenzielle Kommunikation durchführen muss, die gestört werden könnte. Die rechenintensivsten Arbeiten werden nun vom Server durchgeführt. Aufgrund der Trennung der Aufgaben kann sich der Server auf einem anderen Rechner befinden, welcher im Gegensatz zum Client, nicht mobil sein muss. Es besteht daher die Möglichkeit einen leistungsstarken Computer dafür zu benutzen, der selbst komplexe Lokalisierungsalgorithmen in kurzer Zeit bearbeiten kann. Der Client wird durch die Aufteilung der Arbeit deutlich entlastet, wodurch das Ziel, auch Geräte mit geringeren Systemvoraussetzungen verwenden zu können, näher gerückt ist. Jedoch ist dieses Ziel noch nicht endgültig erfüllt, da dem Client noch die Aufgaben der Erstellung des Vorwissens, die Darstellung der Position und das Messen der Fehlerabweichung, obliegen.

Dieses Modell ist deutlich besser als die vorige Variante. Es zeichnet sich durch die gewünschte zentrale Speicherung, eine Entlastung des zu ortenden mobilen Gerätes und die Möglichkeit, mehrere Clients zu lokalisieren, aus. Es ist jedoch noch nicht optimal. Es ist eine bessere Lösung zu finden, die den Client noch mehr entlastet.

minimaler Client-Server-Modell

Dieses Modell beschränkt den Aufgabenbereich des Clients auf das Allernötigste und zwar auf das Messen und Versenden der Signalinformationen. Alle anderen Arbeitsschritte werden durch den Server erledigt (siehe Abbildung 3.2).

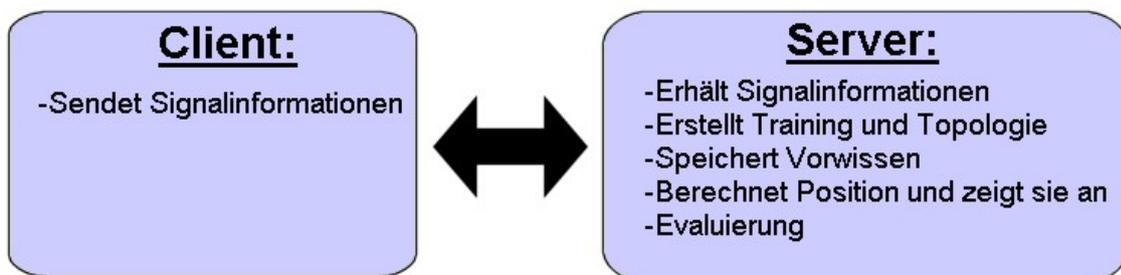


Abbildung 3.2: Aufgaben der Komponenten im minimaler Client-Server-Modell

Dieses Modell hat die gleichen positiven Aspekte, wie das Client-Server-Modell. Es hat eine zentrale Speicherung und es ist möglich, mehrere Clients gleichzeitig zu lokalisieren und auf dem Server grafisch darzustellen. Als Verbesserung zu dem vorigen Modell ist zu nennen, dass der Client dabei nicht unnötig belastet wird. Dadurch ist es möglich, Clients mit geringen Systemressourcen einzusetzen. Ebenfalls muss der Client keine weitere Kommunikation durchführen, die durch die Messung gestört werden könnte.

Somit erfüllt dieses Modell alle eingangs aufgelisteten Anforderungen.

minimaler Client-Manager-Server-Modell

Das in Kapitel 2.4.1 beschriebene EkaHau [13] hat eine geringfügig andere Struktur. Dort wird die Arbeit, die beim minimaler Client-Server-Modell der Server übernimmt, auf zwei Komponenten aufgeteilt. Eine Komponente ist die „Positioning Engine“, welche die Berechnung der Clientposition vornimmt und das Vorwissen speichert. Die andere ist der „Manager“, der das Vorwissen erstellt, lokalisierte Clients grafisch darstellt und die Evaluierung durchführen kann. Der Manager dient zur Steuerung des Servers (siehe Abbildung 3.3).

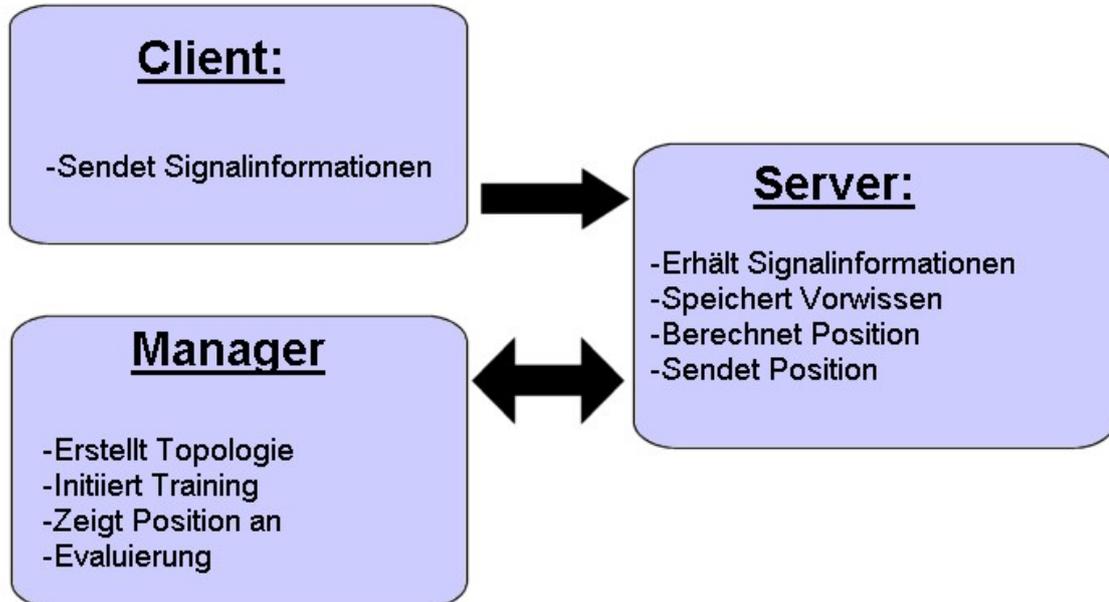


Abbildung 3.3: Komponentenaufgaben im minimaler Client-Manager-Server-Modell

Dieses Modell hat die gleichen positiven Aspekte wie das vorige Modell und erfüllt so-

mit alle verlangten Anforderungen. Jedoch weißt diese Lösung einen weiteren positiven Aspekt auf: Beim minimaler Client-Server-Modell kann der Server ein leistungsstarker, immobiler Computer sein, was bei einigen anspruchsvollen Lokalisierungsalgorithmen auch notwendig werden kann. Der Benutzer des Lokalisierungsverfahrens wäre somit an diesen Rechner gebunden, da er nur dort die nötigen Schritte für die Vorwissenserstellung, Lokalisierung und Evaluierung abarbeiten könnte. Dies ist beim Komponentenmodell von Ekahau nicht so. Hier ist die Benutzerschnittstelle von der eigentlichen Berechnung der Lokalisierung getrennt. Dadurch ist es weiterhin möglich, dass sich der aufwendige Lokalisierungsalgorithmus auf einem immobilen Rechner befindet. Der Benutzer ist jedoch nicht mehr dazu gezwungen, an diesem Rechner zu arbeiten. Er kann sich nun mit einem weniger leistungsstarken Rechner, z. B. einem Laptop, frei bewegen. Von dort aus kann er den Server mittels des Managers steuern und Daten (z. B. Positionen von lokalisierten Clients) von ihm empfangen.

Bewertung der Modelle der Komponententrennung

In diesem Kapitel wurden insgesamt vier mögliche Modelle vorgestellt. Von diesen erfüllen nur das minimaler Client-Server-Modell und das minimaler Client-Manager-Server-Modell die gewünschten Anforderungen. Da jedoch Letzteres den Vorteil hat, dass der Nutzer nicht ortsgebunden sein muss, wird dieser Variante der Vorzug gegeben. Die genauen Leistungsmerkmale der einzelnen Varianten sind in Tabelle 3.1 dargestellt.

Variante	Zentrale Speicherung	Mehrere Clients	Geringe Ressourcen	Wenig Kommunikation	Örtlich ungebunden
Ohne Komponententrennung	Nein	Nein	Nein	Ja	unbetrachtet
Client-Server	Ja	Ja	Nein	Nein	unbetrachtet
Min. Client-Server	Ja	Ja	Ja	Ja	Nein
Min. Client-Manager-Server	Ja	Ja	Ja	Ja	Ja

Tabelle 3.1: Bewertung der einzelnen Varianten der Komponententrennung

3.2.2 Kommunikation zwischen den Komponenten

Die drei Komponenten der Evaluierungsumgebung müssen miteinander kommunizieren, um die Ausführung aller Arbeitsschritte zu ermöglichen. Die Kommunikation zwischen den einzelnen Komponenten ist dabei unterschiedlich strukturiert.

Manager-Server-Kommunikation

Bei dem Inhalt der Pakete, die zwischen dem Manager und dem Server ausgetauscht werden, handelt es sich größtenteils um Befehle, die nicht verloren gehen dürfen. Daher ist hierfür ein zuverlässiges Transportprotokoll zwingend erforderlich, das eine Auslieferung der Pakete garantieren kann. Dies ist nur bei TCP (Transmission Control Protocol) der Fall.

Client-Server-Kommunikation

Bei der Kommunikation zwischen Client und Server sind die Anforderungen anders. Der Client schickt Pakete mit Signalinformationen. Der Server hingegen muss keine Pakete an den Client senden, womit der Datenfluss unidirektional ist. Da diese Pakete keine essenziellen Befehle enthalten, hat der Verlust einzelner Pakete keine gravierenden Folgen. Eine Garantie der Übermittlung ist demzufolge nicht nötig, jedoch wünschenswert. Da TCP diese Garantie bietet, ist es UDP (User Datagram Protocol) in dieser Hinsicht überlegen. Es gibt aber noch weitere Kriterien, die nun folgend betrachtet werden. Dabei wird jeweils die Eignung von TCP und UDP besprochen.

Verlust von Bestätigungen: Bei den Messungen der Signaleigenschaften muss der Client wiederholt den Kanal wechseln (siehe Kapitel 2.1.2). Während dieser Zeit ist er von dem AP, an dem er assoziiert ist, nicht erreichbar. TCP verwendet Bestätigungen der empfangenen Pakete, um die garantierte Versendung sicher zu stellen. Die Bestätigungen des Servers können den Client während dieser Zeit nicht erreichen und gehen dadurch verloren. Dies würde dazu führen, dass der Client Pakete wiederholt verschicken muss, obwohl diese bereits korrekt angekommen sind. Eine Möglichkeit dieses Problem zu umgehen, bietet die Powermanagementfunktion der WLAN-Karte. Dem AP wird mitgeteilt, dass der Client während des Messvorgangs im Stromsparmmodus ist. Somit wird ein Senden von Bestätigungen so lange verzögert, bis dieser seine Messungen beendet hat. Diese Vorgehensweise ist jedoch schwierig zu verwirklichen, da diese Funktionalitäten von der Firmware der WLAN-Karte bereitgestellt werden. Es ist eine aufwendige Änderung an der Firmware aller verwendeten WLAN-Adapter notwendig. Eine weitere leichter umzusetzende Möglichkeit ist die Einführung einer Kanalwechselfase. Nachdem der Client das Paket mit den Signalinformationen an den Server geschickt hat wartet er eine bestimmte Zeit. Dies ermöglicht sowohl das Empfangen einer Bestätigung als auch die Durchfüh-

nung weiterer Kommunikation. Der Nachteil dieser Methode ist, dass dadurch weniger Zeit für Messungen zur Verfügung steht. Bei UDP werden die empfangenen Pakete nicht bestätigt, was zu dem Verlust von Paketen führen kann. Für die Benutzung von UDP sind jedoch keine Veränderung der Firmware und keine Einführung einer Kanalwechselfpause notwendig. Dadurch besteht die Möglichkeit, kontinuierlich Messungen durchzuführen. Die einzige Unterbrechung stellt das Senden der Signalinformationen dar.

Verzögerung durch Verlust von Paketen: TCP gibt die Pakete nur in der gleichen Reihenfolge an die Applikation weiter, in der sie auch abgeschickt wurden. Geht ein Datenpaket verloren, werden alle danach gesendeten Pakete zwischengespeichert und der Client wird darüber informiert. Erst wenn das fehlende Paket angekommen ist, werden dieses und die gespeicherten Pakete an die Applikation (in diesem Fall dem Server) weitergegeben. Obwohl die Reihenfolge wichtig ist, ist diese Verzögerung unerwünscht, weil die Aktualität der Daten von großer Bedeutung ist. Um zu gewährleisten, dass der Server die korrekte Empfangszeit des Paketes benutzt, muss der Client neben den gemessenen Signalparametern auch einen Zeitstempel mitschicken. Da dieser Zeitstempel für die weiteren Berechnungen verwendet wird, ist eine Uhrensynchronisation aller Clients, des Servers und des Managers erforderlich. Ein Zeitstempel und eine aufwendige Uhrensynchronisation sind bei der Verwendung von UDP nicht nötig, da hier Pakete nicht absichtlich zwischengespeichert werden. Es kann jedoch dadurch zu einer Umsortierung der Pakete kommen, wenn mehrere verschiedene Routen zum Ziel existieren.

Bewertung der Alternativen: Bei der Kommunikation zwischen Client und Server ist die Benutzung von TCP aufgrund dieser Fakten angebracht. Es bietet gegenüber UDP einige Vorteile. Andererseits ist die Benutzung von TCP sehr aufwendig: Es muss eine Lösung für das Kanalwechselproblem gefunden und eine ständige Uhrensynchronisierung durchgeführt werden. UDP ist zwar nicht völlig ungeeignet für diese Aufgabe, jedoch bietet TCP mehr Vorteile. Im Gegenzug sind für UDP keine weiteren Implementierungen notwendig. In Tabelle 3.2 ist eine Zusammenfassung der Eigenschaften dargestellt.

Variante	Paketverlust ausgeschlossen	Korrekte Reihenfolge garantiert	Kontinuierliche Messungen möglich	Keine weiteren Arbeiten nötig
TCP	Ja	Ja	Nein	Nein
UDP	Nein	Nein	Ja	Ja

Tabelle 3.2: Eignung der Transportprotokolle für Client-Server-Kommunikation

3.2.3 Austauschbarkeit der Algorithmen

Die in Kapitel 3.2.1 entwickelte Komponententrennung der Evaluierungsumgebung ermöglicht es, die Berechnung des Ortes zentral beim Server auszuführen. Dort wird das Vorwissen gespeichert und die vom Client geschickten Signalparameter werden vom Lokalisationsalgorithmus verarbeitet, um eine Position zu berechnen. Um mehrere Lokalisierungsverfahren zu benutzen, ist es notwendig, diesen Algorithmus austauschen zu können. Diese Anforderung führt dazu, dass der Aufbau des Servers vom minimaler Client-Manager-Server-Modell näher betrachtet werden muss.

Die Aufgaben des Servers sind die Speicherung des Vorwissens, die Berechnung der Position und die Kommunikation mit dem Manager und dem Client. Um den Algorithmus als austauschbares Modul behandeln zu können, werden diese Aufgaben in zwei Teilkomponenten gegliedert. Der so genannte *EngineHolder* übernimmt alle Aufgaben - außer der Bestimmung der Position - welche vom *Algorithm* erledigt wird. Der *Algorithm* stellt die Funktionalität des Lokalisierungsverfahrens dar. Der *EngineHolder* speichert alle nötigen Daten und kommuniziert mit dem Manager und dem Client. Er reicht die für den *Algorithm* wichtigen Daten an diesen weiter und übernimmt die von ihm errechneten Positionen und sendet sie zur grafischen Darstellung an den Manager. Auf diese Weise kann der Lokalisierungsalgorithmus ohne Veränderung des *EngineHolder*s ausgetauscht werden. Der Aufbau ist in Abbildung 3.4 ersichtlich.

Bei den vom Client gesendeten Daten ist zu beachten, dass diese nur für den *Algorithm* bestimmt sind. Der *EngineHolder* ist nicht dazu in der Lage, sie auszuwerten oder zu verarbeiten (siehe Kapitel 3.1.3), da ihr Datenformat von der Implementierung des Lokalisierungsalgorithmus und der verwendeten Art der Signaleigenschaften abhängt und somit variabel ist. Es ist dennoch angebracht, dass der Client Pakete mit einem einheitlichen und vielseitig verwendbaren Format verschickt, damit verschiedene Algorithmen diese Daten verwenden können. Somit haben Client und *Algorithm* einen logischen Zusammenhang. Gegebenenfalls sind beide aneinander anzupassen.

3.3 Evaluierung von Lokalisierungsalgorithmen

Die Bewertung der Leistungsfähigkeit von Lokalisierungsalgorithmen ist das Hauptziel dieser hier entwickelten Evaluierungsumgebung. Die Beurteilung der Güte erfolgt maßgeblich durch die Genauigkeit des Verfahrens. Je näher die errechnete Position an der rea-

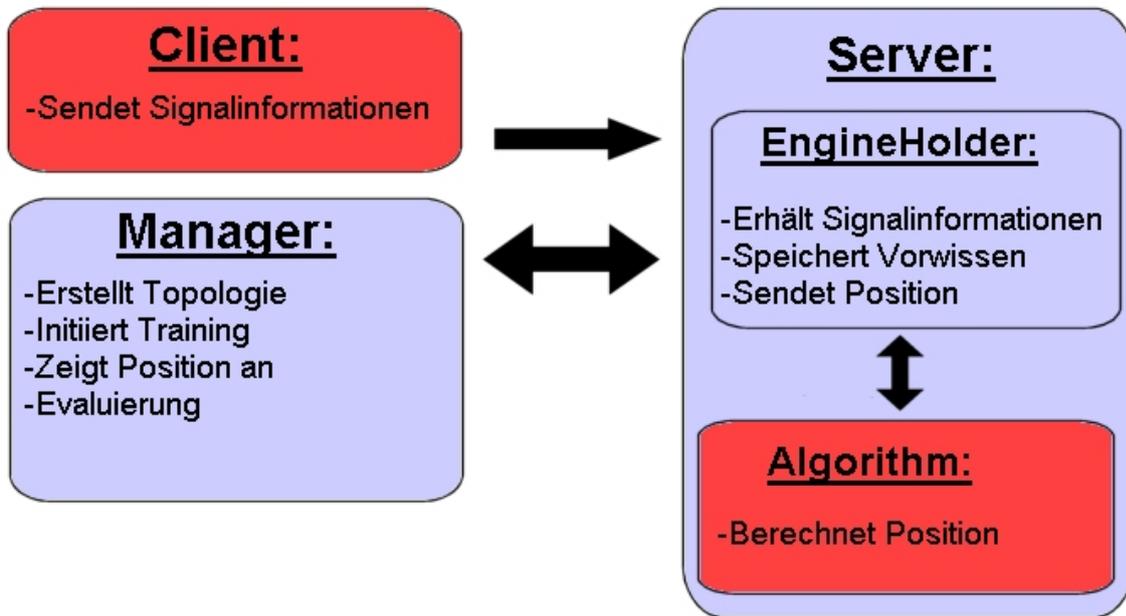


Abbildung 3.4: gekapselter Aufbau des Servers

len Position ist, desto besser ist das Verfahren. Es gibt zwei Möglichkeiten das Maß der Genauigkeit anzugeben. Entweder durch den Mittelwert aller Fehlerabweichungen oder durch die Angabe eines Radius, mit der Bedingung, dass ein bestimmter Prozentsatz aller errechneten Positionen innerhalb dieses Kreises - mit der realen Position als Mittelpunkt - liegt. Für beide Arten der Darstellung benötigt man eine Liste mit realen und errechneten Positionen. Dies soll durch die Evaluierungsumgebung zur Verfügung gestellt werden. Nun folgend werden die verschiedenen Testparameter beschrieben. Der Vergleich mehrerer Lokalisierungsalgorithmen folgt danach. Den Abschluss bildet die Fragestellung, wie die Punkte, an denen die Lokalisierungsleistung getestet wird, ermittelt werden.

3.3.1 Festlegung und Variation der Testparameter

Um eine Evaluierung durchzuführen, müssen vorher mehrere Testparameter festgelegt werden. Einige Parameter können zu Versuchszwecken verändert werden, um die Abhängigkeiten zu beobachten.

Mögliche Parameter sind:

- Anzahl, Dichte und Verteilung der Trainingspunkte
- Menge der Signalinformationen je Trainingspunkt bzw. Evaluierungspunkt

- Abstand der Evaluierungspunkte von den Trainingspunkten
- Anzahl, Dichte und Verteilung der APs
- Nutzerprofil: statisch oder mobil
- Besonderheiten: Uhrzeit, Wochentag, Hardware

Es besteht eine Abhängigkeit der Ortungsgenauigkeit von der Anzahl, Dichte und Verteilung der Trainingspunkte und von der Menge der Signalinformationen je Trainingspunkt bzw. Evaluierungspunkt. Um die Auswirkungen einer Veränderung dieser Parameter testen zu können, ermöglicht es die Evaluierungsumgebung, wahlweise einige Trainingspunkte für die Ortung zu deaktivieren und die Menge der Signalinformationen, die für das Training und die Lokalisierung verwendet werden, zu bestimmen. Eine ähnliche Abhängigkeit existiert bezüglich der Anzahl der APs, die für die Lokalisierung verwendet werden. Um diese Anzahl verändern zu können, stellt die Evaluierungsumgebung ein Feature bereit, welches es ermöglicht, nur bestimmte APs für die Berechnung des Ortes zuzulassen. Die Eliminierung von Einflüssen von APs, die nicht unter der eigenen Kontrolle stehen, ist ein weiterer Grund, dass dieses Feature implementiert wurde.

3.3.2 Vergleich mehrerer Algorithmen

Wenn man die Leistungen mehrerer Lokalisierungsverfahren gegenüberstellen möchte, müssen diese unter gleichen Bedingungen überprüft werden. Nur so ist es möglich, die Ergebnisse der Evaluierung einzelner Verfahren zu vergleichen und verwertbare Rückschlüsse daraus zu ziehen. Jetzt wird erläutert, welche Möglichkeiten es dazu gibt. Das Ergebnis der Diskussion wird in Tabelle 3.3 präsentiert.

Online-Evaluierung

Der Name dieser Evaluierungsmethode ergibt sich aus dem Umstand, dass die Messungen, Berechnungen und Bewertungen gleichzeitig erfolgen. Ein mobiles Gerät, welches sich an einer frei wählbaren Position befindet, wird mittels eines Lokalisierungsalgorithmus geortet und die Fehlerdistanz kann durch den Nutzer sofort angegeben werden. Durch eine Umpositionierung ist dieser Vorgang beliebig oft wiederholbar. Diese Art der Evaluierung kann man nun anschließend mit den anderen zu testenden Algorithmen durchführen.

Die Messungen können in der gleichen Testumgebung und unter den gleichen Testparametern gemacht werden. Dies führt dazu, dass die Ergebnisse der Messungen vergleichbar

sind. Jedoch unterliegen die Messwerte bestimmten Schwankungen, die trotz einer konstanten Testumgebung und konstanten Testparametern auftreten. So würden bei mehreren aufeinander folgenden Evaluierungen eines Algorithmus, die Messwerte geringfügig voneinander abweichen. Die Ergebnisse sind also nicht reproduzierbar. Der Grund dafür liegt in der in Kapitel 2.2.2 beschriebenen Variation der Signalparameter. Das mit einem Training zur Verfügung gestellte Vorwissen und die verwendeten Evaluierungspunkte sind nur Momentaufnahmen der Signaleigenschaften. Selbst Messungen, zwischen denen nur wenige Sekunden liegen, werden höchstwahrscheinlich unterschiedliche Werte aufweisen. Je weiter die Messungen zeitlich auseinander liegen, desto weniger Aussagekraft hat ein Vergleich dieser Messungen.

Bei dieser Methode der Evaluierung muss eine Person mit einem lokalisierbaren Gerät wiederholt Messungen durchführen und das errechnete Ergebnis bewerten. Sollen einige der in Kapitel 3.3.1 erwähnten Parameter verändert werden, so sind wiederum Messungen notwendig. Das ist extrem zeitaufwendig, hat jedoch auch Vorteile. Durch die direkte Kopplung der Person mit der Evaluierung kann sie ein Gefühl für die Stärken und Schwächen des gerade benutzten Verfahrens entwickeln. Dies ist besonders bei einem mobilen Nutzerprofil hilfreich, da die Evaluierungspunkte spontan bestimmt werden können. Diese Spontaneität hat jedoch auch Nachteile. Sie erleichtert zwar die Bewertung eines einzelnen Verfahrens, jedoch bedeutet dies auch, dass jedes Verfahren an anderen Evaluierungspunkten getestet wird. Dadurch sind die Resultate mehrerer Tests weniger vergleichbar. Einige Nachteile kann man beseitigen, indem man allen zu testenden Verfahren dasselbe Vorwissen bereitstellt. Es werden also nur ein Training und eine Topologie verwendet. Die Schwierigkeit besteht darin, dass in diesem Vorwissen alle Daten enthalten sein müssen, welche die unterschiedlichen Verfahren benötigen (Kapitel 3.1.2). Der Einfluss der Varianz von Signalinformationen ist somit nur noch bei den Evaluierungspunkten vorhanden. Der Arbeitsaufwand wird dadurch ebenfalls verringert.

Offline-Evaluierung

Im Gegensatz zur Online-Evaluierung müssen während der Berechnung des Ortes und der Bestimmung der Fehlerabweichung hier keine Messungen durchgeführt werden. Alle nötigen Messungen werden vorher absolviert und gespeichert. Das heißt, das Training und die Evaluierungspunkte werden - unabhängig vom verwendeten Algorithmus - vorher ermittelt. Aus diesem Grund werden die Evaluierungspunkte „Offline-Evaluierungspunkte“

genannt. Wie genau diese Punkte für die Offline-Evaluierung ermittelt werden, ist in Kapitel 3.3.3 beschrieben. Während der Berechnungen für die Evaluierung greift der Lokalisierungsalgorithmus, anstatt auf gerade gesendete Signalparameter, auf die gespeicherten Daten zurück. So ist es möglich, die Berechnung von verschiedenen Algorithmen ausführen zu lassen.

Diese Methode benutzt dasselbe Vorwissen und dieselben Evaluierungspunkte für jeden Algorithmus. Dadurch werden die Variationen der Signalinformationen komplett eliminiert und ein exakt eindeutiger Vergleich zwischen mehreren Verfahren ist möglich. Das Ergebnis einer Berechnung ist reproduzierbar, da alle Eingaben konstant sind und das Resultat demzufolge ausschließlich vom verwendeten Lokalisierungsalgorithmus abhängt. Diese Variante ist sehr zeitsparend, da der Aufwand für die Messungen von der Anzahl der zu testenden Verfahren unabhängig ist. Des Weiteren ist positiv zu bemerken, dass das Ändern einiger Testparameter sehr einfach ist. So kann z. B. durch eine Veränderung der Trainingsdaten eine andere Dichte der Trainingspunkte für die Tests verwendet werden. Man kann auch beliebige Zeit später erneut Algorithmen mit diesen Daten evaluieren, ohne dabei Abstriche bei der Vergleichbarkeit mit bereits geprüften Algorithmen zu machen.

Bewertung der Evaluierungstechniken

Es ergeben sich also zwei Möglichkeiten, wie die Messungen für eine Evaluierung durchgeführt werden können. Für eine kurze Übersicht der Vor- und Nachteile der beiden Evaluierungsarten dient Tabelle 3.3.

Variante	Gleiche Testparameter	Gleiche Messwerte	Ergebnisse reproduzierbar	Direkte Behandlung
Online	Ja	Nein	Nein	Ja
Offline	Ja	Ja	Ja	Nein

Tabelle 3.3: Leistungsmerkmale der Evaluierungstechniken

Sowohl bei der einen, als auch bei der anderen Art können anschließend die Ergebnisse miteinander verglichen werden, da die Messungen unter gleichen Bedingungen absolviert wurden. Die Offline-Evaluierung sticht jedoch mit mehreren Vorteilen heraus. Es werden die gleichen Messwerte für die verschiedenen Algorithmen verwendet, wodurch ein Vergleich mehrerer Algorithmen deutlich aussagekräftiger ist, als bei der Online-Evaluierung. Die Ergebnisse der Offline-Evaluierung sind reproduzierbar und verlieren nie ihre Gültig-

keit. Der Arbeitsaufwand ist ebenfalls geringer. Aufgrund der vielen Vorteile ermöglicht die Evaluierungsumgebung eine Offline-Evaluierung. Die Online-Evaluierung ist im Gegenzug weniger geeignet, jedoch nicht vollkommen unnütz. Die Bearbeitung von Hand und der daraus resultierende direkte Kontakt mit der Ortsberechnung sind positiv zu erwähnen. Dadurch ist es angebracht, auch die Online-Evaluierung zu implementieren, was auch getan wurde.

3.3.3 Ermittlung der Offline-Evaluierungspunkte

Bevor eine Evaluierung durchgeführt werden kann, müssen die Punkte ermittelt werden, an denen der Lokalisierungsalgorithmus getestet werden soll. Offline-Evaluierungspunkte haben große Ähnlichkeit mit Trainingspunkten, da sie die gleichen Mindestanforderungen haben: Sie müssen eine Position und die dort empfangenen Signalparameter beinhalten. Die Erstellung der Offline-Evaluierungspunkte wird nun folgend beschrieben. In Tabelle 3.4 sind die positiven und negativen Aspekte der einzelnen Varianten ablesbar.

Isolierung von Trainingspunkten

Eine Möglichkeit der Ermittlung von Offline-Evaluierungspunkten ist es, das dem Algorithmus zur Verfügung stehende Training als Grundlage zu verwenden. Diese Technik wurde von den Entwicklern von RADAR (Kapitel 2.3.1) und [17] eingesetzt.

Man benutzt einen einzelnen Trainingspunkt, samt den dort gemessenen Signalparametern, als Offline-Evaluierungspunkt. Um das Endergebnis nicht zu verfälschen, wird dieser Punkt für die Zeit der Berechnung aus der Radiomap entfernt. Würde man das nicht tun, wären einfache Verfahren, die nur nach einer perfekten Übereinstimmung mit einem Eintrag der Radiomap suchen, deutlich im Vorteil.

Sobald dem Algorithmus die für die Berechnung notwendigen Daten zur Verfügung stehen, kann auch eine Evaluierung durchgeführt werden. Das ist zeitsparend, da keine Aufnahme von weiteren Messpunkten durchgeführt werden muss. Leider hat diese Variante aber viele Nachteile. Es können beispielsweise nur Verfahren evaluiert werden, die mit einer Radiomap arbeiten. Wegen der Löschung des gewählten Trainingspunktes ist die Radiomap bei jeder Messung anders. Da der Punkt genau bei der gesuchten Position liegt, sind die Auswirkungen der Löschung noch größer. Die Dichte und Anordnung der Trainingspunkte entspricht dort nicht mehr den normalen Bedingungen im Rest der Radiomap. Diese Inkonsistenz des Datenbestandes lässt einen ordnungsgemäßen Test nicht zu. Für

die Evaluierung können nur die Positionen verwendet werden, die auch im Training absolviert wurden. Diese diskrete Evaluierung hat große Nachteile, da es dadurch nicht möglich ist, Positionen zwischen Trainingspunkten oder weit entfernt von Trainingspunkten zu benutzen. Dadurch ist auch die Anzahl der Offline-Evaluierungspunkte begrenzt. Die Evaluierung mit einem mobilen Nutzerprofil ist auf diese Art und Weise nicht möglich, weil die dafür verwendeten Trainingspunkte einen rein statischen Charakter haben. Ein letzter, aber wichtiger Fakt ist, dass die empfangenen Signaldaten keiner Veränderung durch Alterung unterliegen. In der Realität sind die Signaleigenschaften an einem Punkt nicht konstant (siehe Kapitel 2.2.2). Sie schwanken und verändern sich über einen längeren Zeitraum. Diese Tatsache wird in dieser Variante der Offline-Evaluierungspunkteermittlung ignoriert, da die Werte, die für Training und Evaluierung verwendet werden, die gleichen sind.

Aufnahme neuer Evaluierungspunkte

Das erneute Messen der Signaleigenschaften ist eine weitere Möglichkeit der Ermittlung von Offline-Evaluierungspunkten. Verfahren wie Horus [5, 6, 12] benutzen diese Technik. Die Erstellung eines Offline-Evaluierungspunktes ähnelt sehr der Erstellung eines Trainingspunktes. Dazu werden an mehreren Positionen Signalinformationen aufgenommen. Die Lage der Punkte ist dabei variabel und sie können alle Bereiche der Umgebungskarte abdecken. Im Gegensatz zum Absolvieren eines Trainings, bei dem die Punkte gleichmäßig verteilt werden sollten, ist bei den Offline-Evaluierungspunkten eine zufällige Verteilung möglich.

Das Verfahren der Ermittlung der Offline-Evaluierungspunkte durch einzelne Messungen ist - wie ein Training auch - sehr aufwendig. Man hat jedoch durch die Separierung der Daten mehrere Vorteile. Die Evaluierungsdaten sind unabhängig von einer Radiomap. Dadurch werden auch Verfahren ohne Radiomap unterstützt und eine vorhandene Radiomap wird nicht während der Messungen verändert. Ebenfalls müssen die Offline-Evaluierungspunkte nicht die gleichen Positionen wie die Trainingspunkte haben. Dadurch ist eine freie Verteilung der Punkte möglich. Durch eine Bewegung während der Messungen ist auch eine Evaluierung mit einem mobilen Nutzerprofil möglich. Um eine möglichst große Unabhängigkeit von den Trainingsdaten zu erreichen, können die Messungen für die Offline-Evaluierungspunkte an einem anderen Tag, zu einer anderen Uhrzeit, von einer anderen Person und mit anderer Hardware durchgeführt werden. Dadurch unterliegen die aufgenommenen Daten einer zufälligen Varianz und Alterung.

Bewertung der Ermittlung von Offline-Evaluierungspunkten

Von den zwei hier vorgestellten Arten der Offline-Evaluierungspunkteermittlung ist nur eine wirklich für eine vielseitige Evaluierung gebräuchlich. Die Variante der Isolierung von Trainingspunkten disqualifiziert sich gleich durch mehrere gravierende Unzulänglichkeiten. Dabei sind die größten Mankos, dass nur Radiomap-basierte Verfahren unterstützt werden und dass eine Inkonsistenz des Datenbestandes während der Berechnung auftritt. Die zweite Variante ist jedoch sehr gut für eine Evaluierung eines Algorithmus geeignet. Sie ermöglicht eine variable Anpassung an die gewünschten Testparameter (Kapitel 3.3.1). Der einzige Nachteil gegenüber der ersten Variante ist der erhöhte Arbeitsaufwand. Dieser kann sogar noch weiter erhöht werden, wenn zu Versuchszwecken einige Parameter verändert werden und dadurch eine komplett neue Ermittlung der Offline-Evaluierungspunkte nötig wird. Daher ist es ein Implementierungsziel, dass man die Testparameter ändern kann, ohne dadurch eine erneute Aufnahme der Evaluierungspunkte zu erzwingen. Aufgrund der vielen Pluspunkte der zweiten Variante wird die Methode der Aufnahme neuer Evaluierungspunkte für die Evaluierungsumgebung gewählt. Eine kurze Zusammenfassung der Vor- und Nachteile der einzelnen Varianten ist in Tabelle 3.4 dargestellt.

Kriterium	Trainingspunkt-Isolierung	Neuaufnahme
Zeitsparende Methode	Ja	Nein
Unterstützt Verfahren ohne Radiomap	Nein	Ja
Konsistenz des Vorwissens	Nein	Ja
Menge und Position der Punkte frei wählbar	Nein	Ja
Statisches und mobiles Nutzerprofil	Nein	Ja
Alterung der Daten	Nein	Ja

Tabelle 3.4: Bewertung der Ermittlungsmethoden der Offline-Evaluierungspunkte

4 Implementierung

Dieses Kapitel beschäftigt sich mit der Implementierung, der im vorherigen Kapitel 3 beschriebenen Konzepte der Evaluierungsumgebung.

Zuerst wird erläutert, wie die Anforderungen an die Universalität umgesetzt wurden. Dabei wird auf die Erstellung und Speicherung des Vorwissens, die Bearbeitung unterschiedlicher Signalinformationen und die Benutzer-Korrektur eingegangen. Danach folgt die Umsetzung der Komponentenaufteilung mit den Hauptthemen Kommunikation zwischen den Komponenten und Modularität der Algorithmen. Abschließend wird die Evaluierung näher betrachtet. Die Testparameter, die Ermittlung von Offline-Evaluierungspunkten und die Durchführung einer Evaluierung stehen dabei im Vordergrund.

4.1 Umsetzung der Universalitätsanforderungen

Wie in Kapitel 3.1 beschrieben, muss die Evaluierungsumgebung eine Vielzahl von Anforderungen erfüllen, um möglichst viele verschiedene Algorithmen zu unterstützen und so ihre Evaluierung zu ermöglichen. Es wird nun folgend beschrieben, wie diese Universalität erreicht wurde. Angefangen wird mit dem Vorwissen. Danach geht es mit den unterschiedlichen Signalinformationen weiter.

4.1.1 Daten des Vorwissens

Ein wichtiger Aspekt ist das Vorwissen, welches dem Lokalisierungsalgorithmus zur Verfügung steht und mit dessen Hilfe er die Ortsberechnung des Clients ausführt. In den Kapiteln 3.1.1 und 3.1.2 wurde beschrieben, welche Daten für die Topologie und für das Training wichtig sind und dass die Evaluierungsumgebung in der Lage sein muss, beide Arten des Vorwissens zu erstellen. Darum werden nun die Daten der Topologie, die Daten des Trainings und die Speicherung von beiden behandelt.

Topologie-Daten

Beginnend mit den Daten der Topologie wird in den beiden folgenden Tabellen (Tabelle 4.1 und 4.2) dargestellt, welche einzelnen Fakten zu jeder Umgebungskarte und zu jedem AP gespeichert werden.

Daten der Umgebungskarten: Die Ebene (`floor`) dient zur eindeutigen Identifikation einer Umgebungskarte. Darum ist jede Umgebungskarte genau einer einzigen Ebene zugeordnet. Dabei ist zu beachten, dass mehrere Umgebungskarten nicht zwangsläufig auf unterschiedlichen Etagen eines Gebäudes liegen müssen, wie der Begriff Ebene bzw. Floor es vielleicht impliziert. Die Ebene dient ausschließlich zur eindeutigen Unterscheidbarkeit. Dadurch ist es möglich, Teile einer Etage oder Außenbereiche auf verschiedene Ebenen zu legen und somit voneinander abzugrenzen. Des Weiteren wird in der Topologie die Länge (`length`) und Breite (`width`) der Umgebungskarte vermerkt. Die Ausmaße der Karte werden in Meter angegeben. Als letzte Angabe folgt der Speicherort der Bilddatei (`pictureSource`).

Bezeichner	Darstellung	Beispiel	Bedeutung
<code>floor</code>	<code>int</code>	4	Ebene
<code>length</code>	<code>double</code>	78.3379	Länge
<code>width</code>	<code>double</code>	27.814	Breite
<code>pictureSource</code>	<code>String</code>	Geb29_Etage4_final.svg	Pfad der Umgebungskarte

Tabelle 4.1: Umgebungskarten-Daten in der Topologie

Die Bilddatei dient nicht nur zur Darstellung der Umgebungskarte beim Manager, sondern ist ebenfalls Grundlage zur Ermittlung des begehbaren Bereichs. Als Format wurde aus mehreren Gründen SVG¹ (Scalable Vector Graphics) gewählt:

1. Eine SVG-Grafik ist besonders zur Darstellung großer Umgebungskarten geeignet, da man beliebig zoomen kann, ohne dass die Darstellung pixelig wird. Dies ist besonders bei großen Karten erforderlich, um eine exakte Positionsangabe durch Klicken auf die Umgebungskarte zu ermöglichen.

¹SVG-Grafiken sind in einer XML formulierten Sprache erstellte zweidimensionale Vektorgrafiken. Die Bildinformationen werden als Textinformationen gespeichert. Es können dazu Elemente wie Linien, Rechtecke, Kreise, Ellipsen, Polygone, Kurven, Text und weitere grafische Mittel verwendet werden.

2. Im Vergleich zu Rastergrafiken (z. B. jpeg (Joint Photographic Experts Group)) ist die Dateigröße besonders gering, da nicht die Ausmaße des Bildes sondern die textbasiert gespeicherten Inhalte bei SVG-Grafiken für die Dateigröße verantwortlich sind. Dieser Tatsache ist es auch zu verdanken, dass für die Darstellung von SVG-Grafiken wenig Arbeitsspeicher notwendig ist.
3. Die Daten werden im XML-Format (eXtensible Markup Language) gespeichert. Die dargestellten Objekte (Wände etc.) sind bereits als separate Strukturen vorhanden. Das erleichtert bei geometrischen Verfahren die Verarbeitung der Datei in Hinsicht auf die Ermittlung der Position von Hindernissen und des begehbaren Bereiches.

Daten der APs: Die Identifizierung eines APs erfolgt durch seine MAC-Adresse² (Media Access Control) (`mac`), da diese eindeutig ist. Die Angabe der MAC-Adresse ist demzufolge zwingend erforderlich. Ebenso erforderlich ist die Angabe des Ortes an dem sich der AP befindet. Der Ort wird durch die Koordinaten X (`xpos`), Y (`ypos`) und Z (`zpos`) sowie durch die Ebene (`floor`), auf der er sich befindet, festgelegt. Die Angabe der Koordinaten erfolgt in der Maßeinheit Meter. Die Ebene dient der Zuordnung eines APs zu einer bestimmten Umgebungskarte. Weitere wichtige Fakten über den AP sind das Protokoll (`protocol`) (siehe Kapitel 2.1.4) und die Sendeleistung (`transmitPower`), da diese Informationen für die Berechnung der maximalen Reichweite des APs notwendig sind. Zusätzlich können über den AP noch weitere Angaben hinzugefügt werden. Das sind die IP-Adresse (Internet Protocol) (`IP`), die SSID³ (Service Set Identifier) (`ssid`), der Kanal auf dem der AP sendet (`channel`) (siehe Kapitel 2.1.2) und ein frei wählbarer Kommentar (`comment`).

Erstellung und Veränderung der Topologie-Daten: Die APs sind immer einer bestimmten Umgebungskarte zugeordnet. Darum ist das Hinzufügen einer Umgebungskarte immer der erste Schritt bei der Erstellung einer Topologie. Der Nutzer gibt dazu die Ebene und den Speicherort der Bilddatei an. Der Speicherort kann sowohl durch einen absoluten Pfad, als auch durch einen relativen Pfad in Abhängigkeit zur Topologie-Datei dargestellt werden. Der Nutzer kann auch die Länge und Breite des durch die Bilddatei ab-

²Eine MAC-Adresse ist eine Hardware-Adresse, die zur weltweit eindeutigen Identifikation eines am Netzwerk teilnehmenden Adapters verwendet wird. Sie besteht aus 48 Bit.

³Die SSID ist der Netzwerkname eines WLANs. Diese Kennung wird vom AP festgelegt und dient zur Identifikation des Funknetzes.

Bezeichner	Darstellung	Beispiel	Bedeutung
mac	String	00:0F:B5:0F:9A:40	MAC-Adresse
xpos	double	68.407	X-Koordinate
ypos	double	1.5714	Y-Koordinate
zpos	double	0	Z-Koordinate
floor	int	4	Ebene
protocoll	String	g	Protokoll
transmitPower	int	100	Sendeleistung
IP	String	192.168.66.66	IP-Adresse
ssid	String	awds	SSID
channel	int	3	Kanal
comment	String	Test-AP	Kommentar

Tabelle 4.2: AP-Daten in der Topologie

gebildeten Arealen angeben. Da diese Information dem Nutzer jedoch meistens nicht exakt vorliegt, besteht die Möglichkeit, dass die Evaluierungsumgebung die korrekten Ausmaße ermittelt. Dazu muss der Nutzer lediglich eine von ihm selbst gewählte kurze Strecke abmessen. Mittels der angezeigten und der realen Länge der Strecke lässt sich die Größe der Karte berechnen. Man kann beliebig Karten zur Topologie hinzufügen oder sie wieder entfernen.

Zu jeder Umgebungskarte kann man beliebig viele APs erstellen. Nach Auswahl des Befehls zum Hinzufügen eines neuen APs im Menü kann durch einen Mausklick auf die Umgebungskarte der AP frei positioniert werden. Die obligatorischen Angaben der Koordinaten und der Ebene sind dann bereits in die nun erscheinende Eingabemaske eingetragen. Der Nutzer kann weitere Daten angeben, darunter auch die MAC-Adresse, die für die Identifikation des APs benötigt wird. Bereits vorhandene APs können verschoben, editiert und gelöscht werden. Sie können ebenfalls einer anderen Umgebungskarte zugeordnet werden.

Speicherung der Topologie-Daten: Es besteht die Möglichkeit, die oben beschriebenen Daten in einer sequenziellen Weise in einer Datei zu speichern, wobei jede Zeile einer Umgebungskarte bzw. einem AP entspricht. Solch eine Datei wäre von der Evaluierungsumgebung relativ einfach zu lesen und zu erzeugen. Zudem beschränkt sich der Inhalt der Datei ausschließlich auf die Daten und verbraucht daher wenig Speicherplatz. Andererseits ist die Datei in dieser Form schwer von Menschen und von anderen Appli-

kationen les- und erzeugbar. Dies stimmt nicht mit den in der Aufgabenstellung (Kapitel 1.2) geforderten Eigenschaften überein. Eine bessere Möglichkeit bietet die Speicherung im XML-Format. Das Auslesen und Schreiben ist zwar ein wenig aufwendiger, jedoch flexibler gegenüber strukturellen Veränderungen im Aufbau der Datei. Die Dateigröße übersteigt die einer Datei mit sequenziellem Aufbau geringfügig, jedoch erfüllt diese Variante die eingangs gestellten Forderungen. Die XML-Struktur ist leichter verständlich und ist auch von anderen Applikationen lesbar. Aus diesem Grund wurde die Topologie in einer Datei im XML-Format gespeichert. Die Datei hat das Suffix „project“ und wird daher Projekt-Datei genannt.

Trainings-Daten

Ein Training besteht aus einer Ansammlung von Orten und den dort gemessenen Signalparametern. Um die Universalität zu gewährleisten, sind noch weitere Daten (Orientierung, Zeitpunkt der Messung) notwendig, die im zweiten Teil des Kapitels 3.1.2 aufgeführt sind. Des Weiteren wird in Kapitel 3.1.6 verlangt, dass auch jeder Trainingspunkt mit dem Clientnamen versehen wird. Um zu Versuchszwecken einige Trainingspunkte für die Lokalisierung zu deaktivieren, wie in Kapitel 3.3.1 gefordert, ist ebenfalls ein Status jedes Trainingspunktes und daraus folgend eine eindeutige Identifizierung notwendig. In diesem Kapitel wird auch erwähnt, dass die Menge der Signalparameterwerte je Trainingspunkt veränderbar sein sollte. Darum werden alle während eines Trainings aufgezeichneten Signalparameter zusammen mit der aktuellen Uhrzeit gespeichert. Anhand dessen kann eine bestimmte Zeitspanne zur Verwendung für das Training ausgewählt werden.

Speicherung der Trainings-Daten: Aufgrund der Gleichförmigkeit der Trainingsdaten und der Tatsache, dass diese Daten von keiner anderen Applikation gelesen und geschrieben werden müssen, ist eine sequenzielle Speicherung angebracht. Zu jeder Position werden mehrere Signalparameterwerte gespeichert. Die sich daraus ergebende Datei mit Namen „training.txt“ hat die im folgenden Beispiel (Abbildung 4.1) ersichtliche Struktur. Aufgrund der Vorgaben besteht jeder Trainingspunkt aus folgenden Angaben: ID, Status, Clientname (IP-Adresse), Zeitpunkt an dem das Training für diesem Punkt abgeschlossen wurde, X- und Y-Koordinaten, Orientierung und Ebene. Darunter werden die Signalparameterwerte mit den dazugehörigen Zeitpunkten der Messungen geschrieben.

```
5|allowed|10.1.2.41|10.07.2006 Mon 11:28:55.810|73.6074|25.0386|S|4|
|10.07.2006 Mon 11:28:36.415|awds|00:0F:B5:0F:9A:D0|-50|EUK-LAB...
|10.07.2006 Mon 11:28:37.199|awds|00:0F:B5:0F:9A:D0|-44|EUK-LAB...
...
6|forbidden|10.1.2.41|10.07.2006 Mon 11:29:26.271|75.4834|27.2641|W|4|
...
```

Abbildung 4.1: Struktur der Datei „training.txt“ am Beispiel

4.1.2 Behandlung unterschiedlicher Signalinformationen

Je nachdem, welche Art von Signalinformationen (siehe Kapitel 2.2.1) für die Lokalisierung verwendet werden sollen, haben die vom Client geschickten Daten eine andere Struktur und einen anderen Inhalt. Aufgrund des in Kapitel 3.2.3 vorgestellten gekapselten Aufbaus des Servers ist nur der *Algorithm* in der Lage, diese Daten auszuwerten. Demzufolge muss der *EngineHolder*, der für den Empfang der Pakete vom Client zuständig ist, diese Daten - ungeachtet des Inhalts - speichern oder an den *Algorithm* weiterleiten. Aus diesem Grund wird der gesamte Inhalt des Paketes als String (Zeichenkette) behandelt. Im Falle eines Trainings (siehe Abbildung 4.1) wird dieser String hinter Datum und Uhrzeit gespeichert. Dieses Training kann dann vom *Algorithm* ausgewertet werden.

Soll der *Algorithm* eine Lokalisierung durchführen, benutzt er dafür die in der Datenstruktur `lastSignalValues` gespeicherten Daten. Sie enthält die aus den Paketen der Clients gewonnenen Strings der letzten `secsToSaveClientSendInfos`⁴ Sekunden. Um ein mobiles Nutzerprofil zu unterstützen, wird zu jedem String die Empfangszeit des Paketes gespeichert. Als Datenstruktur wurde eine Hash-Tabelle gewählt, da die enthaltenen Daten durch die Empfangszeit identifiziert werden müssen. Eine Hash-Tabelle ermöglicht eine sehr schnelle Suche nach den zu einem bestimmten Schlüssel gespeicherten Daten.

4.1.3 Benutzer-Korrektur

Lokalisierungsverfahren, die eine Benutzer-Korrektur unterstützen, können die berichtigte Position eines Clients verwenden, um ihre Genauigkeit zu erhöhen. Der Nutzer des Managers ist in der Lage, seine korrekte Position anzugeben und diese an den Server zu

⁴Die Variable `secsToSaveClientSendInfos` gibt an, nach wie vielen Sekunden die vom Client gesendeten Daten gelöscht werden sollen. Dadurch wird auch festgelegt, nach wie vielen Sekunden ein inaktiver Client aus der Liste der bekannten Clients entfernt wird.

übertragen. Diese Angabe kann dabei auf zwei Weisen erfolgen. Der zu korrigierende Client kann per Drag&Drop an die richtige Position auf der Umgebungskarte verschoben werden. Ein anderer Weg ist die Aktivierung des Korrekturmodus im Menü. Hier reicht es den betreffenden Client und dann die korrekte Position anzuklicken. Es wird ein Paket an den Server gesendet (siehe Kapitel 4.2.1), das die errechnete und die wirkliche Position des Clients enthält.

4.2 Umsetzung des Komponentenaufbaus

In Kapitel 3.2 wurde eine Trennung der Evaluierungsumgebung in drei Teile gewählt. Nun folgend wird erklärt, wie diese Komponenten miteinander kommunizieren. Danach erfolgt die Beschreibung der Implementierung des Konzeptes der austauschbaren Algorithmen.

4.2.1 Kommunikation

Wie in Abbildung 3.3 ersichtlich, müssen Manager und Server sowie Clients und Server miteinander kommunizieren, um alle Aufgaben der Evaluierungsumgebung erfüllen zu können. In den folgenden Kapiteln wird erläutert, welche Daten zwischen den einzelnen Komponenten versendet werden.

Kommunikation zwischen Manager und Server

Zwischen dem Manager und dem Server werden größtenteils Befehle und Daten ausgetauscht, welche für die ordnungsgemäße Funktionsweise der Evaluierungsumgebung von großer Bedeutung sind. Aufgrund dieser Tatsache wurde als Transportprotokoll TCP implementiert.

Der Manager steuert den Server, indem er ihm Befehle sendet. Außerdem initiiert er die Aufnahme von Trainings- und Evaluierungspunkten. Er schickt ihm Daten, die für die Ortsberechnung wichtig sind, z. B. die Topologie und welche APs unter der eigenen Kontrolle stehen. Der Server hingegen versorgt den Manager mit allen Informationen, die er angefordert hat. Dazu gehören z. B. die Clientpositionen und die Orte der Trainings- und Offline-Evaluierungspunkte. Er teilt ihm auch mit, welche Clients momentan lokalisierbar sind und welche APs bekannt sind. Es folgen nun die Erläuterungen von Befehlen und ihren Bedeutungen.

Absolvieren eines Trainings: Um ein Training zu absolvieren, muss die Position des Clients bekannt und konstant sein. Es werden die Befehle aus Tabelle 4.3 dazu verwendet.

Befehl	Richtung	Bedeutung
trainingPoint	Manager \Rightarrow Server	Position eines Trainingspunktes
trainingPointFinished	Manager \Leftarrow Server	Trainingspunkt fertig
sendAllTrainingPoints	Manager \Rightarrow Server	Anfrage von Trainingspunktpositionen
allTrainingPoints	Manager \Leftarrow Server	Positionen der Trainingspunkte
trainingPointStatus	Manager \Rightarrow Server	Änderung des Trainingspunktstatus
trainingFinished	Manager \Rightarrow Server	Training beendet

Tabelle 4.3: Kommunikation für das Absolvieren eines Trainings

Der Manager schickt ein Paket mit dem Befehl `trainingPoint` und der Position des Clients an den Server. Dieser speichert alle vom Client geschickten Signalparameter zusammen mit der Position in der Datei „training.txt“. Dies tut der Server für eine bestimmte Sekundenzahl, die durch die Variable `secsForTraining` festgelegt wird. Nach Ablauf dieser Zeit sendet der Server den Befehl `trainingPointFinished` und signalisiert dem Manager so, dass der Client umpositioniert werden kann, um einen neuen Trainingspunkt aufzunehmen.

Mithilfe der Anweisung `sendAllTrainingPoints` wird der Server veranlasst dem Manager die Position der bereits trainierten Punkte mitzuteilen. Dies tut der Server dann durch den Befehl `allTrainingPoints`. Nun kann der Manager den Status der Trainingspunkte ändern, um die Menge der für die Lokalisierung erlaubten Trainingspunkte festzulegen. Mittels `trainingPointStatus` werden diese Informationen an den Server weitergereicht. Ein Versenden des Befehls `trainingFinished` informiert den Server darüber, dass das Training abgeschlossen ist.

Austausch ortungsrelevanter Informationen: Die in Tabelle 4.4 beschriebenen Befehle dienen zum Informationsaustausch.

Wenn der Manager die Liste der bekannten APs mittels des Befehls `sendKnownAPsList` anfordert, antwortet der Server mit `knownAPsList`. Nun kann der `controlled`-Status jedes einzelnen bekannten APs verändert werden, wie in Kapitel 3.3.1 erwähnt. Die Veränderungen werden dem Server durch den Befehl `changedKnownAPsList` mitgeteilt und sind von nun an gültig.

Befehl	Richtung	Bedeutung
sendKnownAPsList	Manager \Rightarrow Server	Anfrage der bekannten APs
knownAPsList	Manager \Leftarrow Server	Bekannte APs
changedKnownAPsList	Manager \Rightarrow Server	Modifizierte bekannte APs
sendImportantVariables	Manager \Rightarrow Server	Anfrage der wichtigen Variablen
ImportantVariables	Manager \Leftarrow Server	Wichtige Variablen
changedImportantVariables	Manager \Rightarrow Server	Veränderte wichtige Variablen
sendKnownClientsList	Manager \Rightarrow Server	Anfrage der bekannten Clients
knownClientsList	Manager \Leftarrow Server	Bekannte Clients
projectFile	Manager \Rightarrow Server	Topologie-Datei
mapFile	Manager \Rightarrow Server	Umgebungskarten-Datei
errorMessage	Manager \Leftarrow Server	Fehlermeldung

Tabelle 4.4: Kommunikation für den Austausch ortungsrelevanter Informationen

Auf die gleiche Weise können die Variablen des Servers (`secsToSaveClientSendInfos`, `secsForTraining`, `secsForLocalization`, `noConnectionValue`) verändert werden. Durch den Befehl `sendImportantVariables` fordert der Manager sie an. Der Server übersendet sie mittels `ImportantVariables`. Nun können die Werte vom Nutzer des Managers verändert und per `changedImportantVariables` zurück geschickt werden.

Mit den Befehlen `sendKnownClientsList` und `knownClientsList` verhält es sich ähnlich. So erfährt der Manager, welche Clients dem Server gerade bekannt sind und lokalisiert werden können.

Dem Server muss die Topologie des Netzwerkes bekannt sein, um die Lokalisierung mit einem geometrischen Verfahren durchführen zu können. Aus diesem Grund kann der Manager die Projekt-Datei (`projectFile`) und die Umgebungskarten-Dateien (`mapFile`) zum Server senden.

Wenn der Server ein Paket mit einem unbekanntem Befehl erhält oder ein Fehler auftritt, wird der Manager benachrichtigt, indem er ein Paket mit dem Befehl `errorMessage` und der Fehlerbeschreibung erhält. Diese Beschreibung wird dann beim Manager angezeigt.

Lokalisierung von Clients: Um einen Client zu lokalisieren sind die in Tabelle 4.5 aufgelisteten Befehle notwendig.

Der Manager verlangt die Lokalisierung eines Clients durch Versenden eines Paketes mit

Befehl	Richtung	Bedeutung
locateClient	Manager \Rightarrow Server	Anfrage der Clientposition
clientLocation	Manager \leftarrow Server	Clientposition
clientLocationCorrection	Manager \Rightarrow Server	Korrektur der Clientposition

Tabelle 4.5: Kommunikation für die Lokalisierung von Clients

dem Befehl `locateClient`. Daraufhin berechnet der Server die Position aus den Signalparameterwerten der letzten - durch die Variable `secsForLocalization` festgelegten - Sekunden. Die Position wird dann mittels des Befehls `clientLocation` an den Manager zur grafischen Darstellung geschickt.

In manchen Fällen weiß der Nutzer seine eigene Position exakt und könnte diese dem Algorithmus mitteilen (siehe Kapitel 3.1.6). Dies ist möglich, indem der Manager den Befehl `clientLocationCorrection` und die korrekte Position an den Server schickt.

Evaluierung von Algorithmen: In der folgenden Tabelle 4.6 sind alle nötigen Kommunikationsschritte erläutert, die zur Evaluierung eines Algorithmus notwendig sind.

Befehl	Richtung	Bedeutung
firstEvalPoint	Manager \Rightarrow Server	Erster Offline-Evaluierungspunkt
moreEvalPoints	Manager \Rightarrow Server	Weiterer Offline-Evaluierungspunkt
pauseCollectingEvalPoints	Manager \Rightarrow Server	Pausiere Aufnahme von Evaluierungspunkten
stopCollectingEvalPoints	Manager \Rightarrow Server	Beende Aufnahme von Evaluierungspunkten
sendAllOfflineEvalPoints	Manager \Rightarrow Server	Anfrage von Offline-Evaluierungspositionen
allOfflineEvalPoints	Manager \leftarrow Server	Offline-Evaluierungspositionen
processOfflineEvalData	Manager \Rightarrow Server	Berechne Offline-Evaluierung
offlineEvalResult	Manager \leftarrow Server	Ergebnis der Offline-Evaluierung

Tabelle 4.6: Kommunikation für die Durchführung einer Evaluierung

Zuerst müssen die Offline-Evaluierungspunkte aufgezeichnet werden (Kapitel 3.3.3). Die Befehle `firstEvalPoint` und `moreEvalPoints` ermöglichen dies. Beide Befehle enthalten die Position des Evaluierungspunktes und veranlassen den Server dazu, die Aufzeichnung der Signalparameter zu starten. Beim ersten Befehl wird dem Server zusätzlich das Nutzerprofil mitgeteilt, damit dieser die Art der Aufzeichnung der Signalparameter anpassen kann. Der Manager kann die Aufnahme der Evaluierungspunkte durch die Anweisungen `pauseCollectingEvalPoints` kurzzeitig unterbrechen und durch `stopCollectingEvalPoints` beenden. Eine kurzzeitige Pause ist bei einem statischen Nutzerprofil unerlässlich, um während der Aufnahme mehrerer Evaluierungspunkte die

Position des Clients zu verändern, ohne dass währenddessen Signalparameter gespeichert werden.

Es lassen sich durch den Befehl `sendAllOfflineEvalPoints` die Positionen aller bereits aufgezeichneter Offline-Evaluierungspunkte anfordern. Der Server schickt diese dann per `allOfflineEvalPoints` an den Manager. Sind genug Offline-Evaluierungspunkte gesammelt worden, lässt der Befehl `processOfflineEvalData` die Berechnung starten. Dies ist ebenfalls durch ein Klicken auf die Schaltfläche „Process offline-eval data“ beim Server möglich. Dieser verwendet die Signalparameterwerte der Offline-Evaluierungspunkte für die Lokalisierung. Die errechnete Position lässt sich mit der bekannten Position vergleichen und somit kann man die Fehlerabweichung ermitteln. Das Resultat der Evaluierung wird sowohl beim Server gespeichert, als auch per `offlineEvalResult` an den Manager geschickt.

Kommunikation zwischen Client und Server

Der Client schickt ortsabhängige Signalparameter zum Server. Die Kommunikation ist unidirektional, da der Server keine Daten zum Client schicken muss. In Kapitel 3.2.2 wurde über die Wahl des Transportprotokolls diskutiert, wobei sich TCP als besser geeignet herausgestellt hat. Für die Verwendung von TCP ist viel zusätzliche Arbeit notwendig um die Uhrensynchronisation zu gewährleisten und das Kanalwechselproblem zu lösen. Im Zeitrahmen dieser Diplomarbeit ist dieser Mehraufwand jedoch nicht zu bewältigen. Aus diesem Grund wurde die Implementierung prototypisch mit UDP durchgeführt. Die in Kapitel 5 beschriebenen Messungen wurden so absolviert, dass das Problem der Umsortierung der Pakete eliminiert wurde. Alle verwendeten Geräte befinden sich im selben Subnetz. Es wird kein Routing auf der IP-Ebene durchgeführt, wodurch es keine Umsortierung geben kann. Der potenzielle Verlust von Paketen kann dadurch jedoch nicht verhindert werden. Er kann jedoch minimiert werden. Der Übertragungsweg der Pakete verläuft über Unicast-Nachrichten von Client zum AP und von dort aus über Ethernet zum Server. Da Ethernet ein sehr sicheres Kommunikationsmedium ist, gehen hier sehr wenig Pakete auf dem Weg vom AP zum Server verloren. Diese Sicherheit bietet die drahtlose Kommunikation zwischen Client und AP nicht. Da jedoch Unicast-Pakete auf der MAC-Ebene bestätigt werden, unabhängig ob UDP oder TCP als Transportprotokoll verwendet wird, werden verlorene Pakete wiederholt übertragen. Bei einer Auszählung der Verluste von ca. 3300 Paketen ergab sich, dass kein einziges verloren ging. Die MAC-Ebene

musste jedoch 6 Stück (0,18%) wiederholt übermitteln. Ein Einfluss auf die Evaluierung ist bei derart wenigen Verlusten kaum zu bemerken.

4.2.2 Algorithmen als Module

Um mehrere verschiedene Algorithmen verwenden zu können, muss der Server so aufgebaut sein, dass der Algorithmus als austauschbares Modul in den Server integriert werden kann (siehe Kapitel 3.2.3). Dies wird durch die Trennung des Servers in *EngineHolder* und *Algorithm* verwirklicht. Der *EngineHolder* wird als abstrakte Klasse⁵ definiert, damit der *Algorithm* die von ihm angebotenen Dienste verwenden kann. Dazu erbt er vom *EngineHolder*. Die Kommunikation zwischen den beiden Teilkomponenten erfolgt dabei über virtuelle und rein virtuelle Methoden⁶, die bei bestimmten Ereignissen vom *EngineHolder* aufgerufen werden. Diese nun folgend beschriebenen Methoden können von *Algorithm* überschrieben werden. Die einzigen beiden Funktionen, die überschrieben werden müssen, sind `calculateClientPosition` und `clientDataReceived`, da sie elementare Aufgaben erfüllen. Aus diesem Grund sind die beiden Funktionen rein virtuell. Der *Algorithm* implementiert also nur diejenigen Funktionen, welche für die korrekte Abarbeitung des Lokalisierungsverfahrens notwendig sind.

Die Ereignisse, bei denen diese Methoden aufgerufen werden, lassen sich grob in vier Kategorien einteilen: Befehle vom Manager (Tabelle 4.7), Daten vom Manager (Tabelle 4.8), Veränderungen wichtiger Variablen (Tabelle 4.9) und Client-bezogene Informationen (Tabelle 4.10).

Befehle vom Manager: Die rein virtuelle Funktion `calculateClientPosition` ist die wohl wichtigste Funktion, da sie ausgeführt wird, wenn der Manager die Lokalisierung eines Clients verlangt. Als Daten erhält der *Algorithm* den Clientnamen und eine Uhrzeit (diese ist nötig für die Offline-Evaluierung). Es wird die aus den gespeicherten Signalparametern errechnete Position des Clients und die erwartete Fehlerabweichung an den *EngineHolder* zurückgegeben. Wenn der Manager einen Trainingspunkt geschickt hat, so wird `trainingPointReceived` mit dem Clientnamen, der Position und der Orientierung aufgerufen. Ist das gesamte Training beendet, so signalisiert der Manager

⁵Eine abstrakte Klasse enthält mindestens eine rein virtuelle Methode. Dadurch kann man keine Instanzen anlegen, sondern nur von dieser Klasse erben.

⁶Eine virtuelle Methode **kann** von einer erbenenden Klasse überschrieben werden. Eine rein virtuelle Methode **muss** überschrieben werden, da sie in der Basisklasse nur deklariert, jedoch nicht definiert wird.

dies und die Methode `trainingFinished` wird ausgeführt. Der Zweck der Funktion `correctionReceived` ist die Mitteilung einer Korrektur einer Clientposition. Dazu wird der Clientname, die Uhrzeit sowie die falsche, als auch die korrekte Position mitgeteilt.

Methodenname	Bedeutung	Übermittelte Daten
<code>calculateClientPosition</code>	Lokalisierung	Clientname, Uhrzeit
<code>trainingPointReceived</code>	Trainingspunkt erhalten	Trainingspunktinformation
<code>trainingFinished</code>	Training beendet	keine
<code>correctionReceived</code>	Positionskorrektur erhalten	Falscher und korrekter Ort

Tabelle 4.7: rein virtuelle Methoden des *EngineHolders*, Befehle vom Manager

Daten vom Manager: Auf die Übermittlung wichtiger Daten vom Manager kann ebenfalls reagiert werden. Dazu gibt es die Methode `projectFileReceived`, die den Empfang der Projekt-Datei signalisiert. Des Weiteren dient `mapFileReceived` zur Benachrichtigung im Falle des Empfanges einer Umgebungskarte. Hat der Manager den `controlled`-Status der bekannten APs verändert und übertragen, wird die Funktion `knownAPsListReceived` gestartet. Der *Algorithm* kennt die Speicherorte der gesendeten Daten. Die Befehle dienen daher nur dazu, ihn darüber zu informieren, dass die Daten verändert wurden.

Methodenname	Bedeutung	Übermittelte Daten
<code>projectFileReceived</code>	Projekt-Datei erhalten	keine
<code>mapFileReceived</code>	Umgebungskarte erhalten	keine
<code>knownAPsListReceived</code>	Status der bekannten APs geändert	keine

Tabelle 4.8: rein virtuelle Methoden des *EngineHolders*, Daten vom Manager

Veränderung wichtiger Variablen: Der *Algorithm* muss informiert werden, falls sich wichtige Variablen, Einstellungen oder Daten ändern. Diese Informationen werden ihm durch die in Tabelle 4.9 aufgelisteten Befehle mitgeteilt.

Direkt nach dem Start des Servers wird `engineHolderStarted` ausgeführt. Die Wahl eines anderen Arbeitsverzeichnis führt zu dem Aufruf von `workingDirChanged`.

4 Implementierung

Methodenname	Bedeutung	Übermittelte Daten
engineHolderStarted	Programmstart	keine
workingDirChanged	Arbeitsverzeichnis geändert	keine
readAPsFromFileFinished	APs aus knownAPs.txt gelesen	keine
knownAPsSavedToFile	knownAPs gespeichert	keine
readFromProjectFileFinished	Aus Projekt-Datei gelesen	keine
trainingSaved	Training gespeichert	Trainingspunktinformation
secsToSaveClientSendInfosChanged	Diese Variable geändert	keine
secsForTrainingChanged	Diese Variable geändert	keine
secsForLocalizationChanged	Diese Variable geändert	keine
noConnectionValueChanged	Diese Variable geändert	keine

Tabelle 4.9: rein virtuelle Methoden des *EngineHolder*s, Veränderung wichtiger Variablen

Das ist beim Server über die Schaltfläche „Change working DIR“ möglich. Durch das Arbeitsverzeichnis wird festgelegt, wo sich die Dateien befinden, die für den Betrieb des Servers notwendig sind. Wurden die bekannten APs aus der Datei „knownAPs.txt“ gelesen, wird die Funktion `readAPsFromFileFinished` aufgerufen. Ist hingegen das Schreiben in diese Datei erfolgreich beendet worden, führt dies zum Aufruf von `knownAPsSavedToFile`. Ein fehlerloses Auslesen der Projekt-Datei wird durch `readFromProjectFileFinished` angezeigt. Nachdem der Manager das Training initiiert hat und `secsForTraining` Sekunden vergangen sind, wird das Training gespeichert. Dieses Ereignis wird mit `trainingSaved` signalisiert, wobei noch einige Daten (Clientname, Position, Orientierung) übergeben werden. Werden die Variablen geändert, die bestimmen, wie viele Sekunden für Training und Lokalisierung verwendet werden, so führt dies zur Ausführung von `secsToSaveClientSendInfosChanged`, `secsForTrainingChanged` und `secsForLocalizationChanged`. Eine weitere wichtige Variable ist die `noConnectionValue`, die angibt, welcher Zahlenwert (z. B. RSSI) angenommen werden soll, wenn keine Verbindung zu einem AP besteht. Es wird `noConnectionValueChanged` aufgerufen, sollte dieser Wert geändert worden sein. Diese vier Variablen sind sowohl vom Manager, als auch vom Server änderbar. Der Server bietet dazu die Schaltfläche „Change important variables“ an.

Client-bezogene Informationen: Der *EngineHolder* hat eine Liste aller Clients, die gerade Informationen senden. Beginnt ein neuer Client mit dem Senden von Signalparametern, wird sein Clientname durch `newClientDetected` an den *Algorithm* übergeben. Im Falle eines Clients, der keine Daten mehr an den Server sendet, wird `clientDisappeared` benutzt. Die zweite rein virtuelle Methode ist `clientDataReceiv-`

ed. Sie dient dazu, Informationen aus den vom Client geschickten Paketen zu extrahieren, die der *EngineHolder* selbst nicht auslesen kann. Dazu werden der Clientname und der Inhalt des Paketes an den *Algorithm* weitergeleitet. Dieser kann dann z. B. die Liste APs ermitteln, die der Client gerade empfangen hat und diese mit der Liste der bekannten APs abgleichen.

Methodenname	Bedeutung	Übermittelte Daten
newClientDetected	Neuer Client entdeckt	Clientname
clientDisappeared	Client verschwunden	Clientname
clientDataReceived	Daten vom Client erhalten	Clientname, Daten

Tabelle 4.10: rein virtuelle Methoden des *EngineHolder*s, Client-bezogene Informationen

4.3 Umsetzung der Evaluierung

Nun folgend soll die Evaluierung von Lokalisierungsalgorithmen näher erläutert werden, die in Kapitel 3.3 konzipiert wurde. Die Testparameter werden als Erstes behandelt. Dabei wird geklärt, wie die einzelnen Parameter festgelegt und verändert werden können. Im Anschluss daran werden die Erstellung der Offline-Evaluierungspunkte und die Implementierung der Evaluierung beschrieben.

4.3.1 Variable Testparameter

Bevor man einen Test durchführen kann, ist es notwendig, die Testparameter festzulegen. Es wird im Folgenden angenommen, dass die Testumgebung (siehe Kapitel 2.5) dabei gleich bleibt. Nur Tests mit exakt gleichen Parametern können miteinander verglichen werden. Zu Testzwecken ist es jedoch unabdingbar, dass diese Parameter auch geändert werden können. Die Evaluierungsumgebung stellt Möglichkeiten bereit, dies zu tun.

Anzahl, Dichte und Verteilung der Trainingspunkte: Der Manager kann vom Server eine Liste der Positionen der Trainingspunkte anfordern (siehe Kapitel 4.2.1). Diese Positionen werden auf der Umgebungskarte durch Kreise dargestellt. Durch ein einfaches Anklicken eines Trainingspunktes kann dessen Status geändert werden. Dieser Status legt fest, ob ein Trainingspunkt für die Lokalisierung verwendet wird (*allowed*) oder nicht (*forbidden*). Diese Informationen werden in der Datei „training.txt“ (siehe Abbildung

4.1) gespeichert. Im Manager dient eine farbliche Kennzeichnung zur leichten Unterscheidung. Dunkelblaue Trainingspunkte haben den Status `allowed` und türkise `forbidden` (siehe Abbildung A.1). Der veränderte Status der Trainingspunkte wird dem Server mitgeteilt, wenn im Manager der Menüpunkt „Training finished“ benutzt wird. Es besteht außerdem jederzeit die Möglichkeit, neue Trainingspunkte aufzunehmen und dem schon bestehenden Training hinzuzufügen.

Menge der Signalinformationen: Der Server ist in der Lage, die Menge der Signalinformationen für die Lokalisierung zu begrenzen. Es werden nur die Signalparameterwerte benutzt, die innerhalb einer Zeitspanne liegen, welche durch die Variable `secsForLocalization` definiert wird. Auf die gleiche Weise kann die Menge der Signalinformationen verändert werden, die für einen Trainingspunkt verwendet werden. Die Variable `secsForTraining` legt das fest. Bei einem bereits absolvierten Training ist es so möglich die Menge der Signalinformationen zu beschränken, die für die Berechnung der Radiomap verwendet werden.

Anzahl, Dichte und Verteilung der APs: Die Clients senden diejenigen Adressen der APs zum Server, die sich in ihrer Reichweite befinden. Das geschieht automatisch durch das Versenden der Signalinformationen. Der Server führt eine Liste der ihm bekannten APs in der Datei „knownAPs.txt“. Zu jedem AP (identifiziert durch seine MAC-Adresse) wird dabei seine SSID und ein `controlled`-Status gespeichert. Dieser Status gibt an, ob der jeweilige AP in die Berechnung mit einbezogen werden darf oder nicht. Ein neu in die Liste aufgenommener AP muss erst explizit für die Berechnung zugelassen werden, um nicht eine eventuell laufende Messung zu beeinflussen. Der Manager kann sich diese Liste anfordern, den `controlled`-Status jedes einzelnen APs ändern und die aktualisierte Liste wieder zurück senden (siehe Kapitel 4.2.1).

4.3.2 Aufnahme von Offline-Evaluierungspunkten

Bevor eine Offline-Evaluierung durchgeführt werden kann, müssen die Evaluierungspunkte ermittelt werden (siehe Kapitel 3.3.3). Nun folgend wird beschrieben, wie das geschieht und wie das Nutzerprofil Einfluss darauf hat. Die Berechnung der Evaluierung wird im darauf folgenden Kapitel dargelegt.

Offline-Evaluierungspunkte-Ermittlung bei statischem Nutzerprofil

Weil sich ein statischer Nutzer während einer Lokalisierung nicht bewegt, ähnelt die Ermittlung der Offline-Evaluierungspunkte dem Absolvieren eines Trainings. Der Client wird positioniert und auf der Umgebungskarte wird diese Position angeklickt. Die Aufzeichnung wird nach Auswahl des Nutzerprofils durch das Versenden eines Paketes mit dem Befehl `firstEvalPoint` gestartet. Der Server speichert nun das Nutzerprofil, die Position des Clients und alle ab dem jetzigen Zeitpunkt empfangenen Signalparameter dieses Clients in der Datei „evalPoints.txt“. Soll der Client umpositioniert werden, um einen neuen Offline-Evaluierungspunkt zu erstellen, wird auf die betreffende Stelle der Umgebungskarte geklickt. Die Aufzeichnung wird durch diese Aktion mittels des Befehls `pauseCollectingEvalPoints` so lange unterbrochen, bis der Nutzer anschließend auf „OK“ klickt um zu bestätigen, dass sich der Client nun an dem gewählten Ort befindet. Dieser Ort wird durch den Befehl `moreEvalPoints` an den Server übermittelt, welcher wiederum die Aufzeichnung der vom Client gesendeten Signalparameter startet. Sollte er die Aufzeichnung jedoch abbrechen wollen, so besteht durch den „Cancel“-Knopf die Möglichkeit dazu. Dieser gesamte Vorgang kann beliebig oft wiederholt werden und die Aufzeichnung wird schließlich durch den Befehl `stopCollectingEvalPoints` beendet. Wird zu einem späteren Zeitpunkt wieder eine Aufzeichnung von Offline-Evaluierungspunkten mit einem statischen Nutzerprofil gestartet, so werden sie zu den bereits vorhandenen Evaluierungspunkten hinzugefügt. Die erstellte Datei hat die im folgenden Beispiel ersichtliche Struktur (Abbildung 4.2).

```
static user
10.1.2.41|11.07.2006 Tue 15:51:01.070|72.107|26.0889|4
|11.07.2006 Tue 15:51:01.683|awds|00:0F:B5:0F:9A:33|-75|EUK-LAB...
|11.07.2006 Tue 15:51:02.438|awds|00:0F:B5:0F:9A:33|-77|EUK-LAB...
...
10.1.2.41|11.07.2006 Tue 15:51:38.907|71.4165|24.6422|4
...
```

Abbildung 4.2: Struktur der Datei „evalPoints.txt“ am Beispiel (statisches Nutzerprofil)

In der ersten Zeile steht das Nutzerprofil. In der zweiten Zeile befinden sich die Positionsdaten über den ersten Offline-Evaluierungspunkt: Clientname, Zeitpunkt des Beginns der Aufnahme für diesen Punkt, Position (X- und Y-Koordinate) und die Ebene. In den darauf folgenden Zeilen befinden sich die zu diesem Punkt gehörenden Signalparameterwerte und die Zeitpunkte, wann sie übermittelt wurden. Kommt eine Zeile, die den gleichen

Aufbau hat wie Zeile 2, so beginnt dort der nächste Offline-Evaluierungspunkt. Jeder Position sind auf diese Weise eine bestimmte Menge von Signalparameterwerten zugeordnet. Somit ist der Inhalt der Datei eine Liste von unabhängigen Punkten.

Offline-Evaluierungspunkte-Ermittlung bei mobilem Nutzerprofil

Im Gegensatz zum statischen Nutzerprofil ist hier die Ähnlichkeit zu einem Training nicht mehr vorhanden. Die Aufzeichnung der Evaluierungspunkte wird durch Versenden des Befehls `firstEvalPoint` gestartet, nachdem der Nutzer eine Position und das Nutzerprofil ausgewählt hat. Der Server speichert diese Informationen und die Signalparameter, die vom Client geschickt werden. Wenn sich die Position des Clients ändert, wird seine aktuelle Position durch Klicken auf die Umgebungskarte an den Server übermittelt (Befehl: `moreEvalPoints`). Aufgrund des mobilen Nutzerprofils wird die Aufzeichnung dabei nicht unterbrochen. Durch Versenden des Befehls `stopCollectingEvalPoints` wird die Aufnahme abgeschlossen. Ein Weiterführen der Aufzeichnung zu einem späteren Zeitpunkt - wie beim statischen Nutzerprofil - ist hier nicht möglich. Stattdessen wird eine neue Aufzeichnung begonnen. Das Ergebnis der Aufnahme der Evaluierungspunkte wird in der Datei „evalPoints.txt“ gespeichert. Der strukturelle Aufbau ist in Abbildung 4.3 dargestellt.

```
mobile user
10.1.2.41
|14.07.2006 Fr 11:48:12.064|awds|00:0F:B5:0F:9A:13|-45|EUK-LAB...
|14.07.2006 Fr 11:48:12.812|awds|00:0F:B5:0F:9A:13|-55|EUK-LAB...
...
10.1.2.41|14.07.2006 Fr 11:48:42.014|43.5051|7.54144|4
|14.07.2006 Fr 11:48:42.313|awds|00:0F:B5:0F:9A:13|-54|EUK-LAB...
...
```

Abbildung 4.3: Struktur der Datei „evalPoints.txt“ am Beispiel (mobiles Nutzerprofil)

Am Anfang steht wiederum das Nutzerprofil. Darauf folgt der Clientname. Das ist notwendig, da beim mobilen Nutzerprofil als erster Schritt die bisher bereits vom Server erhaltenen Signalparameterwerte (Zeilen 3-5) gespeichert werden. Um diese später auszuwerten zu können, ist der Clientname notwendig. Diese gespeicherten Parameterwerte dienen als initiale Einträge der Datenstruktur `lastSignalValues`. Danach (Zeile 6) erscheint die erste Zeile mit einer Positionsangabe. Diese enthält die Daten aus dem Paket `firstEvalPoint` und hat den Aufbau: Clientname, Zeitpunkt der Positionsmel-

dung durch den Nutzer, Position (X- und Y-Koordinate) und die Ebene. Anschließend folgen wieder Signalparameterwerte mit den Zeitpunkten der Übermittlung. Die Speicherung weiterer Zeilen erfolgt abhängig von den empfangenen Daten, je nachdem, ob der Manager ein Paket mit dem Befehl `moreEvalPoints` oder der Client ein Paket mit Signalparametern schickt. Aufgrund des Nutzerprofils hat der Inhalt dieser Datei demzufolge einen kontinuierlichen Aufbau. Jeder Position sind alle davor stehenden Signalparameterwerte zugeordnet, nicht nur die, die zwischen zwei Positionsangaben stehen, wie es beim statischen Nutzerprofil der Fall ist. Bei der Lokalisierung werden dabei nur die letzten `secsForLocalization` Sekunden verwendet.

4.3.3 Berechnung der Evaluierung

In Kapitel 3.3.2 wurden die Online- und die Offline-Evaluierung vorgestellt. Beginnend mit der Online-Evaluierung werden diese beiden Varianten im Bezug auf ihre Implementierung behandelt.

Implementierung der Online-Evaluierung

Eine Online-Evaluierung besteht aus drei Arbeitsschritten, die wiederholt ausgeführt werden: Positionierung des Clients, Ortung des Clients und Bestimmung der Fehlerdistanz zwischen realer und errechneter Position. Die Angabe der realen Position erfolgt dabei auf die gleiche Art und Weise, wie bei der Benutzerkorrektur (siehe Kapitel 4.1.3). Anschließend wird die errechnete Position mit der vom Benutzer angegebenen Position verglichen und die Fehlerdistanz wird beim Manager in einer Datei gespeichert. Der Name der Datei ist abhängig vom Namen der Projekt-Datei. Lautet dieser Name beispielsweise „xyz.project“, so werden die Online-Evaluierungsdaten in der Datei „xyz.eval“ gespeichert. Sie haben dabei die in Abbildung 4.4 ersichtliche Struktur.

```
10.1.2.41|13.07.2006 Do 10:32:58.843|71.395|25.6025|4|70.6829|22.8656|0|4|-1|2.828  
10.1.2.41|13.07.2006 Do 10:33:57.500|71.403|23.1915|4|69.1529|23.851|0|4|-1|2.3448  
...
```

Abbildung 4.4: Struktur einer Evaluierungsdatei am Beispiel

Die während der Evaluierung ermittelten Daten werden zeilenweise in Textform gespeichert. Auf diese Weise können die Daten leicht ausgewertet und weiterverarbeitet werden. Jede Zeile repräsentiert dabei einen Evaluierungspunkt. Die Daten jeder Zeile dieser Datei

sind: Clientname, Zeitpunkt der Evaluierung, korrekte Position (X- und Y-Koordinate), korrekte Ebene, errechnete Position (X-, Y- und Z-Koordinate), errechnete Ebene, vom Algorithmus geschätzter Fehler (negative Zahlen bedeuten: keine Angabe einer Fehlerabschätzung), Fehlerdistanz.

Implementierung der Offline-Evaluierung

Bei einer Offline-Evaluierung werden zur Lokalisierung nicht die Signalparameter verwendet, die gerade von einem Client geschickt werden. Stattdessen wird auf bereits gespeicherte Werte zurückgegriffen. Diese Werte sind die Offline-Evaluierungspunkte, deren Erstellung in Kapitel 4.3.2 besprochen wurde.

Während der Offline-Evaluierung wird die Hash-Tabelle `lastSignalValues` mit den Signalparameterwerten der Offline-Evaluierungspunkte gefüllt und dem *Algorithm* zur Berechnung des Ortes zur Verfügung gestellt. Damit der *Algorithm* diese „alten“ Werte verarbeitet, wird ihm ebenfalls der Aufzeichnungszeitpunkt der Evaluierungspunkte als Anhaltspunkt mitgeteilt. Darum führt er die Berechnungen genau so aus, als wenn es gerade geschickte Signalparameter wären. Auf diese Weise ist keine Anpassung des *Algorithm* an die andere Bereitstellung der Daten erforderlich. Die ermittelte Position des Clients wird mit der Position des jeweiligen Offline-Evaluierungspunktes verglichen und dadurch die Fehlerdistanz bestimmt. Die einzelnen Daten und Ergebnisse werden in einer Datei gespeichert, deren Name sich aus folgenden Teilen zusammensetzt: „offlineEvalResult“, Name des Algorithmus, Nutzerprofil, Werte der Variablen `secsForTraining` und `secsForLocalization` und dem Suffix „eval“. Ein Beispiel dafür ist der Dateiname „offlineEvalResult-RADAR-static user-SFT20-SFL10.eval“. Durch die Namensgebung kann man die Resultate der Evaluierungen mit unterschiedlichen Algorithmen und Parametern voneinander unterscheiden. Diese Datei wird auch an den Manager versendet und dort gespeichert. Die Struktur dieser Datei ist - bis auf eine kleine Ausnahme - die gleiche, wie bei der Online-Evaluierung (siehe Abbildung 4.4). Der einzige Unterschied ist, dass am Ende der Datei eine Zeile „Time of calculation:“ folgt. Dort ist die Zeit in Millisekunden angegeben, die für die komplette Berechnung der Offline-Evaluierung benötigt wurde. Dadurch ist eine Abschätzung möglich, wie aufwändig die Berechnung war. Die Art der Verarbeitung der Daten aus der Datei „evalPoints.txt“ ist vom Nutzerprofil abhängig. Nun folgend werden beide Varianten näher erläutert.

Offline-Evaluierung bei statischem Nutzerprofil: Eine Position und die fest dazu gehörenden Signalparameterwerte bilden zusammen einen Offline-Evaluierungspunkt. Zuerst wird die reale Position (siehe Abbildung 4.2, Zeile 2) eingelesen und `lastSignalValues` mit allen folgenden Signalparameterwerten (Zeile 3-5) gefüllt. Danach kann die Berechnung des Ortes und der Fehlerdistanz ausgeführt werden. Abschließend wird die Hash-Tabelle wieder geleert, damit sie für den nächsten Evaluierungspunkt (Zeile 6) vorbereitet ist. Das heißt, dass beim statischen Nutzerprofil eine Berechnung der Fehlerdistanz ein mal pro Position durchgeführt wird.

Offline-Evaluierung bei mobilem Nutzerprofil: In der Hash-Tabelle `lastSignalValues` befinden sich zu Beginn der Offline-Evaluierung keine Daten. Sie wird durch die Werte gefüllt, die zu Beginn der Aufzeichnung der Offline-Evaluierungspunkte gespeichert wurden (siehe Abbildung 4.3, Zeile 3-5). Bei der ersten Ortsangabe (Zeile 6) wird auch die erste Ortsberechnung durchgeführt. Im Gegensatz zum statischen Nutzerprofil wird die Hash-Tabelle nach einer Berechnung nicht geleert, da die Signalparameterwerte nicht allein zu einem einzigen Ort zuzuordnen sind. Stattdessen wird die Datei zeilenweise ausgelesen und jede Zeile mit Signalparameterwerten wird der Hash-Tabelle hinzugefügt. Dann wird die Clientposition errechnet. Als reale Position für die Berechnung der Fehlerdistanz werden die Daten der zuletzt gelesenen Positionszeile verwendet. Beim mobilen Nutzerprofil wird eine Berechnung der Fehlerabweichung bei jedem vom Client gesendeten Signaleigenschaftspaket ausgeführt.

4.4 Sonstige Informationen

Die Implementierung der Evaluierungsumgebung erfolgte in C++ und Qt [18] (Version 4.1.4). Es wurde sowohl unter Windows XP, als auch unter Linux (Debian) programmiert und getestet. Die Evaluierungsumgebung ist demzufolge für beide Betriebssysteme geeignet. Insgesamt besteht das gesamte Projekt aus 44 Quelldateien mit insgesamt 7344 Zeilen (2367 Zeilen davon sind Leer- und Kommentarzeilen). Der Manager benötigt während der Benutzung auf einem Windows XP-Rechner zwischen 20 MegaByte und 30 MegaByte Arbeitsspeicher, wodurch es möglich ist, den Manager auf einem Gerät mit beschränkten Ressourcen einzusetzen.

Eine mit dieser Evaluierungsumgebung erstellte Topologie wurde für Tests verwendet, die

4 Implementierung

für die Erstellung eines Artikels [19] benötigt wurden. Die mit dieser Evaluierungsumgebung erstellbare Topologie ist demzufolge auch von anderen Applikationen nutzbar.

5 Experimentelle Bewertung der Evaluierungsumgebung

Dieses Kapitel dient dazu, die Funktionsfähigkeit der hier entwickelten Evaluierungsumgebung zu belegen. Um dies zu tun, sollen zwei Lokalisierungsalgorithmen evaluiert werden, welche unterschiedliche Anforderungen an die Evaluierungsumgebung stellen. Ziel ist es, zu beweisen, dass die Evaluierung erleichtert wird und die aussagekräftigen Ergebnisse mehrerer Lokalisierungsalgorithmen vergleichbar sind.

Im ersten Unterkapitel werden der generelle Aufbau der Testumgebung und die verwendete Hardware besprochen. Anschließend folgt die Evaluierung von RADAR [1] und danach werden die Testparameter verändert um zu sehen, welchen Einfluss sie auf die Leistungsfähigkeit des RADAR-Algorithmus haben. WCL [2, 3] wird dann evaluiert und abschließend werden die Ergebnisse beider Algorithmen verglichen.

5.1 Testumgebung und verwendete Hardware

Die für die Tests notwendigen Messungen wurden in der vierten Etage des Gebäudes 29 der Otto-von-Guericke-Universität Magdeburg durchgeführt. Die Gesamtfläche dieser Etage beträgt 2106m^2 ($78\text{m} \times 27\text{m}$), wovon jedoch nur annähernd die Hälfte ($35\text{m} \times 27\text{m}$, 945m^2) für die Messungen verwendet wurde. Für die Tests standen 9 APs (3x Netgear WGT634U, 6x Netgear WG602 v2) zur Verfügung, die an festen Positionen aufgestellt waren. 4 dieser APs waren durchgängig im gesamten Versuchsbereich empfangbar. Der Client, mit dem das Training absolviert wurde und die Evaluierungspunkte aufgezeichnet wurden, war ein Laptop (Dell Laptitude D410 mit 1GB RAM und einem Intel M 1,6 GHz-Prozessor) mit einer Orinoco Silver PCMCIA-Karte mit einer externen Antenne. Da dieser WLAN-Adapter eine minimale Signalstärke von -99 dBm verwendet, bekam die Variable `noConnectionValue` den Wert -100 . MagicMap [8, 9] (siehe Kapitel 2.4.2) verwendet

für die Messung der Signalstärke spezielle Software, die für diesen Test adaptiert wurde. Sie besteht aus dem Programm Netstumbler [20], das die Messung der Signalstärken der empfangbaren APs durchführt und einem Perl-Skript¹, das die gemessenen Werte an den Server versendet. Alle 750 Millisekunden schickte der Client ein Paket mit Signalstärkewerten. Die verwendeten APs, der Server und die Clients befanden sich im gleichen Subnetz, wodurch die Umsortierung von Paketen verhindert wurde, da nur ein möglicher Weg zwischen Client und Server existierte. Die Berechnungen der Offline-Evaluierung wurden auf einem Windows-XP-Rechner mit 512 MB RAM und einem AMD Athlon XP 1700+-Prozessor ausgeführt.

5.2 Evaluierung von RADAR

Um zu beweisen, dass die Evaluierungsumgebung korrekt arbeitet, muss damit die Evaluierung eines bereits bekannten Algorithmus durchgeführt werden. Die hier ermittelten Werte sollten den Werten ähnlich sein, welche die Entwickler angeben. Aus diesem Grund wurde der kartenbasierte Algorithmus RADAR [1] (siehe Kapitel 2.3.1) implementiert und mit allen dort beschriebenen und vergleichbaren Variationen getestet. Dadurch kann gezeigt werden, dass die entwickelte Umgebung für ein reales Evaluierungsszenario tauglich ist.

5.2.1 Versuchsaufbau

Als Grundlage für diese Tests dient der im RADAR-Artikel [1] (im Folgenden als „original“ bezeichnet) beschriebene Versuchsaufbau. Die Tests wurden auf einer Fläche von 43,50m x 22,50m (ca. 980m²) durchgeführt. Es waren 3 APs vorhanden und es wurden insgesamt 70 Trainingspunkte benutzt. An jedem Trainingspunkt wurden für jede der vier möglichen Orientierungen 20 Messwerte gespeichert, wobei alle 250 Millisekunden eine Messung durchgeführt wurde. Für die Tests mit einem mobilen Nutzerprofil verwendete man nur 10 Messwerte für die Lokalisierung. In Abbildung A.2 ist die Umgebungskarte der Originaluntersuchungen zu sehen.

Bei den hier gemachten Tests (im Folgenden als „aktuell“ bezeichnet) wurde versucht, diese Parameter so genau wie möglich zu kopieren. Aufgrund der etwas geringeren Fläche

¹Perl ist eine plattformunabhängige Programmiersprache, bei deren Entwicklung Teile von C, Pascal und UNIX-Shell mit einfließen.

(945m²), auf der das Training absolviert wurde, wurden statt 70 nur 67 Trainingspunkte für die Erstellung der Radiomap benutzt. Die Anzahl der APs wurde bei 3 belassen. Für das Training und die Lokalisierung wurden jeweils 15 Sekunden verwendet, da in dieser Zeit 20 Messwerte aufgezeichnet werden konnten. Die Umgebungskarte der aktuellen Tests ist in Abbildung A.1 dargestellt.

Der Abstand zwischen den einzelnen Signalstärkemessungen (RADAR original: 250ms, aktuelle Messungen: 750ms) spielt beim statischen Nutzerprofil keine Rolle, da die Position des Clients während der Messung konstant bleibt. Es dauert demzufolge einfach nur ein wenig länger, bis ein Trainings- oder Evaluierungspunkt aufgenommen ist. Bei einem mobilen Nutzerprofil hat dieser Unterschied jedoch große Auswirkungen, da sich der Client bewegt. Eine Verwendung von 10 Messwerten würde bedeuten, dass dazu 7,5 Sekunden in die Berechnung mit eingehen würden. Entgegen den von RADAR verwendeten 2,5 Sekunden ist dies das dreifache der Zeit. Die Position des Nutzers kann sich in der längeren Zeitspanne mehr verändern und das würde zu einem verfälschten Ergebnis gegenüber den Originalergebnissen führen. Im RADAR-Artikel wird jedoch beschrieben, dass beim statischen Nutzer bereits 3 Messwerte ausreichen, um annähernd das gleiche Ergebnis wie bei 20 Messwerten zu erreichen. Aus diesem Grund wurden die Tests mit dem mobilen Nutzerprofil bei unterschiedlichen Werten für die Lokalisierungszeit durchgeführt. Die Möglichkeit die Geschwindigkeit auf $\frac{1}{3}$ der bei den Originalmessungen verwendeten Geschwindigkeit zu reduzieren ist nicht durchführbar, da diese im RADAR-Artikel nicht angegeben ist.

5.2.2 Durchführung

Während der Lokalisierung des statischen Nutzers verwendeten die Entwickler von RADAR die NNSS-Technik (Nearest Neighbor(s) in Signal Space). Sie bestimmten anfänglich nur den nächsten Nachbarn und gaben dessen Position als errechnete Position aus. Später benutzen sie das Mittel der Positionen von mehreren (2-10) Nachbarn. Anschließend wurden nur die größten Signalstärkewerte eines APs je Messpunkt in die Berechnung mit einbezogen, um orientierungsunabhängige Werte zu erhalten. Abschließend wurde die Lokalisierungsleistung bei einem mobilen Nutzer getestet. In den aktuellen Tests wurden all diese Szenarien nachgestellt und die Ergebnisse mit den Originaldaten verglichen.

Um diese Tests durchführen zu können, wurde zuerst das Training absolviert. Durch die Variable `secsForTraining` besteht die Möglichkeit, weniger Messwerte für die Be-

rechnung der Radiomap zu verwenden, als während des Trainings aufgezeichnet wurden. Um auch mehr als nur die von den Entwicklern verwendeten 20 Messwerte je Trainingspunkt und Orientierung nutzen zu können, wurden statt der erwähnten 15 Sekunden 20 Sekunden aufgezeichnet. Insgesamt wurden 97 Trainingspunkte mit einem durchschnittlichen Abstand von 1,50m gesammelt. Anschließend wurden an 74 Positionen statische und an 87 Positionen mobile Offline-Evaluierungspunkte aufgenommen. Die Lage der einzelnen Evaluierungspunkte ist in den Abbildungen A.3 und A.4 zu sehen.

Aufgrund der Beschränkung auf 67 Trainingspunkte, wurden 30 der vorhandenen Trainingspunkte deaktiviert, indem ihr Status auf `forbidden` gesetzt wurde. Die Variablen `secsForTraining` und `secsForLocalization` erhielten anfänglich den Wert 15. Beim Test des mobilen Nutzerprofils betrug die Zeit für die Lokalisierung 2 bis 9 Sekunden. Die Liste der bekannten APs hat sich während des Trainings und Sammelns der Evaluierungspunkte gefüllt. Diese wurde editiert, sodass nur 3 APs für die Lokalisierung zugelassen waren. Nun wurde die Offline-Evaluierung mit allen oben beschriebenen Szenarien durchgeführt.

5.2.3 Ergebnisse

In diesem Unterkapitel sollen die Ergebnisse der Messungen beschrieben werden. Da sich bei der Durchführung an dem RADAR-Artikel [1] orientiert wurde, sind die Ergebnisse in der gleichen Reihenfolge dargestellt. Zuerst werden die Resultate der Messungen erläutert, bei der zur Berechnung nur ein Nachbar verwendet wurde. Anschließend werden mehrere Nachbarn benutzt. Danach werden ausschließlich die maximalen Signalstärkewerte betrachtet, wobei dort wiederum der Einfluss der Nachbaranzahl getestet wird. Am Ende erfolgt der Test mit einem mobilen Nutzerprofil.

Ein Nachbar: Ein Vergleich mit den originalen Resultaten von RADAR ist in Tabelle 5.1 zu sehen.

Die jetzigen Messungen sind den im RADAR-Artikel angegebenen ähnlich, da sie nicht mehr als 11% voneinander abweichen. Sie sind um durchschnittlich ca. 7% besser. Der Grund für die besseren Ergebnisse ist die andere Art der Ermittlung der Offline-Evaluierungspunkte. Bei den Originalmessungen wurden sie durch Isolierung von Trainingspunkten (siehe Kapitel 3.3.3) ermittelt. Dabei wird jeweils ein Trainingspunkt aus der Radiomap entfernt. Im RADAR-Artikel wird erwähnt, dass dies zu einer geringfügig

	Fehlerdistanz [Meter]		
	25%	50%	75%
Aktuelle Tests (ein Nachbar)	1,71	2,86	4,37
Originalergebnisse [1] (ein Nachbar)	1,92	2,94	4,69
Differenz	0,21 (10,9%)	0,08 (2,7%)	0,32 (6,8%)

Tabelle 5.1: RADAR: Vergleich der Testergebnisse mit einem Nachbar. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

schlechter angegebenen Genauigkeit führen kann, als RADAR in der normalen Benutzung wirklich hat. Die Tatsache, dass die aktuellen Ergebnisse geringfügig besser sind, unterstreicht die Effektivität und Verwendbarkeit der in Kapitel 3.3.3 gewählten Art der Offline-Evaluierungspunkte-Ermittlung.

Mehrere Nachbarn: In Tabelle 5.2 sind die Ergebnisse der Berechnungen in Abhängigkeit von der Anzahl der Nachbarn dargestellt.

Anzahl der Nachbarn	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
1	1,71	2,86	4,37	3,53	15,03
2	1,54	2,44	4,64	3,23	15,68
3	1,69	2,74	4,65	3,51	16,56
4	1,62	2,90	4,21	3,44	16,18
5	1,55	2,71	4,03	3,27	16,60
10	1,63	2,60	4,21	3,23	14,23

Tabelle 5.2: RADAR: Messergebnisse in Abhängigkeit von der Anzahl der Nachbarn. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

Im RADAR-Artikel wird berichtet, dass bei der Verwendung mehrerer Nachbarn die Ortungsgenauigkeit erhöht wird, wenn auch nur geringfügig. Bei den aktuellen Tests ist dieser Effekt ebenfalls bemerkbar (siehe Tabelle 5.3). Die Resultate verbessern sich um durchschnittlich ca. 7%. Bei den originalen Werten ist die Verbesserung mit rund 14% deutlicher ausgeprägt. Der Grund für diese höhere Genauigkeitssteigerung ist wiederum die Methode der Evaluierungspunkteermittlung. Da man für die Ortung nun mehrere Punkte (Nachbarn) verwendet hat, wurde der Effekt des gelöschten Trainingspunktes

teilweise aufgewogen. Bemerkenswert ist der geringe Unterschied zwischen den aktuellen und den originalen Ergebnissen bei 5 Nachbarn. Er beträgt nur 4 bzw. 5 Zentimeter.

	Fehlerdistanz [Meter]	
	25%	50%
Originalergebnisse [1] (ein Nachbar)	1,92	2,94
Originalergebnisse [1] (5 Nachbarn)	1,50	2,75
Verbesserung	0,42 (21,9%)	0,19 (6,5%)
Aktuelle Tests (ein Nachbar)	1,71	2,86
Aktuelle Tests (5 Nachbarn)	1,55	2,71
Verbesserung	0,16 (9,4%)	0,08 (5,2%)

Tabelle 5.3: RADAR: Vergleich der Testergebnisse mit variabler Anzahl von Nachbarn. Ein bestimmter Prozentsatz (25% oder 50%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

Maximale Signalstärke: Indem der Einfluss der Orientierung eliminiert wurde, sollte eine Verbesserung der Ortungsgenauigkeit erreicht werden. Dazu wurde die maximale Signalstärke über alle Orientierungen bestimmt. Das heißt, an jeder Position wurden nur die jeweils höchsten Signalstärkewerte der einzelnen APs aller 4 Orientierungen verwendet. Daraus wurde dann ein einziger Trainingspunkt erzeugt. Warum diese Verringerung der Informationsdichte zu einem besseren Ergebnis führen soll, wird im RADAR-Artikel [1] jedoch nicht erwähnt. Es wird lediglich geschrieben, dass sich die Genauigkeit bei der Verwendung mehrerer Nachbarn in Verbindung mit der Benutzung der maximalen Signalstärke verbessert (siehe folgenden Absatz). Eine Begründung warum die Resultate bereits bei der Verwendung eines einzigen Nachbarn besser sind, wird nicht gegeben. Es kann jedoch vermutet werden, dass diese Maßnahme dazu führt, dass der Zufall eine größere Rolle bei der Berechnung spielt. Dadurch kann es vorkommen, dass zufällig mehr Trainingspunkte mit sehr geringerer Fehlerdistanz als berechneter Ort gewählt werden. Genauso müssten auch mehr Punkte zufällig gewählt werden, die eine sehr hohe Fehlerdistanz haben. Dies könnte man dann an der 75%-Grenze, an der durchschnittlichen und an der maximalen Fehlerdistanz erkennen. Diese wurde jedoch im RADAR-Artikel nicht angegeben. Die schlechteren Ergebnisse bei der 50%-Grenze der aktuellen Messungen könnten bereits Ausdruck dieser Tatsache sein.

Der Effekt dieser Maßnahme ist in Tabelle 5.4 dargestellt. Bei den Originaldaten ist in beiden Kategorien eine leichte Verbesserung (ca. 8%) zu erkennen. Anders bei den aktuellen

	Fehlerdistanz [Meter]	
	25%	50%
Originalergebnisse [1]	1,92	2,94
Originalergebnisse [1] (maximale Signalstärke)	1,80	2,67
Verbesserung	0,12 (6,2%)	0,27 (9,2%)
Aktuelle Tests	1,71	2,86
Aktuelle Tests (maximale Signalstärke)	1,59	3,10
Verbesserung	0,12 (9,4%)	-0,24 (-8,4%)

Tabelle 5.4: RADAR: Vergleich der Testergebnisse bei Verwendung der maximalen Signalstärke. Ein bestimmter Prozentsatz (25% oder 50%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

Tests, wo sich der Wert nur bei der 25%-Spalte um 9,4% verbessert. In der 50%-Spalte ist eine Verschlechterung des Ergebnisses um 8,4% ersichtlich.

Mehrere Nachbarn & maximale Signalstärke: Die Entwickler von RADAR verbanden abschließend die Vorteile beider Verbesserungsvarianten. Sie benutzten nur die maximalen Signalstärkewerte und mittelten die Position über mehrere Nachbarn. Sie begründeten diese erneute Leistungssteigerung dadurch, dass die Nachbarn zwangsläufig unterschiedliche Positionen haben und dass dadurch der Mittelwert der Positionen aussagekräftiger ist. Dadurch erreichte RADAR die höchste Genauigkeit. In Abbildung 5.1 sind alle Messwerte präsentiert. Wenn zur Berechnung 2 oder 4 Nachbarn benutzt werden, sind die errechneten Punkte mit einer Wahrscheinlichkeit von 25% ca. einen Meter von der realen Position entfernt (Tiefpunkte der dunkelblauen Kurve).

Während die bis jetzt aufgelisteten aktuellen Messergebnisse den originalen Daten teilweise sehr ähnlich waren (siehe Tabelle 5.1 und 5.3), ist dies hier nicht mehr der Fall. Die Kurven haben zwar teilweise einen ähnlichen Verlauf, jedoch sind sowohl die Werte für die 25%-Grenze, als auch die Werte für die 50%-Grenze durchschnittlich 50 Zentimeter größer als bei den Originaldaten. Eine positive Auswirkung der Beschränkung auf die maximale Signalstärke ist hier nicht erkennbar.

Die Ursache dieser Abweichung liegt in der Lage und Verteilung der Trainingspunkte. Bei den aktuellen Tests waren die Trainingspunkte gleichmäßig auf der Fläche verteilt. Es waren sowohl auf dem Flur, als auch in den Räumen Trainingspunkte (siehe Abbildung A.1). Das war bei den Originalmessungen von RADAR anders. Dort bildeten mehr als 70%

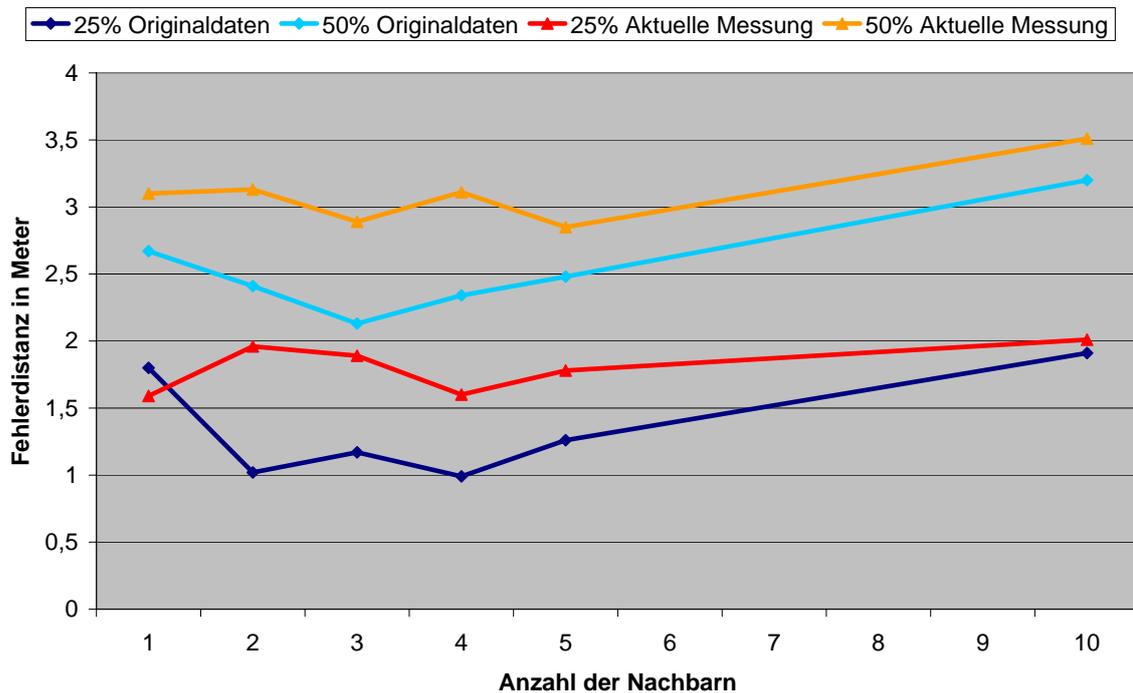


Abbildung 5.1: RADAR: Vergleich der Messergebnisse bei maximaler Signalstärke und variabler Anzahl von Nachbarn. Ein bestimmter Prozentsatz (25% oder 50%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

der Punkte auf einem Flur ein Rechteck, welches ebenfalls die äußere Grenze des trainierten Bereichs darstellte (siehe Abbildung A.2). Wurde nun ein Trainingspunkt für die Evaluierung gewählt, der am Rand des trainierten Bereichs lag, so war er Teil einer Seite dieses Rechteckes. Die ermittelten Nachbarn waren mit hoher Wahrscheinlichkeit ebenfalls Teil dieser Seite. Die dadurch errechnete Position befand sich demzufolge auch auf dieser Seite und blieb auf einer Höhe mit dem Rand des trainierten Bereiches. Aufgrund des rechtwinkligen Aufbaus war dadurch zumindest eine Koordinate der berechneten Position korrekt. Durch die gleichmäßige Verteilung der Trainingspunkte in den aktuellen Tests trat dieser Effekt nicht auf. Wurde hier ein Evaluierungspunkt am Rand des trainierten Bereiches gewählt (siehe Abbildung A.3), so befanden sich die nächsten Nachbarn nicht auf einer Linie mit ihm, so wie es bei den Originalmessungen der Fall war. Stattdessen lagen die Nachbarn weiter innen und haben demzufolge die errechnete Position ein wenig zum Mittelpunkt der Karte gezogen. Das führte zu einem schlechteren Ergebnis, als im RADAR-Artikel angegeben.

Mobiles Nutzerprofil: Die Offline-Evaluierungspunkte wurden bei einem Gang entlang des Flures aufgenommen (siehe Abbildung A.4). Für die Berechnung wurden ein Nachbar und alle Signalstärkewerte benutzt. Aufgrund der von den Originalmessungen abweichenden Sendehäufigkeiten wurde die Offline-Evaluierung mit verschiedenen Werten für `secsForLocalization` durchgeführt. Die einzelnen Messwerte können in Tabelle 5.5 betrachtet werden.

	Fehlerdistanz [Meter] 50% mobile user	Verschlechterung zu static user [Meter]
Originalergebnisse [1] (2,5s, 10 Werte)	3,50	0,56 (19%)
Aktuelle Tests (2s, 2-3 Werte)	3,64	0,78 (27,3%)
Aktuelle Tests (3s, 4 Werte)	3,36	0,50 (17,5%)
Aktuelle Tests (4s, 5-6 Werte)	2,66	-0,20 (-7%)
Aktuelle Tests (5s, 6-7 Werte)	2,50	-0,36 (-12,6%)
Aktuelle Tests (6s, 8 Werte)	2,66	-0,20 (-7%)
Aktuelle Tests (7s, 9-10 Werte)	2,68	-0,18 (-6,3%)
Aktuelle Tests (8s, 10-11 Werte)	2,73	-0,13 (-4,5%)
Aktuelle Tests (9s, 12 Werte)	2,83	-0,03 (-1%)

Tabelle 5.5: RADAR: Vergleich der Testergebnisse, mobiles Nutzerprofil. Ein bestimmter Prozentsatz (50%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt. In der letzten Spalte ist der Unterschied zum statischen Nutzer eingetragen.

Im RADAR-Artikel [1] verschlechterte sich die Ortungsgenauigkeit der 50%-Grenze von 2,94m auf 3,50m. Bei den jetzigen Messungen ist eine Verschlechterung nur dann zu bemerken, wenn zu wenig Werte verwendet wurden (Messungen mit 2s und 3s). Die Messung, bei der 5 Sekunden verwendet wurden, zeigt sogar eine deutliche Verbesserung gegenüber den Messungen des statischen Nutzerprofils. Ebenfalls bessere Werte zeigen die Messungen mit 4, 6, 7 und 8 Sekunden.

Die Ortung eines mobilen Nutzers ist schwieriger als die Ortung eines statischen Nutzers. Eine Verschlechterung der Ortungsgenauigkeit, ähnlich den Originalresultaten, wäre daher zu erwarten gewesen. Stattdessen sind die Werte sogar etwas besser (zum Vergleich: Tabelle 5.1). Diese Tatsache wird durch die Lage der Trainings- und Evaluierungspunkte begründet. Die statischen Offline-Evaluierungspunkte sind gleichmäßig auf die Fläche verteilt. Sie sind manchmal direkt auf, manchmal relativ nah oder manchmal relativ weit weg von einem Trainingspunkt. Es besteht also keine direkte Beziehung zwischen der Lage der Trainingspunkte und der Lage der Evaluierungspunkte. Anders ist das beim mo-

bilen Nutzerprofil. Aufgrund der Enge des Ganges, in dem die Evaluierungspunkte aufgezeichnet wurden, sind die meisten dieser Punkte sehr nah an einem Trainingspunkt. Da nur die Position des einen nächsten Nachbarn als errechneter Ort ausgegeben wird, ist dieser dadurch häufig nah am realen Ort. Ebenso beeinflusst die geringe Geschwindigkeit des Nutzers während der Messungen (ca. $0,4 \frac{m}{s}$) das Resultat positiv. Aus diesem Grund liefern selbst Messungen, die eine lange Zeitspanne in die Berechnung mit einbeziehen (siehe Zeile „Aktuelle Tests (9s, 12 Werte)“ in Tabelle 5.5), gute Werte.

5.2.4 Schlussfolgerung

Die Versuche mit RADAR zeigen, dass die Evaluierungsumgebung in der Lage ist, kartenbasierte Verfahren zu evaluieren. Sie ermöglicht die Aufnahme eines Trainings, die Lokalisierung von Clients und die Durchführung von Tests mit mobilem und statischem Nutzerprofil. Die Testparameter können komplikationslos verändert werden. Dies vereinfacht den Evaluierungsprozess deutlich. Die verschiedenen Versionen des Algorithmus konnten unter den exakt gleichen Bedingungen getestet werden, wodurch es möglich war, den Effekt der Änderungen zu analysieren. Die Evaluierung gestaltete sich zeitsparend, da die Aufnahme der Evaluierungspunkte nur ein einziges Mal absolviert werden musste. Beim Vergleich mit den Originalergebnissen der Entwickler von RADAR kann man feststellen, dass die Resultate nicht exakt übereinstimmen. Sie weisen jedoch eine gewisse Ähnlichkeit auf. Gerundet über alle absolvierten Messungen beträgt die durchschnittliche Abweichung von den Originalergebnissen nur 10%. Dadurch kann angenommen werden, dass die Ergebnisse korrekt sind und die Evaluierungsumgebung ordnungsgemäß arbeitet.

5.3 Optimierung von RADAR

In diesem Kapitel soll ausgelotet werden, welche Umgebungsparameter verändert werden müssen, damit RADAR noch genauer als bisher beschrieben, Ortsberechnungen durchführen kann.

5.3.1 Versuchsaufbau

Generell entspricht der Versuchsaufbau dem in Kapitel 5.2.1 beschriebenen Aufbau. Jedoch werden zu Versuchszwecken einige Parameter verändert: Anzahl der APs, Anzahl

der Trainingspunkte, Menge der Messwerte für das Training und für die Lokalisierung. Der Lokalisierungsalgorithmus selbst bleibt bei allen Tests unverändert. Es werden für die Messungen 2 Nachbarn und alle Signalstärkewerte benutzt, da bei diesem Szenario die besten Werte erreicht wurden (siehe Kapitel 5.2.3).

5.3.2 Durchführung

Für die Durchführung der Messungen werden die gleichen 97 Trainingspunkte und 74 statische Offline-Evaluierungspunkte verwendet, die bei der Evaluierung von RADAR (siehe Kapitel 5.2.2) erstellt wurden. Die Offline-Evaluierung wurde durchgeführt, wobei nur jeweils ein Parameter verändert wurde und der Rest der Parameter konstant blieb. Soweit nicht anders erwähnt, wurden 3 APs, 67 Trainingspunkte und 15 Sekunden je Trainingspunkt und Evaluierungspunkt verwendet. Die Messergebnisse, die mit diesen Parametern gemacht wurden, werden als „Standard“ bezeichnet.

Die Anzahl der APs (3, 6, 9) wurde zuerst verändert. Die Anzahl der Trainingspunkte (50, 67, 97) wurde im Anschluss daran variiert. Eine höhere oder niedrigere Menge an Messwerten je Trainingspunkt und Evaluierungspunkt (1, 5, 10, 15, 20 Sekunden) wurde dann untersucht. Am Ende folgt ein Test mit der verbesserten Konstellation der Testparameter, die in den vorigen Messungen ermittelt wurden. Dieser letzte Test wurde mittels der Offline- und Online-Evaluierung durchgeführt, wobei für letztere Variante an 60 Positionen die Fehlerabweichung von Hand gemessen wurde.

5.3.3 Ergebnisse

Im Gegensatz zu den Messungen in Kapitel 5.2, wo die Testparameter konstant blieben und stattdessen der Algorithmus angepasst wurde, ist das jetzt umgekehrt. Jetzt bleibt der Algorithmus unangetastet, es werden jedoch die Testparameter geändert, um ein möglichst gutes Ergebnis zu erhalten.

Anzahl der APs: In Tabelle 5.6 sind die Ergebnisse der Messungen mit unterschiedlicher Anzahl von APs zu sehen.

Es ist klar zu erkennen, dass die Verwendung von mehr Basisstationen auch zu einer höheren Genauigkeit führt. Dies ist durch die Informationssteigerung begründet, da die vom Client gesendeten Signalparameter Informationen von mehr APs enthalten. Alle Werte

Anzahl der APs	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
3 (Standard)	1,54	2,44	4,64	3,23	15,68
6	1,33	2,43	3,79	2,76	9,09
9	1,26	2,34	3,57	2,59	9,74

Tabelle 5.6: Optimierung von RADAR: Messergebnisse in Abhängigkeit von der Anzahl der APs. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

(mit Ausnahme der maximalen Fehlerdistanz bei 9 APs) sinken mit steigender AP-Anzahl. Die Verbesserung zwischen 6 und 9 APs (durchschnittlich ca. 3%) ist dabei deutlich kleiner, als zwischen 3 und 6 APs (durchschnittlich ca. 18%). Es ist daher abzusehen, dass noch mehr Basisstationen zu keiner signifikant besseren Ortungsgenauigkeit führen werden.

Anzahl der Trainingspunkte: Bei der Veränderung der Anzahl der Trainingspunkte ist das Resultat der Messungen widersprüchlich (siehe Tabelle 5.7).

Anzahl der Trainingspunkte	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
50	1,39	2,53	4,39	3,53	16,20
67 (Standard)	1,54	2,44	4,64	3,23	15,68
97	1,56	2,48	4,34	3,15	10,93

Tabelle 5.7: Optimierung von RADAR: Messergebnisse in Abhängigkeit von der Anzahl der Trainingspunkte. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

Mit steigender Anzahl sinken die durchschnittliche (um 10,5%) und maximale Fehlerdistanz (um 32,5%), was bedeutet, dass die Ortung allgemein genauer geworden ist. Jedoch sind die Werte in den Spalten der Fehlerabweichung mit 25%, 50% und 75% teilweise sogar geringfügig höher, wenn mehr Trainingspunkte verwendet werden. Mehr Trainingspunkte führen demzufolge dazu, dass besonders große Fehler eliminiert werden. Durch eine höhere Anzahl liegen deutlich mehr Punkte in der Nähe der realen Position. Dadurch ist es unwahrscheinlicher, dass ein Trainingspunkt zur Berechnung herangezogen wird, der eine große Fehlerdistanz hat. Die vielen Trainingspunkte sorgen jedoch nicht dafür,

dass kleine Fehlerabweichungen noch kleiner werden. Der positive Effekt des Gebrauchs von mehr Trainingspunkten ist vorhanden, jedoch ist er nur in den letzten beiden Spalten ersichtlich. Eine höhere Trainingspunkteanzahl ist demzufolge besser.

Menge der Messwerte je Trainingspunkt: In der Tabelle 5.8 ist ersichtlich, dass mit steigender Sekundenzahl die Ortung kontinuierlich besser wird.

Sekunden für Training	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
1	1,76	3,21	5,24	4,15	26,68
5	1,80	2,91	4,86	3,57	10,32
10	1,52	2,75	4,86	3,60	15,68
15 (Standard)	1,54	2,44	4,64	3,23	15,68
20	1,39	2,41	4,36	3,32	15,68

Tabelle 5.8: Optimierung von RADAR: Messergebnisse in Abhängigkeit von der Menge der Messwerte je Trainingspunkt. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

Weil für die Erzeugung eines Trainingspunktes mehr Werte zur Verfügung stehen, werden die natürlichen Schwankungen der Signalstärke teilweise ausgeglichen. Die Genauigkeitssteigerung zwischen den einzelnen Stufen ist zwar relativ gering, jedoch ist der Unterschied zwischen einer Sekunde und 20 Sekunden deutlich zu erkennen (durchschnittlich ca. 25% besser). Bei 20 Sekunden erreicht der Algorithmus in diesem Test die besten Werte. Die einzige Ausnahme bildet die maximale Fehlerabweichung, die bei 5 Sekunden mit 10,32m ihren Tiefpunkt hat, wobei alle höheren Sekundenzahlen die exakt gleiche maximale Abweichung (15,68m) aufweisen. Dieser konstante Wert in der Tabelle ergab sich durch den Umstand, dass bei allen drei Tests für den gleichen Evaluierungspunkt der gleiche Trainingspunkt als errechneter Ort ausgegeben wurde.

Menge der Messwerte je Evaluierungspunkt: Das Ergebnis in Tabelle 5.9 ähnelt dem Ergebnis aus der vorherigen Tabelle 5.8 sehr.

Das beste Resultat wird wiederum bei einer Sekundenzahl von 20 erreicht. Der Grund dafür ist abermals die Senkung des Einflusses der natürlichen Signalschwankungen durch die Verwendung vieler Messwerte. Die Steigerung der Ortungsgenauigkeit zwischen einer Sekunde und 20 Sekunden beträgt ca. 26%. Das ist fast der gleiche Wert, der auch bei der

Sekunden für Lokalisierung	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
1	1,57	2,60	6,48	4,08	22,62
5	1,90	3,00	5,54	3,85	12,45
10	1,66	2,52	5,12	3,69	15,68
15 (Standard)	1,54	2,44	4,64	3,23	15,68
20	1,49	2,41	4,19	3,04	9,69

Tabelle 5.9: Optimierung von RADAR: Messergebnisse in Abhängigkeit von der Menge der Messwerte je Evaluierungspunkt. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

Variation der Sekundenzahl für das Training erreicht wurde. Besonders bemerkenswert ist die starke Senkung des maximalen Fehlers um 57,2% von 22,62m auf 9,69m.

Verbesserte Konstellation: In den vorherigen Tests wurden einige Parameter verändert und ihre Auswirkungen auf die Ortungsgenauigkeit beobachtet. Dabei hat sich ergeben, dass die verbesserte Konstellation bei der Verwendung von 9 Basisstationen, 97 Trainingspunkten, 20 Sekunden für das Training und 20 Sekunden für die Lokalisierung liegt. Aus diesem Grund wurde der folgende Test mit diesen Testparametern durchgeführt. Es wurden sowohl die Offline-, als auch die Online-Evaluierung mit diesen Parametern durchgeführt. Die durch die veränderten Parameter erhöhte Genauigkeit gegenüber den Messungen bei Standardparametern ist deutlich in Tabelle 5.10 zu erkennen.

verwendete Parameter	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
Standard	1,54	2,44	4,64	3,23	15,68
Verbessert (Offline-Evaluierung)	1,27	2,06	3,19	2,34	7,05
Verbessert (Online-Evaluierung)	1,24	1,80	2,37	1,98	6,13

Tabelle 5.10: Optimierung von RADAR: Vergleich der Messergebnisse zwischen verbesserter Konstellation und Standard-Konstellation. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

Die Resultate sind auch besser als die Resultate der Messungen, bei denen nur ein Parameter verändert wurde. Somit wirken die positiven Effekte der einzelnen Veränderungen zusammen. Besonders sind die Werte für die durchschnittliche und maximale Fehlerdi-

stanz gesunken.

Auffällig ist, dass die Ergebnisse der Online-Evaluierung in allen Spalten besser sind, als die der Offline-Evaluierung. Bei der Durchführung der Online-Evaluierung ist dem Benutzer aufgefallen, dass er das Ergebnis positiv beeinflusst hat. Durch die Schwankungen der Signalstärke ist die errechnete Position selten konstant. Sie „springt“ stattdessen. Da der Zeitpunkt der Bewertung der errechneten Position nicht festgelegt war, konnte der Benutzer so lange warten bis die errechnete Position seinen Vorstellungen entsprach. Diese Vorstellung hat sich in der bereits absolvierten Offline-Evaluierung mit verbesserter Konstellation gebildet. Der Benutzer wollte die besonders guten Ergebnisse der Offline-Evaluierung bestätigen und hat demzufolge das Ergebnis positiv beeinflusst. Besonders ist dieser Effekt in der 75%-Spalte und bei der maximalen Fehlerdistanz zu erkennen. Bei einer großen Fehlerabweichung war der Benutzer bestrebt zu warten, bis eine „realere“ errechnete Position angezeigt wurde. Dieses Phänomen wird in der Psychologie „Versuchsleiter-Erwartungseffekt“ oder „Rosenthal-Effekt“² [21] genannt.

5.3.4 Schlussfolgerung

Die Tests haben gezeigt, dass die Genauigkeit von RADAR durch die Wahl der Testparameter sowohl positiv, als auch negativ beeinflusst werden kann. Es konnte ebenfalls gezeigt werden, dass die Testparameter einfach veränderbar sind. Somit kann mit der Evaluierungsumgebung sehr leicht eine aussagekräftige Evaluierung durchgeführt werden. Aufgrund der Testresultate konnte ebenfalls die Korrektheit der Online-Evaluierung verifiziert werden. Es hat sich jedoch herausgestellt, dass das Ergebnis der Online-Evaluierung durch den Benutzer beeinflusst werden kann. Diese starke Abhängigkeit vom Benutzer verringert die Brauchbarkeit. Dieser Effekt tritt bei der Offline-Evaluierung nicht auf. Das ist ein weiterer Pluspunkt, der die Leistungsfähigkeit der Offline-Evaluierung unterstreicht.

5.4 Evaluierung von WCL

Die Evaluierungsumgebung muss es ermöglichen, geometrische Verfahren verwenden zu können. Aus diesem Grund wurde WCL [2, 3] (siehe Kapitel 2.3.2) implementiert und getestet.

²Beim Versuchsleiter-Erwartungseffekt oder Rosenthal-Effekt handelt es sich um die unwissentliche Beeinflussung einer Datenerhebung durch den Untersucher selbst, aufgrund seiner Erwartungshaltung.

5.4.1 Versuchsaufbau

Die Tests für WCL wurden auf der gleichen Fläche ausgeführt, auf der auch RADAR getestet wurde. Alle 9 verfügbaren APs befanden sich dabei im Einsatz. Der Grund für die hohe Anzahl von APs ist, dass sich die berechneten Clientpositionen von WCL nur innerhalb der von den APs aufgespannten Fläche befinden können. Im späteren Verlauf der Tests wurden weniger als 9 Basisstationen verwendet.

5.4.2 Durchführung

Damit der Algorithmus arbeiten kann, müssen ihm die Positionen der APs bekannt sein. Diese werden ihm durch die Topologie zur Verfügung gestellt. Dazu wurde eine Projekt-Datei mit einer passenden Umgebungskarte geladen, worauf die Positionen der APs eingetragen wurden. Dies ist in Abbildung A.5 zu sehen. Die Offline-Evaluierungspunkte, die während der Evaluierung von RADAR aufgezeichnet wurden, wurden ebenfalls für WCL benutzt. Dadurch kann man die Ergebnisse der beiden Evaluierungen vergleichen und es müssen keine neuen Evaluierungspunkte aufgenommen werden. Bei der Offline-Evaluierung wurde die Anzahl der Sekunden für die Lokalisierung (`secsForLocalization`) variiert, um deren Einfluss zu beobachten. Sie wurde sowohl mit einem statischen, als auch einem mobilen Nutzer absolviert. Anschließend wurde die Online-Evaluierung eines statischen Nutzers an 58 Punkten durchgeführt. Später wurde beobachtet, wie es sich auswirkt, wenn man die Anzahl der APs verringert.

5.4.3 Ergebnisse

Sekunden für die Lokalisierung (statischer Nutzer): Die Messungen erbrachten die in Tabelle 5.11 aufgelisteten Werte.

Die relativ schlechten Ortungsergebnisse waren zu erwarten, da WCL ausschließlich die Positionen der APs verwendet und eine sehr einfache Berechnung durchführt. Es wäre jedoch zu erwarten gewesen, dass die Genauigkeit mit steigender Sekundenzahl zunimmt, weil dadurch die natürlichen Schwankungen der Signalstärke (siehe Kapitel 2.2.2) teilweise ausgeglichen werden. Entgegen dieser Annahme liegen die besten Ergebnisse bei einer Lokalisierungszeit von einer bzw. 5 Sekunden. Jedoch beträgt die Varianz der durchschnittlichen Fehlerabweichungen nicht mal 2% (9cm). Auch die Werte der Radien, in deren Kreise sich 25%, 50% oder 75% aller Punkte befinden, variieren nur geringfügig

Sekunden für Lokalisierung	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
1	5,01	6,78	8,32	6,85	15,98
5	5,14	6,60	8,40	6,85	15,57
10	4,98	6,73	8,48	6,94	15,77
15	4,96	6,75	8,51	6,93	15,93
20	5,09	6,72	8,53	6,92	15,82

Tabelle 5.11: WCL: Messergebnisse in Abhängigkeit von der Lokalisierungszeit bei statischem Nutzerprofil (9 APs). Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

(maximal 21cm). Die Werte der maximalen Fehlerabweichung sind ebenfalls relativ konstant. Sie weichen vom Durchschnitt (15,81m) nur maximal 24cm ab. Die Resultate sind sich so ähnlich, dass kein eindeutiger Effekt einer höheren oder niedrigeren Sekundenzahl erkennbar ist. Die hohe Zahl der APs ist der Grund dafür. Je mehr APs vorhanden sind, desto weniger Einfluss haben die Signalstärkewerte jedes einzelnen APs auf das Endergebnis. Somit werden die Schwankungen der Signalstärke kompensiert.

Sekunden für die Lokalisierung (mobiler Nutzer): Unter den gleichen Bedingungen wie die Tests mit dem statischen Nutzerprofil wurden auch die Messungen mit dem mobilen Nutzerprofil durchgeführt. Da unterschiedliche Evaluierungspunkte (siehe Abbildung A.4) verwendet wurden, sind die Ergebnisse nicht direkt miteinander vergleichbar. Es ist jedoch in allen Spalten zu erkennen, dass die Ortung mit steigender Sekundenzahl schlechter wird. Das ist auf die Strecke zurückzuführen, die während der Berechnungszeit zurückgelegt wurde. Je weniger Sekunden in die Berechnung mit einbezogen werden, desto geringer ist dieser Effekt.

Online-Evaluierung: Die Anzahl der Sekunden für die Lokalisierung wurde auf 20 festgelegt, um mögliche Schwankungen der Signalstärke auszugleichen. Dies ist zwar nicht nötig, wie in der Tabelle 5.11 zu sehen ist, jedoch hat es auch keine negativen Effekte. In Tabelle 5.13 sind die Ergebnisse zu sehen.

Die Ergebnisse der Online-Evaluierung sind denen der Offline-Evaluierung sehr ähnlich. Sie weichen durchschnittlich nur ca. 6% voneinander ab. Dabei sind die online ermittelten Werte leicht besser, als die Werte der Offline-Evaluierung. Nur die maximale Fehlerdistanz ist schlechter. Die Ursache für diese besseren Resultate ist wiederum bei dem Men-

Sekunden für Lokalisierung	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
1	3,68	6,50	9,75	7,00	19,26
5	4,33	6,09	9,87	7,14	17,77
10	4,99	6,18	9,93	7,32	17,63
15	5,07	6,24	10,15	7,49	17,51
20	5,64	6,64	10,15	7,68	17,68

Tabelle 5.12: WCL: Messergebnisse in Abhängigkeit von der Lokalisierungszeit bei mobilem Nutzerprofil (9 APs). Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
Online-Evaluierung	4,77	6,52	8,43	6,59	18,33
Offline-Evaluierung	5,09	6,72	8,53	6,92	15,82

Tabelle 5.13: WCL: Vergleich der Messergebnisse, Online- und Offline-Evaluierung bei statischem Nutzerprofil (9 APs). Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

schen selbst zu finden. Wie schon in Kapitel 5.3.3 beschrieben, hat der Benutzer einen meist positiven Einfluss auf das Ergebnis. Da die errechnete Position aufgrund des WCL-Algorithmus nicht so stark fluktuiert, kann der Benutzer das Ergebnis nicht in dem Maße positiv beeinflussen, wie es bei den Messungen für die Optimierung von RADAR der Fall war. Aus diesem Grund sind die Werte auch nur geringfügig besser.

Des Weiteren konnte man bei der Online-Evaluierung ein Phänomen erkennen, welches maßgeblich für die relativ schlechte Ortungsleistung verantwortlich sein könnte. Die errechneten Client-Positionen waren nicht gleichverteilt um die realen Positionen angeordnet. Sie befanden sich stattdessen stets zu weit in der Mitte der Umgebungskarte. Die Gründe dafür könnten in der Mitte der Umgebungskarte befindliche APs sein, deren zentrale Positionen das Resultat beeinflusst.

Deaktivierung von APs: In der Online-Evaluierung konnte beobachtet werden, dass die errechneten Positionen zu weit in der Mitte der Umgebungskarte lagen. Aus diesem Grund wurden 2 APs deaktiviert, um die These zu belegen, dass zentral gelegene APs das Ergebnis negativ beeinflussen. Diese deaktivierten Basisstationen lagen darum im Zen-

trum der Umgebungskarte (siehe Abbildung A.5). Es wurden nur 2 APs entfernt, weil die für die Lokalisierung wichtige Fläche, die von den APs aufgespannt wird, dadurch nicht beeinflusst wurde. In Tabelle 5.14 ist das Ergebnis zu sehen.

Sekunden für Lokalisierung	Fehlerdistanz [Meter]				
	25%	50%	75%	Durchschnitt	Maximal
1	3,90	5,27	7,08	5,83	17,36
5	3,92	5,72	7,21	5,87	16,76
10	3,79	5,61	7,38	5,95	16,49
15	3,79	5,55	7,24	5,95	17,15
20	3,71	5,46	7,20	5,93	16,93

Tabelle 5.14: WCL: Messergebnisse in Abhängigkeit von der Lokalisierungszeit bei statischem Nutzerprofil (7 APs). Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

Es ist eine deutliche Verbesserung der Genauigkeit abzulesen. Die durchschnittliche Fehlerabweichung und die Werte der Radien sind um ca. einen Meter geringer, als in Tabelle 5.11. Die maximale Fehlerdistanz ist jedoch um durchschnittlich 1,20m höher als bei der Verwendung aller 9 APs. Somit ist zu erkennen, dass die Lage und Anzahl der APs für diesen Algorithmus das entscheidende Kriterium ist.

5.4.4 Schlussfolgerung

Durch die Versuche mit WCL wurde gezeigt, dass die Evaluierungsumgebung in der Lage ist, geometrische Verfahren zu evaluieren. Es ist möglich, eine Topologie zu erstellen und Tests mit statischen und mobilen Nutzern durchzuführen. Die Schwäche des Verfahrens bei zentral gelegenen Basisstationen wurde durch die Online-Evaluierung entdeckt. Obwohl die Online-Evaluierung einige Nachteile hat, so wiegt dieser Erfolg ein wenig davon auf. Es war demzufolge vorteilhaft, sie zu implementieren.

5.5 Vergleich zwischen RADAR und WCL

Ein Vorteil dieser Evaluierungsumgebung ist, dass mit ihr verschiedeneste Lokalisierungsverfahren verglichen werden können. Das ist möglich, da eine Offline-Evaluierung durchgeführt wurde, bei welcher die Tests mit exakt den gleichen Werten gemacht wurden.

Dadurch ist ein direktes Gegenüberstellen der Ergebnisse und damit ein Vergleich der einzelnen Algorithmen möglich. Dies soll in diesem Kapitel gemacht werden.

5.5.1 Versuchsaufbau

Um den Vergleich eindeutig und reproduzierbar zu machen, werden für beide Algorithmen die gleichen 74 Evaluierungspunkte benutzt.

5.5.2 Durchführung

Es wurden insgesamt 2 Versuche durchgeführt. Für den ersten Test wurden die unveränderten Versionen der Algorithmen verwendet. RADAR arbeitete mit nur einem Nachbarn und allen Signalstärkewerten. Die Zeit für das Training betrug 15 Sekunden, wobei 67 Trainingspunkte verwendet wurden. Die einzigen beiden Testparameter, die bei beiden Algorithmen Einfluss haben sind die Zeit für die Lokalisierung (`secsForLocalization`) und die Anzahl der APs. Darum wurden 9 APs und 15 Sekunden für die Lokalisierung benutzt.

Als Zweites wurden die besten in den bisherigen Tests erreichten Ergebnisse miteinander verglichen. So kann man das Potenzial beider Algorithmen abschätzen. WCL arbeitete dazu ohne die 2 zentralen Basisstationen und 20 Sekunden für die Lokalisierung. RADAR hingegen rechnet mit 9 Basisstationen, 20 Sekunden für Training und Lokalisierung und 97 Trainingspunkten.

All diese Testparameter wurden mit den von der Evaluierungsumgebung zur Verfügung gestellten Mitteln festgelegt.

5.5.3 Ergebnisse

Unveränderte Versionen: In Tabelle 5.15 können die Resultate des ersten Tests begutachtet werden.

Deutlich zu erkennen ist, dass die Ortungsgenauigkeit von RADAR besser ist, als die von WCL. Aufgrund des simplen Algorithmus und der wenigen Informationen, die der WCL-Algorithmus verwendet, war das zu erwarten. Die über alle Fehlerdistanzkategorien gerundeten Werte von RADAR sind um 57% kleiner als die von WCL. Die mit 2444 Millisekunden geringfügig höhere Rechenzeit für alle 74 Evaluierungspunkte ist dabei

Algorithmus	Fehlerdistanz [Meter]					Rechenzeit [Millisekunden]
	25%	50%	75%	Durchschnitt	Maximal	
RADAR	1,43	2,33	4,00	2,88	9,74	2444
WCL	4,96	6,75	8,51	6,93	15,93	1763

Tabelle 5.15: RADAR und WCL: Vergleich der Messergebnisse, unveränderte Versionen. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

der einzige Wermutstropfen. Dieser wird jedoch durch die deutlich besseren Ergebnisse mehr als ausgeglichen.

Beste bisher erreichte Genauigkeit: Beide Algorithmen weisen nun bessere Resultate auf. Es ist in Tabelle 5.16 zu sehen, dass RADAR immer noch deutlich genauer ist als WCL. Aufgrund der schlechter gewordenen maximalen Fehlerabweichung bei WCL, sind die gerundeten Fehlerdistanzwerte über alle Spalten in diesem Versuch sogar um 60% kleiner. Aufgrund der etwas komplizierteren Berechnung und der Verarbeitung von mehr Werten beträgt die Rechenzeit bei RADAR nun ca. 3 Sekunden.

Algorithmus	Fehlerdistanz [Meter]					Rechenzeit [Millisekunden]
	25%	50%	75%	Durchschnitt	Maximal	
RADAR	1,27	2,06	3,19	2,34	7,05	3024
WCL	3,71	5,46	7,20	5,93	16,93	1802

Tabelle 5.16: RADAR und WCL: Vergleich der besten Messergebnisse. Ein bestimmter Prozentsatz (25%, 50% oder 75%) aller errechneten Positionen ist weniger als die angegebene Distanz von der realen Position entfernt.

5.5.4 Schlussfolgerung

Beide Tests haben gezeigt, dass RADAR der eindeutig bessere Algorithmus ist. Die einzige Ausnahme ist die Rechenzeit, die bei RADAR etwas höher ist als bei WCL. Beide Algorithmen konnten mit den gleichen Werten geprüft werden, wodurch der Vergleich aussagekräftig ist.

5.6 Zusammenfassung der Experimente

Im Verlauf des Kapitels 5 wurden mehrere Tests durchgeführt. Es wurden die Algorithmen RADAR und WCL sowohl getrennt evaluiert, als auch miteinander verglichen. Dabei wurde getestet, inwiefern die Evaluierung mit der hier entwickelten Evaluierungsumgebung möglich ist. Dabei hat sich herausgestellt, dass sie sowohl kartenbasierte, als auch geometrische Verfahren unterstützt. Die Evaluierungsumgebung ermöglicht die Erstellung eines Trainings und einer Topologie. Sie kann Clients lokalisieren und Tests mit mobilen und statischen Nutzern durchführen. Beide Varianten der Evaluierung (offline und online) ermöglichen eine vielschichtige Untersuchung der Algorithmen, wobei die Testparameter einfach zu ändern sind. Die Evaluierung von Lokalisierungsalgorithmen wird dadurch vereinfacht. Des Weiteren wird durch die Offline-Evaluierung eine zeitsparende Methode der Evaluierung zur Verfügung gestellt. Beim Vergleich mit den Originalergebnissen aus dem RADAR-Artikel [1] konnte die korrekte Arbeitsweise der Evaluierungsumgebung belegt werden.

6 Zusammenfassung und Ausblick

6.1 Zusammenfassung

In dieser Arbeit wurde die Entwicklung einer Evaluierungsumgebung für Lokalisierungsverfahren in drahtlosen Netzen beschrieben. Es sollte ermöglicht werden, verschiedene Verfahren evaluieren zu können. Dies sollte unter den gleichen Bedingungen für jeden Algorithmus geschehen, sodass die Resultate vergleichbar sind.

Dazu wurden in Kapitel 2 die Grundlagen für das Verständnis der Arbeit vermittelt. Es wurde besonders auf die Arbeitsweise von Lokalisierungsverfahren eingegangen und die Gründe dafür erläutert, warum die Ortung in drahtlosen Netzen so schwierig ist. Des Weiteren wurde erklärt, welche Einflüsse die Testumgebung und die Testparameter auf das Ergebnis einer Evaluierung haben.

Die Konzepte der Arbeit wurden dann in Kapitel 3 entwickelt. Es wurde zunächst die Liste der Anforderungen an die Evaluierungsumgebung erarbeitet. Dazu wurden mehrere bekannte Verfahren auf ihre Voraussetzungen hin analysiert. Damit die Evaluierungsumgebung möglichst viele Lokalisierungsverfahren unterstützen kann, mussten diese Voraussetzungen erfüllt werden. Das betraf besonders die Daten, die in der Topologie und im Training gespeichert wurden. Von besonderer Wichtigkeit war es dabei, dass jegliche Art von Signaleigenschaften zur Ortung benutzt werden konnten. Ebenfalls wurden besondere Features, wie die Benutzer-Korrektur, integriert. Um eine einfache Evaluierung zuzulassen, wurde die Evaluierungsumgebung in 3 Komponenten geteilt: Ein minimaler Client, der ausschließlich Signalparameter aufnimmt und an den Server schickt. Der Server führt mittels des integrierten Algorithmus die Ortung durch. Der Server ist dabei so aufgebaut, dass der Algorithmus, der evaluiert werden soll, komplikationslos ausgetauscht werden kann. Der Benutzer steuert den Server durch den Manager. An ihm wird außerdem die Topologie erstellt und das Training absolviert. Durch diese Komponententrennung wurde auch die Bedingung erfüllt, dass mehrere Clients lokalisiert werden können. Zur Erleich-

terung der Evaluierung wurde eine zentrale Speicherung der Daten eingerichtet. Die Evaluierungsumgebung ist ebenfalls in der Lage, viele Testparameter zu ändern und so einen vielschichtigen Test zu ermöglichen. Die wichtigste Eigenschaft der Evaluierungsumgebung ist jedoch die Tatsache, dass jeder Algorithmus mittels der exakt gleichen Daten getestet werden kann. Dies wird durch Offline-Evaluierungspunkte erreicht, die den von einem mobilen Gerät geschickten Signalparametern entsprechen.

Die Implementierung der Umgebung wurde in Kapitel 4 besprochen. Die Topologie wurde in einer XML-Datei gespeichert, die auch außerhalb der Evaluierungsumgebung verwendet und erstellt werden kann. Für die Steuerung des Servers und für den Informationsaustausch zwischen Server und Manager wurde eine Kommunikationsstruktur entworfen. Sie ermöglicht die Erstellung und Speicherung des Vorwissens, die Lokalisierung von Clients und die Evaluierung.

Schließlich wurde in Kapitel 5 die Evaluierung durchgeführt. Die beiden Algorithmen RADAR und WCL wurden dafür implementiert und getestet. Sie wurden in mehreren Variationen und unter anderen Testparametern evaluiert. Abschließend konnten die Ergebnisse miteinander verglichen werden.

Die Evaluierung hat gezeigt, dass die Evaluierungsumgebung korrekt arbeitet und die Evaluierung von Lokalisierungsverfahren erleichtert. Die Tests mit den zwei verschiedenen Algorithmen haben auch exemplarisch gezeigt, dass viele unterschiedliche Algorithmen unterstützt werden. Es ist sehr einfach, Testparameter zu ändern und somit den Algorithmus in all seinen Fassetten auszutesten. Der Evaluierungsprozess selbst ist sehr zeitsparend, da mit den Offline-Evaluierungspunkten eine wiederverwertbare Datenquelle erstellt wurde. Diese werden nur einmal aufgezeichnet und können danach jedem beliebigen Algorithmus zur Verfügung gestellt werden. Auf diese Weise können beliebige Verfahren miteinander verglichen werden.

Obwohl diese Umgebung maßgeblich zur Evaluierung entwickelt wurde, eignet sie sich darüber hinaus auch für die Entwicklung von Lokalisierungsverfahren. Das liegt daran, dass das ständige Prüfen der Leistungsfähigkeit ein maßgeblicher Teil des Entwicklungsprozesses ist.

6.2 Ausblick

Viele bisher entwickelte Lokalisationsverfahren arbeiten 2-dimensional. Der bisherige Entwicklungsstand, der hier entwickelten Evaluierungsumgebung, ist daher ebenfalls 2-dimensional. Eine Erweiterung auf 3 Dimensionen wäre zukunftsweisend. Einige Ansätze sind bereits vorhanden. So haben APs und die errechneten Clientpositionen bereits eine Z-Koordinate.

Während der Implementierung der Evaluierungsumgebung wurde aus Zeitgründen UDP als Transportprotokoll für die Kommunikation zwischen Client und Server verwendet, obwohl TCP die bessere Alternative wäre. Ein Gebrauch von TCP würde die korrekte zeitliche Einteilung und die garantierte Übertragung der Pakete sicherstellen. Die dafür notwendige Uhrensynchronisation müsste dazu implementiert werden. Da der Server die zentrale Komponente ist, mit der sowohl der Manager, als auch die Clients kommunizieren, müsste die Synchronisation von ihm ausgehen.

Es hat sich ebenfalls herausgestellt, dass das Ergebnis der Online-Evaluierung durch den Menschen beeinflusst werden kann. Dieses Manko könnte beseitigt werden, indem die Prozedur zur Angabe der realen Position verändert wird. Die Möglichkeit dem Nutzer die errechnete Position nicht anzuzeigen, um so eine Einflussnahme zu verhindern, ist dabei eine schlechte Wahl, da dadurch der Vorteil des direkten Kontakts mit dem Ergebnis der Berechnung weg fällt.

Es ist also zu erkennen, dass die Evaluierungsumgebung die Anforderungen erfüllt. Es ist aber durchaus Potenzial vorhanden, sie zu erweitern und an zukünftige Neuerungen anzupassen.

A Umgebungskarten

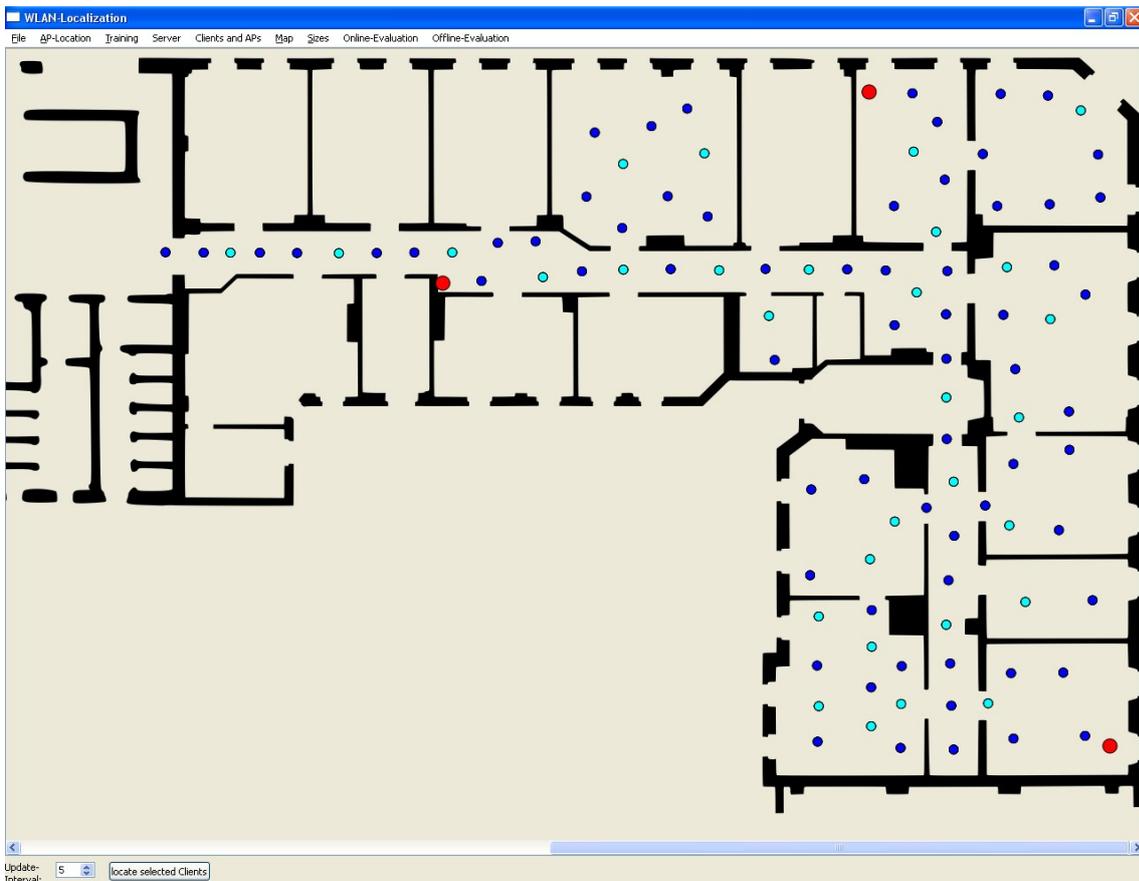


Abbildung A.1: Versuchsaufbau der aktuellen Tests von RADAR (APs und Trainingspunkte). Das ist ein Screenshot der Evaluierungsumgebung. Sie zeigt als Umgebungskarte den westlichen Teil der Etage 4 von Gebäude 29, Otto-von-Guericke-Universität Magdeburg. Die 3 verwendeten APs sind als rote Kreise dargestellt. Kreise in Blautönen markieren die Positionen von Trainingspunkten, wobei türkise den Status *forbidden* und dunkelblaue den Status *allowed* haben.

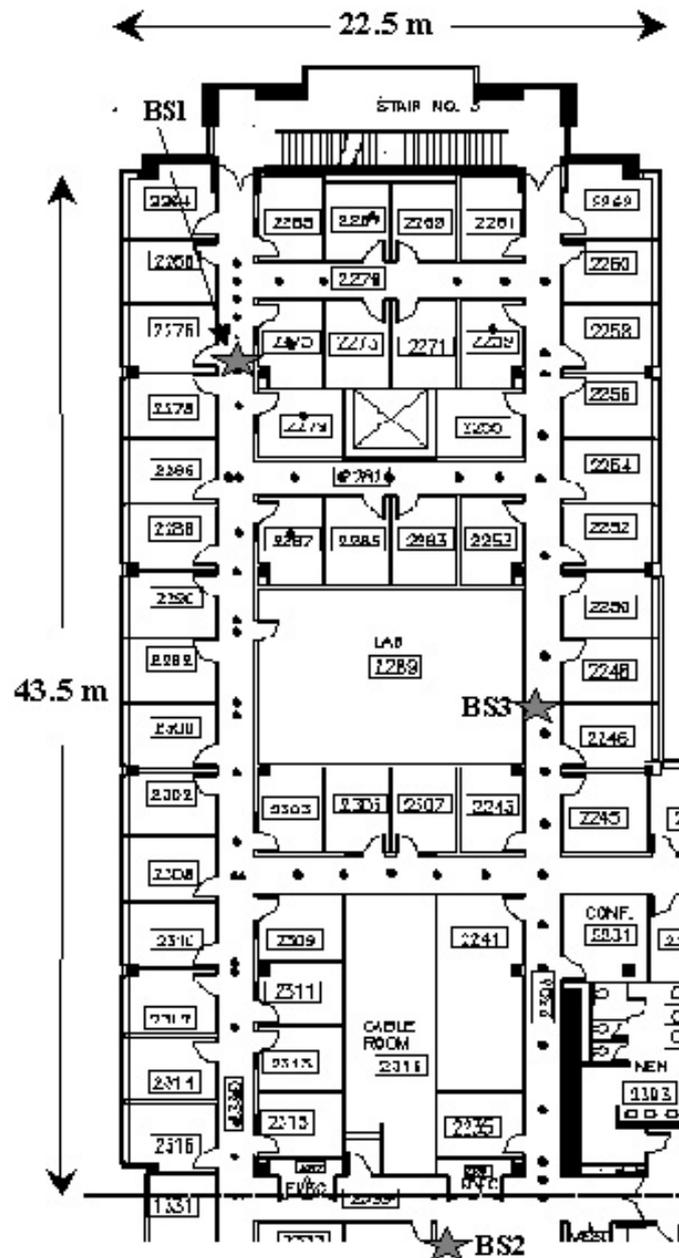


Abbildung A.2: Versuchsaufbau der originalen Tests von RADAR. Das ist der Gebäudeplan in dem RADAR von den Entwicklern getestet wurde. Die Trainingspunkte sind durch kleine schwarze Kreise gekennzeichnet. Die Basisstationen (BS1, BS2 und BS3) sind durch Sterne markiert.

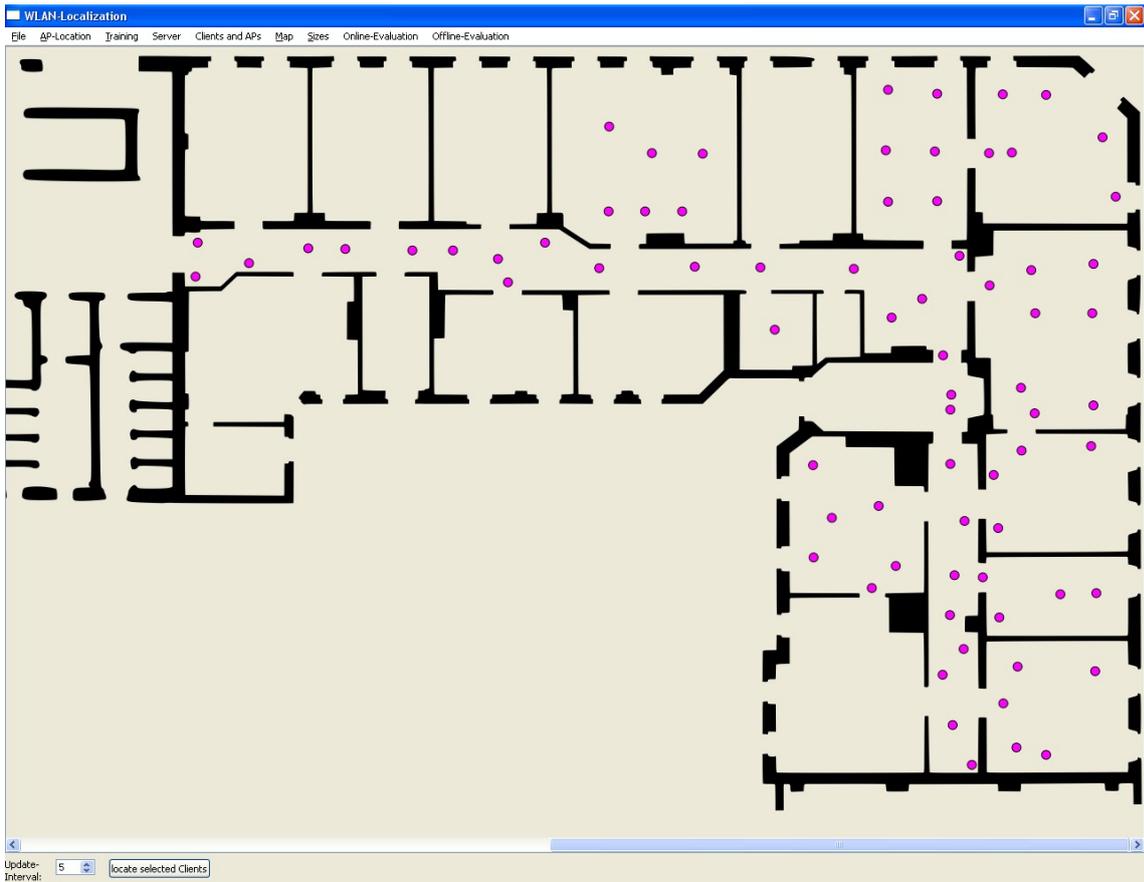


Abbildung A.3: Versuchsaufbau der aktuellen Tests (statische Offline-Evaluierungspunkte)

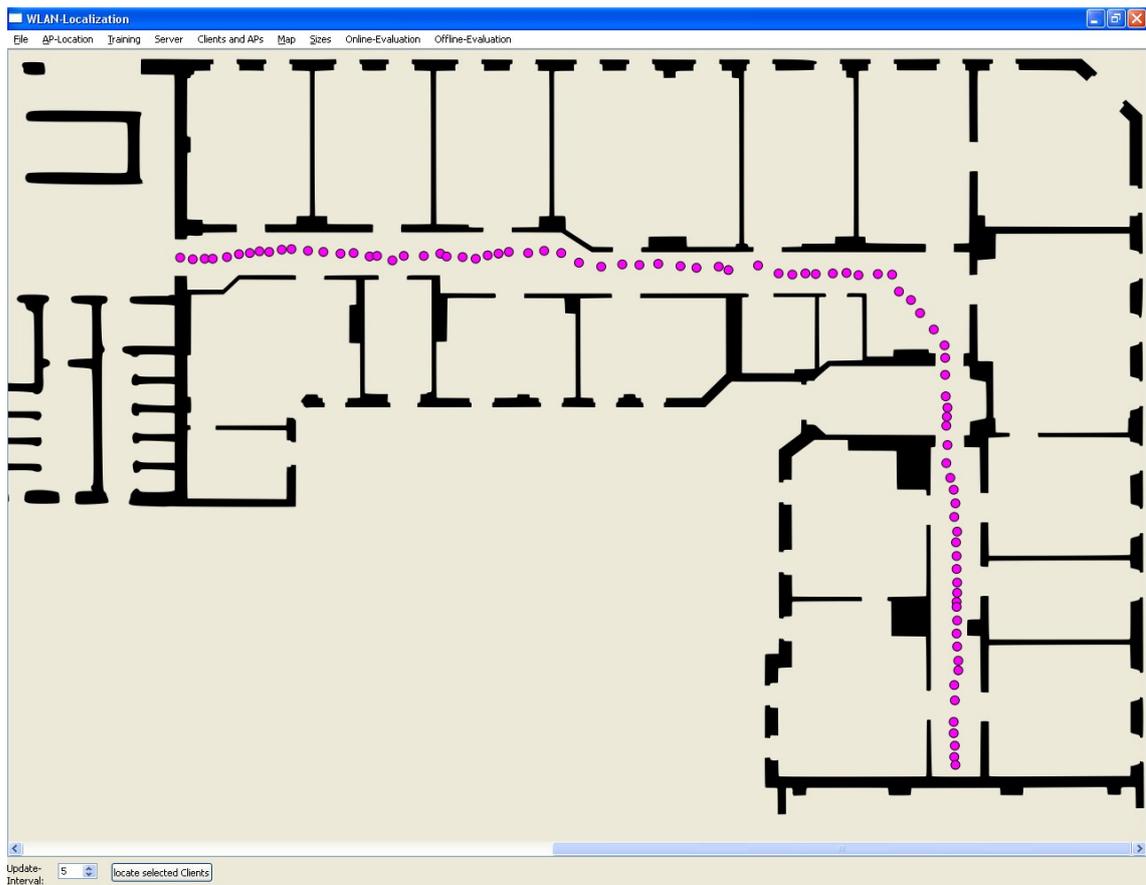


Abbildung A.4: Versuchsaufbau der aktuellen Tests (mobile Offline-Evaluierungspunkte)

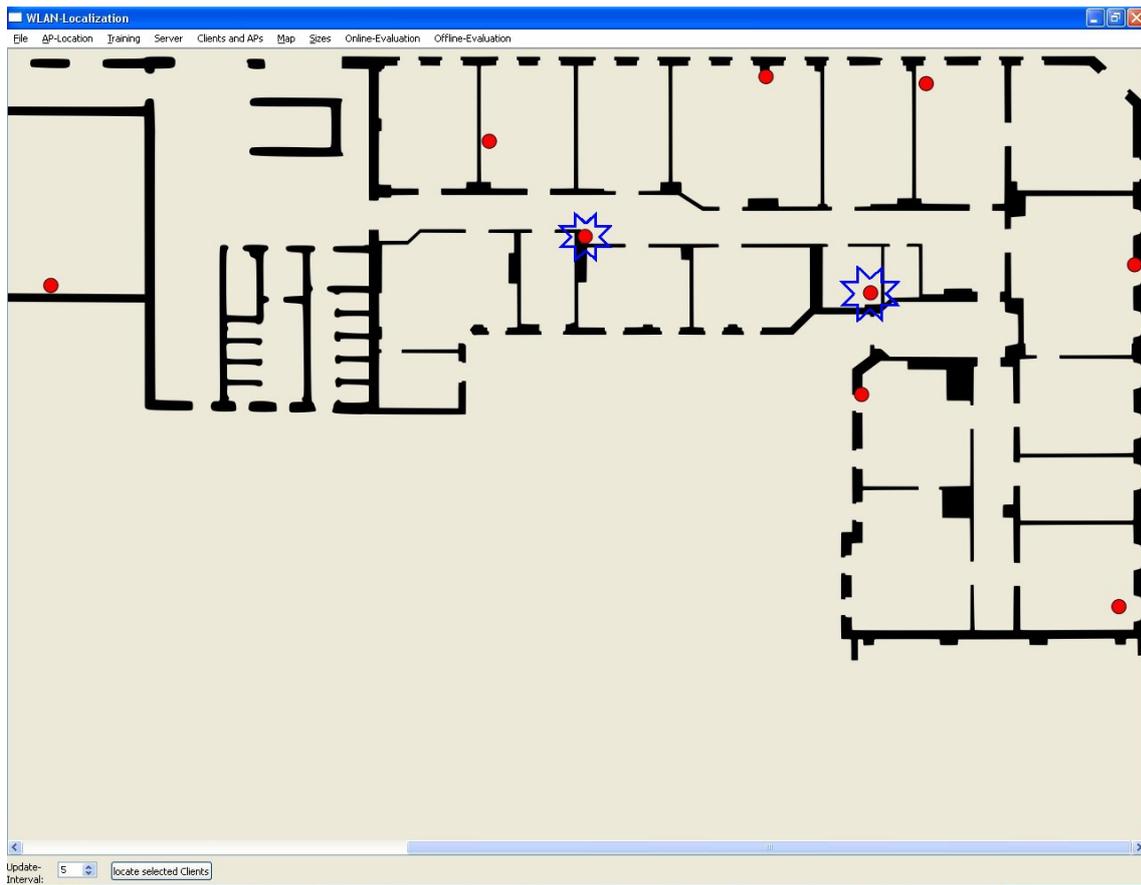


Abbildung A.5: Versuchsaufbau der aktuellen Tests für WCL (APs). Das ist ein Screenshot der Evaluierungsumgebung. Sie zeigt als Umgebungskarte den westlichen Teil der Etage 4 von Gebäude 29, Otto-von-Guericke-Universität Magdeburg. Die 9 verwendeten APs sind als rote Kreise dargestellt. Die beiden zentralen Basisstationen sind mit einem blauen Stern markiert.

Literaturverzeichnis

- [1] P. BAHL AND V. PADMANABHAN: „*RADAR: An in-building RFbased user location and tracking system.*“ In Proceedings of IEEE INFOCOM, volume 2, pages 775-784, March 2000
- [2] JAN BLUMENTHAL, FRANK REICHENBACH, DIRK TIMMERMANN: „*Precise Positioning with a Low Complexity Algorithm in Ad hoc Wireless Sensor Networks.*“ PIK -Praxis der Informationsverarbeitung und Kommunikation, Vol.28 (2005), Journal-Edition No. 2, S.80-85, ISBN: 3-598-01252-7, Saur Verlag, Germany, June 2005
- [3] JAN BLUMENTHAL, FRANK REICHENBACH, DIRK TIMMERMANN: „*Position Estimation in Ad hoc Wireless Sensor Networks with Low Complexity (Slides).*“ Proceedings of the 2nd Workshop on Positioning, Navigation and Communication 2005 (WPNC 05) and 1st Ultra-Wideband Expert Talk 2005 (UET 05)
- [4] ANDRÉ GÜNTHER AND CHRISTIAN HOENE: „*Measuring Round Trip Times to Determine the Distance between WLAN Nodes.*“ In Proc. of Networking 2005, Waterloo, Canada, May 2005
- [5] MOUSTAFA YOUSSEF, ASHOK AGRAWALA: „*The Horus WLAN Location Determination System.*“ In Proceedings of the 3rd international conference on Mobile systems, applications, and services, Seattle, Washington, 2005. Pages 205-218
- [6] M. YOUSSEF, A. AGRAWALA: „*Small-Scale Compensation for WLAN Location Determination Systems.*“ In IEEE WCNC 2003, March 2003.
- [7] JOE BARDWELL, VP OF PROFESSIONAL SERVICES: „*Converting Signal Strength Percentage to dBm Values.*“ Whitepaper, Wildpackets Inc., <http://www.wildpackets.com>, November 2002

- [8] PETER IBACH, TOBIAS HÜBNER, MARTIN SCHWEIGERT: „*MagicMap - Kooperative Positionsbestimmung über WLAN.*“ www.informatik.hu-berlin.de/rok/MagicMap, Chaos Communication Congress, Berlin, 27.-29. Dez. 2004
- [9] TOBIAS HÜBNER, MARTIN SCHWEIGERT: „*WLAN-basierte Ortung mit Magic-Map.*“ Studienarbeit, Humboldt Universität zu Berlin, 18. Mai 2005
- [10] E. S. BHASKER, S.W. BROWN, W. G. GRISWOLD: „*Employing user feedback for fast, accurate, low-maintenance geolocationing.*“ In Proceedings of IEEE PerCom 2004, Orlando, Florida, March 2004
- [11] MOUSTAFA YOUSSEF AND ASHOK AGRAWALA: „*On the Optimality of WLAN Location Determination Systems.*“ In Communication Networks and Distributed Systems Modelling and Simulation Conference, 2004
- [12] M. YOUSSEF, A. AGRAWALA, U. SHANKAR: „*WLAN location determination via clustering and probability distributions.*“ In Proceedings of IEEE PerCom2003, March 2003
- [13] EKAHAU, INC.: „*Ekahau Positioning Engine 3.1 User Guide.*“ <http://www.ekahau.com/>, 28.09.2005
- [14] SCOTT WHITE, JOSHUA O'MADADHAIN, DANYEL FISHER, YAN-BIAO BOEY: „*Java universal network/graph framework.*“ <http://jung.sourceforge.net> (19.06.2006)
- [15] PING TAO, ALGIS RUDYS, ANDREW M. LADD, DAN S. WALLACH: „*Wireless LAN Location Sensing for Security Applications.*“ In Proceedings of the 2003 ACM workshop on Wireless security, San Diego, CA, USA, (2003). Pages 11-20
- [16] JIE YIN, QIANG YANG, LIONEL NI: „*Adaptive Temporal Radio Maps for Indoor Location Estimation.*“ In Proceedings of the 3rd Annual IEEE International Conference of Pervasive Computing and Communications (IEEE PerCom 2005), Hawaii USA, March, 2005. Pages 85-94
- [17] XIAOYONG CHAI AND QIANG YANG: „*Reducing the Calibration Effort for Location Estimation Using Unlabeled Samples.*“ In Proceedings of the 3rd Annual

- IEEE International Conference on Pervasive Computing and Communications (IEEE PerCom 2005), Hawaii USA, March, 2005. Pages 95–104
- [18] TROLLTECH: „Qt“ <http://www.trolltech.com/>, 20.07.2006
- [19] ANDRÉ HERMS, GEORG LUKAS, AND SVILEN IVANOV: „*Realism in Design and Evaluation of Wireless Routing Protocols*“ To appear in First International Workshop on Mobile Services and Personalized Environments (MSPE'06), 16-17.11.2006, Aachen, Germany, „Lecture Notes in Informatics“, GI, 2006
- [20] „*Netstumbler 0.4.0*“ <http://www.netstumbler.com/>, 20.06.2006
- [21] SARRIS, VIKTOR: „*Methodologische Grundlagen der Experimentalpsychologie. 2: Versuchsplanung und Stadien.*“ München, Basel: Ernst Reinhardt Verlag, 1992

Selbstständigkeitserklärung

Ich versichere hiermit, dass ich die vorliegende Diplomarbeit „Evaluierungsumgebung für Lokalisierungsverfahren in drahtlosen Netzen“ selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Hagen Steinicke

Magdeburg, 21. September 2006