

Institut für Parallele und Verteilte Systeme

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Bachelorarbeit Nr. 59

# **Mechanismen zur Optimierung der Effizienz von Public-Sensing-Systemen**

Nikolai Neugebauer

<b>Studiengang:</b>	Softwaretechnik
<b>Prüfer:</b>	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
<b>Betreuer:</b>	Dipl.-Inf. Damian Philipp
<b>Beginn am:</b>	21.05.2013
<b>Beendet am:</b>	20.11.2013
<b>CR-Nummer:</b>	C.2.1, C.2.4





## Kurzfassung

Public Sensing hat sich zu einem wichtigen Forschungsgebiet in Bezug auf das großflächige Erfassen von Umgebungsphänomenen entwickelt. Begünstigt wird diese Entwicklung durch die immer größere Verbreitung von Smartphones. Smartphones sind heute mit GPS zur Positionsbestimmung, sowie mit UMTS und WLAN für die Kommunikation über das Internet ausgestattet. Außerdem haben Smartphones immer mehr Sensoren, die zur Erfassung von Umgebungsphänomenen genutzt werden können.

Public Sensing nutzt die Sensoren der Smartphones, um statische Sensornetzwerke durch Public-Sensing-Systeme zu ersetzen. Die Public-Sensing-Systeme nutzen für die Datenerfassung die Smartphones, welche im Untersuchungsgebiet unterwegs sind.

Probleme entstehen dadurch, dass die Besitzer der Smartphones ständig unterwegs sind. Folglich kann nicht zu jedem Zeitpunkt ein Messwert an einem bestimmten Ort aufgenommen werden. Weiterhin sollen die Besitzer der Smartphones durch die Beteiligung an Public-Sensing-Systemen möglichst wenig eingeschränkt werden. Durch die häufige Nutzung der Sensoren wird sehr viel Energie benötigt, wodurch der Akku zu oft geladen werden muss. Um dies zu verhindern und um die Genauigkeit der Daten zu erhöhen, werden in dieser Arbeit modellgetriebene Ansätze mit Verfahren der Mobilitätsvorhersage kombiniert.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>7</b>
<b>2</b>	<b>Grundlagen &amp; Verwandte Arbeiten</b>	<b>9</b>
2.1	Grundlagen . . . . .	9
2.2	Verwandte Arbeiten . . . . .	10
<b>3</b>	<b>Systemmodell und Anforderungen</b>	<b>13</b>
3.1	Systemmodell . . . . .	13
3.2	Anforderungen . . . . .	15
<b>4</b>	<b>Entwurf</b>	<b>17</b>
4.1	Ablauf einer naiven Anfrage . . . . .	17
4.2	Modellbasierter Ansatz . . . . .	17
4.3	Ausnutzung des kompletten Messzeitraums . . . . .	18
4.4	Optimierung der Kommunikation mit dem Gateway . . . . .	18
4.5	Reduzierung von Positionsbestimmungen . . . . .	19
4.6	WLAN Erweiterung für die Reduzierung von Positionsbestimmungen . . . . .	20
4.7	Modelloptimierung durch Mobilitätsvorhersage . . . . .	21
<b>5</b>	<b>Implementierung</b>	<b>23</b>
5.1	Simulationsumgebung . . . . .	23
5.2	Ablauf einer Simulation . . . . .	23
5.3	Aufbau des Public-Sensing-Systems . . . . .	24
5.4	Implementierte Optimierungen . . . . .	26
<b>6</b>	<b>Evaluation</b>	<b>29</b>
6.1	Methodologie . . . . .	29
6.2	Metriken . . . . .	29
6.3	Ergebnisse . . . . .	30
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>39</b>
7.1	Zusammenfassung . . . . .	39
7.2	Ausblick . . . . .	39
	<b>Literaturverzeichnis</b>	<b>41</b>

# Abbildungsverzeichnis

---

3.1	Übersicht über das Public-Sensing-System . . . . .	14
5.1	Wichtige Komponenten des Public-Sensig-Systems . . . . .	24
6.1	Anfragen ohne Ergebnis . . . . .	31
6.2	Erfolgreiche Anfragen . . . . .	32
6.3	Anfragen ohne Ergebnis . . . . .	34
6.4	Erfolgreiche Anfragen . . . . .	35
6.5	Erfolgreiche Anfragen . . . . .	36
6.6	Erfolgreiche Anfragen . . . . .	38

# 1 Einleitung

Public Sensing ist ein Ansatz, um Umgebungsphänomene in einem ausgewählten Messgebiet zu erfassen. Statt jedoch das Messgebiet mit fest installierten Sensoren auszustatten und ein statisches Sensornetzwerk aufzubauen, werden Smartphones genutzt, um Umgebungsphänomene zu erfassen. Zu erfassende Eigenschaften können zum Beispiel die Temperatur, der Lärmpegel oder die Helligkeit sein.

Smartphones entwickeln sich mehr und mehr zu Alltagsgegenständen. So tragen immer mehr Menschen ihr Smartphone ständig bei sich. Auch die Leistungsfähigkeit von Smartphones nimmt stetig zu. Neben der ständig steigenden Rechenleistung von Smartphones entwickelt sich auch die Anbindung der Smartphones ans Internet sehr schnell. So ist es heute problemlos möglich, Daten über UMTS an Smartphones zu übertragen, beziehungsweise vom Smartphone an einen Server zu schicken. Auf Grund dieser Entwicklungen bietet es sich an, Smartphones für verteilte Berechnungen heranzuziehen.

Des weiteren sind heutige Smartphones mit GPS ausgestattet. Mit Hilfe von GPS ist jederzeit eine sehr exakte Standortbestimmung möglich. Außerdem werden moderne Smartphones auch mit immer mehr Sensoren ausgestattet. Dank GPS und Sensoren können problemlos Umgebungsphänomene beobachtet und mit dem genauen Ort verknüpft werden. Durch die Anbindung an das Internet können diese Informationen jederzeit weitergeleitet und somit mit anderen geteilt werden. Durch diese Entwicklungen hat sich Public Sensing in den letzten Jahren zu einem großen Forschungsgebiet entwickelt.

Neben dem großen Vorteil, dass Messgebiete nicht erst aufwändig mit einem Sensornetzwerk ausgestattet werden müssen, ergeben sich durch Public Sensing aber auch eine Reihe neuer Probleme, für die Lösungen gefunden werden müssen. Eine große Herausforderung ist, dass der Ort von Smartphones, die an einem Public-Sensing-System teilnehmen, nie exakt bekannt ist. Dies liegt insbesondere daran, dass die Besitzer der Smartphones sich ständig bewegen. So kann es auch jederzeit sein, dass zu einem ausgewählten Zeitpunkt für einen bestimmten Messpunkt keine Messdaten vorliegen. Eine weitere Schwierigkeit ergibt sich dadurch, dass die ständige Nutzung von Sensoren, GPS und mobilem Internet sehr viel Energie benötigt. Da diese bei Smartphones von einem Akku geliefert wird, sollte möglichst wenig Energie verbraucht werden. Dies ist ein entscheidender Faktor für Besitzer von Smartphones, da sie sich kaum an einem Public-Sensing-System beteiligen würden, durch das sie ihr Handy nicht mehr wie gewohnt mindestens einen Tag lang nutzen können.

Um das Problem mit nicht verfügbaren Messwerten zu vermindern, wurden bereits verschiedene Ansätze verfolgt. Einer verfolgt das Ziel, Messwerte über ein Modell zu berechnen. Ein anderer Ansatz ist die ständige Bewegung der Smartphones zu nutzen, um zu einem späteren Zeitpunkt doch noch einen Messwert für einen Messpunkt zu erhalten. Mit Hilfe

von Mobilitätsvorhersagen über die Smartphones kann bereits zum Zeitpunkt der Anfrage berechnet werden, für welche Messpunkte keine Messung erfolgen kann. Diese Informationen können genutzt werden, um gezielt Alternativen zu planen.

Diese Arbeit kombiniert die drei oben genannten Ansätze. Eine Analyse erfolgt durch die Simulation in einem simulierten Public-Sensing-System. Die Ergebnisse werden insbesondere in Bezug auf die Zuverlässigkeit der Messergebnisse, sowie in Bezug auf ihre Energieeffizienz untersucht.

Im nächsten Kapitel werden Public Sensing, die modellgetriebene Erfassung und Verfahren zur Mobilitätsvorhersage in Public-Sensing-Systemen vorgestellt. Das dritte Kapitel widmet sich dem Systemmodell und den Anforderungen, die an das System gestellt werden. Im vierten Kapitel, dem Entwurf, wird die Realisierung der Anforderungen vorgestellt. Kapitel fünf geht auf die konkrete Implementierung des Entwurfs ein. Die Vorstellung und Analyse der Simulationsergebnisse werden im sechsten Kapitel behandelt. Mit Kapitel sieben folgen eine Zusammenfassung sowie Ausblicke auf weitere Optimierungsmöglichkeiten.

## 2 Grundlagen & Verwandte Arbeiten

### 2.1 Grundlagen

Public Sensing ist ein junges Phänomen, das erst durch die weite Verbreitung von Smartphones aufgekommen ist. [CLM<sup>+</sup>08]

In [CLM<sup>+</sup>08] werden verschiedene verwandte Begriffe definiert. Public Sensing wird in diesem Zusammenhang so definiert, dass die Messwerterfassung mit dem eigenen Smartphone zum Wohl der Allgemeinheit passiert. Social Sensing und Personal Sensing bilden den Gegenpol. So dient Personal Sensing den eigenen Interessen, wogegen Social Sensing individuell von einer Interessengemeinschaft verfolgt wird. Zusammengefasst werden Public-, Personal- und Social-Sensing als People-Centric-Sensing bezeichnet. [CLM<sup>+</sup>08]

Eine weitere Unterscheidung von Ansätzen im Bereich People-Centric-Sensing ist die Aufteilung in opportunistisches (opportunistic) und mitbestimmendes (participatory) Sensing.

Opportunistisches-Sensing zeichnet sich dadurch aus, dass die Erfassung von Messdaten automatisch erfolgt. Der Besitzer des Smartphones hat also keinen Einfluss darauf ob beziehungsweise wann ein Messwert aufgenommen wird. Die Herausforderung für solche Systeme ist die automatische Erkennung, in welchem Zustand sich das Smartphone gerade befindet. Nur wenn dies funktioniert, können zuverlässige Messwerte aufgenommen werden. [CLM<sup>+</sup>08]

Beim Mitbestimmenden-Sensing muss der Besitzer des Smartphones aktiv werden, um einen Messwert aufzunehmen. Der Besitzer der Messhardware kann also selbst entscheiden, wann Messwerte aufgenommen werden sollen und wann nicht. [CLM<sup>+</sup>08]

[LEM<sup>+</sup>08] beschäftigt sich mit der Frage, ob die Messdaten von opportunistischem oder von mitbestimmendem Sensing zuverlässiger sind. Sie kommen zu dem Schluss, dass opportunistisches Sensing für die großflächige Datenerfassung am sinnvollsten ist.

Damit ist Public Sensing in die Kategorie des opportunistischen-Sensing einzuordnen.

Die Forschung im Bereich People-Centric-Sensing erstreckt sich derzeit hauptsächlich auf zwei Gebiete. Zum einen werden konkrete Systeme entwickelt, die eine spezielle Implementierung verfolgen. Zum anderen werden theoretische Ansätze verfolgt, die Grundlagen für das automatisierte, zuverlässige und effiziente Erfassen von Messdaten, schaffen sollen.

MobGeoSen [KBP<sup>+</sup>08] ist beispielsweise eine konkrete Implementierung für eine Nokia Handy Serie. Das Programm ermöglicht es dem Benutzer seine Umgebung zu beobachten (Temperatur, Geräuschpegel, ...). Die Daten werden direkt auf dem Handy visualisiert.

Weitere Programme, die sich konkret auf das Erfassen bestimmter Daten mittels eines

bestimmten Systems befassen, sind zum Beispiel PollutionSpy, NoiseSpy und MobAsthma. [KBLR09]

Im Bereich der theoretischen Forschung wurden Arbeiten, wie z.B. [PSA<sup>+</sup>13], [PDR11] und [BDR12] veröffentlicht, deren Ziel es ist die Qualität der Messdaten zu verbessern und den Energieverbrauch zu senken.

Diese Thesis reiht sich in die Kategorie der theoretischen Arbeiten ein.

### 2.2 Verwandte Arbeiten

In [PDR11] wird eine sehr flexible Architektur für ein Public-Sensing-System beschrieben. Diese Architektur kapselt das eigentliche Public-Sensing-System hinter einer Benutzerschnittstelle. Diese Schnittstelle, das Gateway, verwaltet die Anfragen, die an das System gestellt werden und leitet sie an das eigentliche System, bestehend aus den Handys weiter. Außerdem bietet das Gateway eine einfache Möglichkeit an, um verschiedene Anfragen und Anfragearten über dasselbe System zu verwalten.

Der Entwurf des Public-Sensing-Systems in [PDR11] sieht die Definition von Messpunkten vor, um zu verhindern, dass Messpunkte an Orten genommen werden, die für das Ergebnis nicht relevant sind, zum Beispiel Messungen in geschlossenen Gebäuden.

[DGM<sup>+</sup>04] zeigt, dass mit Hilfe eines Modells auf ausgefallene Sensoren reagiert werden kann. Auch die Zahl der angefragten Sensoren kann durch die vorgestellte multivariante Gaußverteilung reduziert werden. Beides ist für Public-Sensing-Systeme sehr interessant. Zum einen kommen ausgefallene Sensoren in solchen Systemen sehr häufig vor. Ein ausgefallener Sensor entspricht einem Messpunkt, für den keine tatsächlichen Messwerte vorliegen. Eine solche Situation kann in Public-Sensing-Systemen sehr leicht auftreten, da die Messungen komplett von Smartphones abhängen, die sich frei bewegen.

Zum anderen kostet jede Anfrage an ein Smartphone Energie, die den Akku des Smartphones belastet. Weniger benötigte Anfragen sind also sehr interessant für Public-Sensing-Systeme.

Die Theorie, die in [DGM<sup>+</sup>04] statisch vorgestellt wird, wird in [PSA<sup>+</sup>13] auf ein Public-Sensing-System angewendet. Die Modellerstellung und -pflege erfolgt in Echtzeit auf Basis der eingeholten Messwerte. Die Technik wird genutzt, um die Zahl der anzufragenden Messpunkte zu senken.

In [LBD<sup>+</sup>05] wird aufgezeigt, dass durch mobile Sensoren ein größeres Messgebiet abgedeckt werden kann. Da die Smartphones sowieso mobil sind und sich ständig bewegen, kann man sich diese Eigenschaft zu Nutze machen, um in einem Public-Sensing-System mehr Messwerte zu erhalten. Dies wird von [BDR12] genutzt.

In [BDR12] verwaltet der Server ständig eine Übersicht über die Positionen der Smartphones und wählt auf dieser Basis einen Knoten aus, der vermutlich einen Messpunkt anfragen kann. Die Anfrage wird dabei mit einem Zeitpunkt versehen, zu dem das Smartphone vermutlich an dem Messpunkt ankommt. Es wird also die Mobilität der Smartphones genutzt, um Messwerte für bestimmte Messpunkte zu erhalten. Außerdem soll durch diese

Mobilitätsvorhersage des Servers die Zahl der Positionsbestimmungen und somit der Energieverbrauch reduziert werden.



## 3 Systemmodell und Anforderungen

In diesem Kapitel wird das Systemmodell vorgestellt. Außerdem werden die Anforderungen an die einzelnen Komponenten des Systems erörtert.

### 3.1 Systemmodell

Der Aufbau des Public-Sensing-Systems orientiert sich am Aufbau des Systems in [PSA<sup>+</sup>13]. Im Folgenden wird dieser Aufbau nochmals kurz vorgestellt.

Die zentralen Komponenten des Public-Sensing-Systems sind bei dieser Architektur das Gateway und die mobilen Knoten.

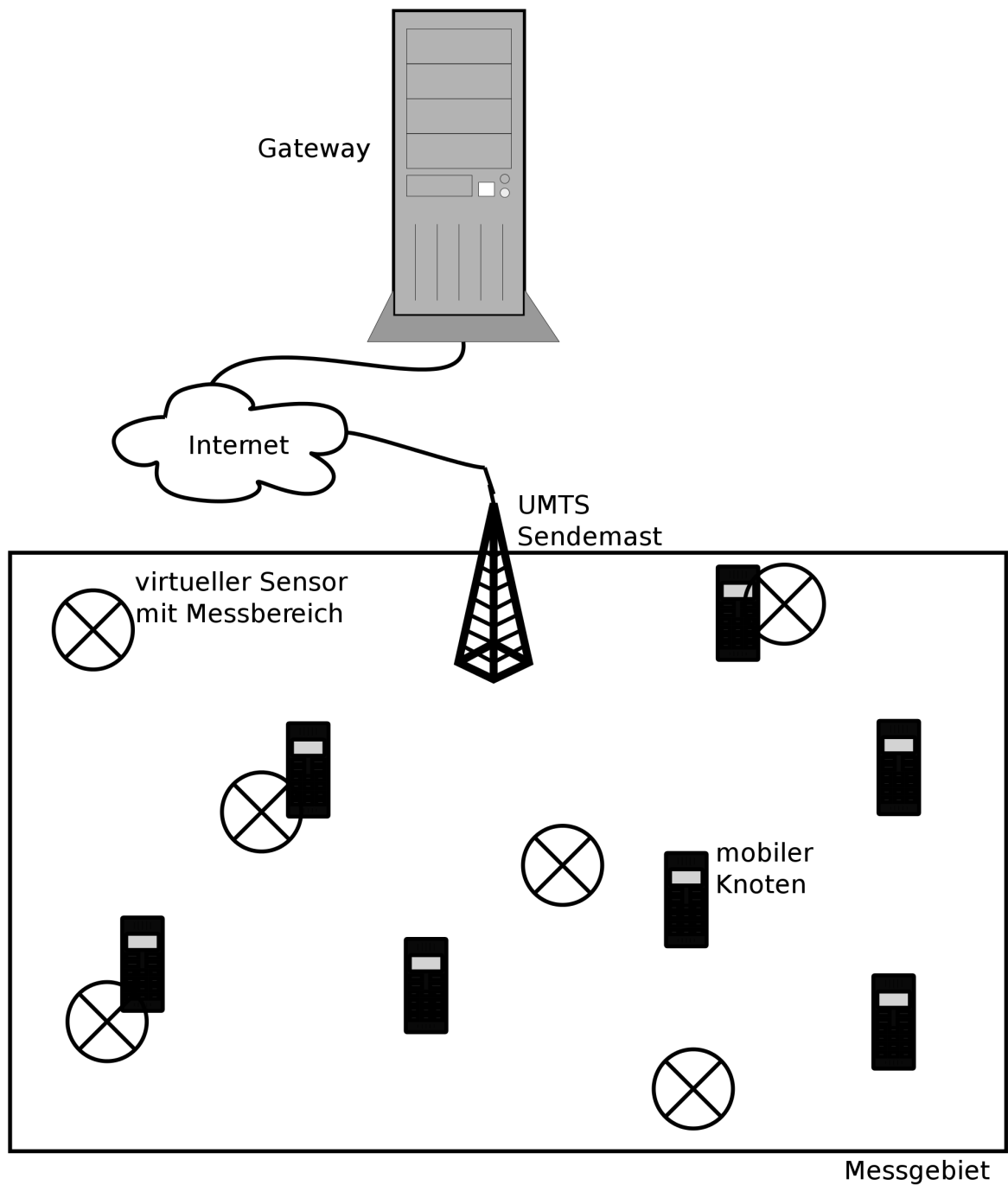
Das Gateway dient als Schnittstelle, um Anfragen an das Public-Sensing-System zu stellen. Die Anfragen werden vom Gateway verarbeitet. Hierfür generiert das Gateway Messaufträge für die mobilen Knoten. Die Messaufträge werden über das Internet an die mobilen Knoten gesendet. Außerdem empfängt das Gateway die Messungen der mobilen Knoten und wertet diese aus. Für Anfragen, die in regelmäßigen Abständen an die mobilen Knoten geschickt werden sollen, baut das Gateway ein Modell auf, um Messwerte vorausberechnen zu können.

Jeder mobile Knoten ist ein Smartphone, das im Messgebiet unterwegs ist. Alle mobilen Knoten verfügen über UMTS für die Kommunikation mit dem Gateway, WLAN für die Kommunikation mit den mobilen Knoten in der Nähe und GPS für die Positionsbestimmung. Außerdem besitzt jedes Smartphone die benötigte Hardware um geforderte Messwerte aufzunehmen.

Eine Anfrage an das Public-Sensing-System besteht aus einer Liste von Messpunkten. Jeder Messpunkt ist ein Ort im Messgebiet, für den Messungen aufgenommen werden sollen. Solche Punkte werden im Folgenden virtuelle Sensoren genannt. Zusätzlich definiert eine Anfrage Qualitätsansprüche, die von den Messungen erfüllt werden müssen. Eine Qualitätsanforderung ist insbesondere, wie groß der Abstand zum virtuellen Sensor für eine Messung sein darf (im folgenden „Messbereich“).

Ein Messauftrag für einen mobilen Knoten besteht aus ausgewählten virtuellen Sensoren, für die der mobile Knoten möglicherweise Messwerte aufnehmen kann, sowie einem Zeitraum, in dem die Messungen erfolgen müssen.

Abbildung 3.1 zeigt eine Übersicht über das System.



**Abbildung 3.1:** Übersicht über das Public-Sensing-System

## 3.2 Anforderungen

In Public-Sensing-Systemen gibt es eine sehr hohe Rate an virtuellen Sensoren, die für eine Anfrage keine Messwerte zurück liefern. Dies liegt daran, dass die mobilen Knoten sich frei im Messgebiet bewegen. Ein zuverlässiges Messergebnis ist also stark von der Dichte der mobilen Knoten im Messgebiet abhängig. Jedoch garantiert auch eine höhere Dichte an mobilen Knoten noch keine zuverlässigeren Ergebnisse, da sich die mobilen Knoten auch außerhalb des Messbereichs von virtuellen Sensoren befinden können.

Daraus leiten sich mehrere Anforderungen ab.

### **Reduzierung der angefragten virtuellen Sensoren**

Das Gateway soll möglichst schnell Zusammenhänge zwischen den virtuellen Sensoren erkennen. Daraus soll das Gateway ein Modell aufbauen, sodass weniger virtuelle Sensoren angefragt werden müssen. Durch das Modell können dann fehlende Messwerte berechnet werden.

### **Senkung der Ausfallwahrscheinlichkeit für virtuelle Sensoren**

Das Gateway soll nach Erstellung des Modells virtuelle Sensoren anfragen, die von mobilen Knoten erreicht werden können und soll die Werte nicht erreichbarer virtueller Sensoren berechnen. Hierfür ist es notwendig, dass das Gateway weiß in welchen Gebieten sich mobile Sensoren befinden.

Die mobilen Knoten sollen den Messzeitraum, den das Gateway ihnen zugesteht komplett nutzen. Da sie ständig in Bewegung sind, kann es gut sein, dass sie zum Zeitpunkt des Eintreffens der Messanfrage keinen Messwert eines virtuellen Sensors aufnehmen können, aber in wenigen Sekunden den Messbereich eines virtuellen Sensor erreichen.

Ein weiteres Problem beim Public Sensing ist es, dass die mobilen Knoten viele energieintensive Operationen durchführen müssen. Solche Operationen sind insbesondere die Kommunikation mit dem Gateway über UMTS und das ständige Abfragen von Positionsdaten via GPS.

Hieraus leiten sich folgende Anforderungen ab:

- Jeder mobile Knoten sammelt Informationen und überträgt so viele Informationen wie möglich auf einmal zum Gateway.
- Das Gateway sendet jedem mobilen Knoten nur Messanfragen für virtuelle Sensoren, die der mobile Knoten erreichen kann.
- Die mobilen Knoten prüfen nur dann ihre Position, wenn sie einen virtuellen Sensor erreichen können.
- Die mobilen Knoten informieren sich gegenseitig darüber, für welche virtuellen Sensoren bereits Messwerte aufgenommen wurden.



## 4 Entwurf

Dieses Kapitel beschreibt, welche Aufgaben die im Systemmodell vorgestellten Komponenten erfüllen müssen, um die an das System gestellten Anforderungen zu erfüllen. Zunächst wird der Ablauf einer naiven Anfrage an das Public-Sensing-System einschließlich der internen Schritte erläutert. Im Anschluss werden die aus [PSA<sup>+</sup>13] übernommenen Optimierungen kurz vorgestellt. Im weiteren Verlauf werden weitere Optimierungen eingeführt, um den im Kapitel Systemmodell und Anforderungen aufgeführten Anforderungen gerecht zu werden.

### 4.1 Ablauf einer naiven Anfrage

Der naive Ablauf einer Anfrage entspricht dem Ablauf einer Anfrage in [PSA<sup>+</sup>13] ohne Optimierungen. Das Gateway verwaltet die Anfragen an das Public-Sensing-System. Zum Zeitpunkt, zu dem für eine Anfrage eine Messung durchgeführt werden muss, generiert das Gateway einen Messauftrag für die mobilen Knoten. Dieser Messauftrag enthält eine Liste mit allen virtuellen Sensoren, sowie den Zeitraum, in dem die Messwerte wieder beim Gateway ankommen müssen. Der Messauftrag wird via Internet an jeden mobilen Knoten verschickt.

Die mobilen Knoten empfangen solche Messaufträge über UMTS. Empfängt ein mobiler Knoten einen Messauftrag, so prüft er, ob er für einen virtuellen Sensor einen Messwert aufnehmen kann. Ist dies der Fall, nimmt er den Messwert auf und sendet ihn über das Internet an das Gateway zurück.

Das Gateway sammelt die erhaltenen Ergebnisse und übernimmt die Messwerte als Ergebnis der Anfrage.

### 4.2 Modellbasierter Ansatz

[PSA<sup>+</sup>13] führt, um weniger virtuelle Sensoren anfragen zu müssen, einen modellbasierten Ansatz ein. Dabei werden zwei Phasen unterschieden.

#### 1. Lernphase

In der Lernphase werden wie beim naiven Ansatz alle virtuellen Sensoren der Anfrage in den Messauftrag übernommen. Allerdings werden die erhaltenen Messwerte vom Gateway zusätzlich ausgewertet. Dabei versucht das Gateway ein Modell zu entwickeln,

das Zusammenhänge zwischen den Messwerten der virtuellen Sensoren nutzt, um nur noch einen Teil der Sensoren anfragen zu müssen. [PSA<sup>+</sup>13]

### 2. Optimierungsphase

In der Optimierungsphase besteht ein Modell mit Zusammenhängen zwischen den virtuellen Sensoren. In die Messaufträge wird in dieser Phase nur noch ein Teil der virtuellen Sensoren aufgenommen. Dies sind zum einen die virtuellen Sensoren, für die kein Modell besteht. Zum anderen werden virtuelle Sensoren ausgewählt, aus denen Werte für nicht angefragte virtuelle Sensoren berechnet werden können. Zusätzlich werden Kontrollmessungen angefragt, um die Korrektheit des Modells zu verifizieren. Ist ein Modell nicht mehr korrekt, so geht das Gateway wieder in die Lernphase über. [PSA<sup>+</sup>13]

Die mobilen Knoten bearbeiten Messaufträge wie beim naiven Ansatz.

## 4.3 Ausnutzung des kompletten Messzeitraums

Das Gateway wartet mit der Auswertung der Messergebnisse grundsätzlich einen gewissen Zeitraum ab. Dieser Zeitraum liegt je nach Anfrage zwischen 10 und 30 Sekunden. Die mobilen Knoten nutzen diesen Zeitraum beim naiven Ansatz jedoch nicht aus. Bei der Verarbeitung des Messauftrags prüfen sie einmalig, ob sie im Messbereich eines virtuellen Sensors sind. Ist dies nicht der Fall, so verwerfen sie den Messauftrag sofort.

Eine Anforderung ist, die Senkung der Ausfallwahrscheinlichkeit für virtuelle Sensoren. Es soll also eine größere Anzahl der angefragten virtuellen Sensoren einen Messwert zurückliefern. Dies soll dadurch gewährleistet werden, dass die mobilen Knoten den vom Gateway zugestandenen Messzeitraum komplett ausnutzen, um Messwerte einzuholen. Jeder mobile Knoten prüft also nicht nur einmalig bei Erhalt des Messauftrags, ob er sich im Messbereich eines virtuellen Sensors befindet. Stattdessen prüft der mobile Knoten dies in regelmäßigen Abständen. Dies ist sinnvoll, da die mobilen Knoten sich ständig im Messgebiet bewegen. Im Gegensatz zur einmaligen Prüfung, ob sich der mobile Knoten im Messbereich eines virtuellen Sensors befindet, kann durch die regelmäßige Prüfung über den gesamten Messzeitraum ein größeres Gebiet abgedeckt werden. Hierdurch können Messwerte für virtuelle Sensoren im gesamten Gebiet aufgenommen werden, in dem sich der mobile Knoten im Messzeitraum bewegt.

Da jeder mobile Knoten in regelmäßigen Abständen prüft, ob er für einen virtuellen Sensor einen Messwert aufnehmen kann, könnte es vorkommen, dass ein mobiler Knoten für einen virtuellen Sensor mehrere Messwerte aufnimmt. Um diese Redundanz zu verhindern, nimmt jeder mobile Knoten nur einmalig eine Messung für einen virtuellen Sensor auf.

## 4.4 Optimierung der Kommunikation mit dem Gateway

Jeder mobile Knoten kann durch die Ausnutzung des kompletten Messzeitraums Messwerte für mehrere virtuelle Sensoren aufnehmen. Nach dem Ansatz, der in Ausnutzung

des kompletten Messzeitraums beschrieben wird, wird bei jedem neu aufgenommenen Messwert eine Nachricht mit dem Messwert an das Gateway versendet. Jeder Versand einer Nachricht benötigt jedoch Energie. Da in dieser Arbeit eine möglichst energieeffiziente Lösung angestrebt wird, soll die Zahl der Nachrichten an das Gateway reduziert werden. Hierzu sammelt jeder mobile Knoten seine aufgenommenen Messungen. Diese werden kurz vor Ende des Messzeitraums gesammelt an das Gateway gesendet. Hierbei muss der Zeitpunkt des Versands so gewählt werden, dass die Messungen mit großer Wahrscheinlichkeit noch am Gateway ankommen. Der Zeitpunkt darf also nicht zu nah am Ende des Messzeitraums liegen, da sonst der Verlust aller Messdaten droht, weil diese dann nicht rechtzeitig am Gateway ankommen. Um Messungen, die nach dem Versand der gesammelten Daten aufgenommen werden, auch noch ans Gateway zu übermitteln, werden diese direkt nach ihrer Aufnahme verschickt. Damit hierbei allerdings nicht zu viele Messungen einzeln verschickt werden, darf der Zeitpunkt, zu dem die gesammelten Daten an das Gateway geschickt werden, nicht zu früh gewählt werden.

## 4.5 Reduzierung von Positionsbestimmungen

Im Abschnitt Ausnutzung des kompletten Messzeitraums bestimmen die mobilen Knoten in regelmäßigen Abständen ihren Ort, um zu prüfen, ob sie Messungen aufnehmen können. Da Positionsbestimmungen Energie benötigen, sollen unnötige Positionsbestimmungen vermieden werden. Gleichzeitig soll jedoch gewährleistet sein, dass für jeden virtuellen Sensor, für den ein Messwert aufgenommen werden kann, auch ein Messwert aufgenommen wird. Um dies zu gewährleisten, wird die Verarbeitung von Messaufträgen von den mobilen Knoten optimiert.

Erhält ein mobiler Knoten einen Messauftrag, so prüft er für jeden im Messauftrag enthaltenen virtuellen Sensor, ob er sich im Messbereich des virtuellen Sensors befindet.

Kann für einen virtuellen Sensor ein Messwert aufgenommen werden, so wird ein Messwert erfasst und gespeichert, wie in Optimierung der Kommunikation mit dem Gateway beschrieben. Der mobile Knoten löscht den virtuellen Sensor dann aus seinem Messauftrag. Kann kein Messwert aufgenommen werden, so berechnet der mobile Knoten, aufgrund seines euklidischen Abstands zum virtuellen Sensor, sowie seiner Bewegungsgeschwindigkeit, den Zeitpunkt, zu dem er den Messbereich erreichen kann. Kann er den Messbereich nicht innerhalb des, vom Gateway zugestandenen Messzeitraums, erreichen, so entfernt der mobile Knoten den virtuellen Sensor aus seinem Messauftrag. Die nächste Positionsbestimmung wird auf den Zeitpunkt verzögert, zu dem frühestens ein virtueller Sensor erreicht werden kann. Zu diesem Zeitpunkt wiederholt der mobile Knoten den gerade beschriebenen Algorithmus; er prüft also alle virtuellen Sensoren, entfernt nicht erreichbare virtuelle Sensoren, nimmt gegebenenfalls Messwerte auf und berechnet den Zeitpunkt für die nächste Positionsbestimmung.

Ein mobiler Knoten verwirft einen Messauftrag, wenn keine virtuellen Sensoren, mehr im Messauftrag enthalten sind, oder wenn das Ende des Messzeitraums vor dem Zeitpunkt der nächsten berechneten Positionsbestimmung erreicht ist.

Bei der Übermittlung von Messwerten an das Gateway wird zunächst wie im Abschnitt

Optimierung der Kommunikation mit dem Gateway, ein Zeitpunkt festgelegt, zu dem gesammelte Messdaten übermittelt werden. Wann das Versenden der Messdaten erfolgt, ist jedoch von verschiedenen Faktoren abhängig.

### **Im Messauftrag verbleiben keine virtuellen Sensoren mehr.**

Der mobile Knoten versendet in diesem Fall sofort alle bisher aufgenommenen Messungen, um den Messauftrag verwerfen zu können.

### **Im Messauftrag verbleiben noch virtuelle Sensoren, die nächste Positionsbestimmung findet allerdings nach dem Zeitpunkt, zu dem die gesammelten Messwerte verschickt werden statt.**

In diesem Fall werden ebenfalls alle bisher aufgenommenen Messwerte sofort versendet, damit die nächste Aktion für den Messauftrag auf den Zeitpunkt der nächsten Positionsbestimmung verschoben werden kann. Außerdem wird so sichergestellt, dass Messungen, die nach dem Zeitpunkt der Gatewaykommunikation gemacht wurden, sofort an das Gateway versendet werden (vergleiche Optimierung der Kommunikation mit dem Gateway).

### **Im Messauftrag verbleiben noch virtuelle Sensoren, die nächste Positionsbestimmung ist vor dem Zeitpunkt, zu dem die gesammelten Messwerte verschickt werden.**

In diesem Fall erfolgt keine Kommunikation mit dem Gateway.

## **4.6 WLAN Erweiterung für die Reduzierung von Positionsbestimmungen**

Durch die im vorherigen Abschnitt vorgestellte Optimierung hat jeder mobile Knoten für sich versucht die Zahl der Positionsbestimmungen zu reduzieren. Die in diesem Kapitel vorgestellte Erweiterung soll die Zahl der Positionsbestimmungen weiter reduzieren, indem die mobilen Knoten Informationen über Messungen austauschen.

Für die hier vorgestellte Optimierung wird es als ausreichend angesehen, wenn dem Gateway für einen Messauftrag ein Messwert für jeden virtuellen Sensor vorliegt. Hat also ein mobiler Knoten für einen virtuellen Sensor einen Messwert aufgenommen, so muss kein anderer mobiler Knoten mehr einen Messwert für denselben virtuellen Sensor aufnehmen. Um dies zu erreichen, sendet ein mobiler Knoten per WLAN eine Broadcast Nachricht an alle anderen mobilen Knoten, die im Empfangsbereich des WLAN sind, sobald er einen Messwert für einen virtuellen Sensor aufgenommen hat. Dieser Broadcast enthält eine Liste mit allen virtuellen Sensoren, für die der mobile Knoten bereits Messwerte aufgenommen hat.

Ein mobiler Knoten, der einen solchen Broadcast empfangen hat, entfernt alle virtuellen Sensoren, die er in der Nachricht erhalten hat aus seinem Messauftrag. Bei der nächsten Positionsbestimmung befinden sich also weniger virtuelle Sensoren im Messauftrag. Es ist in diesem Fall zu erwarten, dass die Zeit bis zur nächsten Positionsbestimmung damit verlängert werden kann und somit die Gesamtzahl der Positionsbestimmungen über alle mobilen Knoten hinweg gesenkt wird.

### 4.7 Modelloptimierung durch Mobilitätsvorhersage

Im Abschnitt Modellbasierter Ansatz wird vorgestellt, wie die Zahl der angefragten virtuellen Sensoren reduziert werden kann. Dieser Ansatz soll weiter optimiert werden, indem das Gateway Vorhersagen trifft, wo sich mobile Knoten aufhalten können und dies in die Auswahl der virtuellen Sensoren mit einbezieht.

Voraussetzung hierfür ist, dass das Gateway Informationen über den Aufenthaltsort der mobilen Knoten hat. Um dies zu erreichen, versendet jeder mobile Knoten, wann immer er Messergebnisse ans Gateway schickt, seine aktuelle Position mit. Sendet ein mobiler Knoten für zwei aufeinanderfolgende Messaufträge keine Daten an das Gateway zurück, so wird das Gateway ohne Messdaten über die aktuelle Position des mobilen Knotens informiert.

In der Lernphase werden die Positionsdaten nur genutzt, um die Messaufträge für die einzelnen mobilen Knoten zu individualisieren. Hierfür prüft das Gateway für jeden mobilen Knoten, welche virtuellen Sensoren innerhalb des Messzeitraums erreicht werden können. Dazu wird mit den gespeicherten Positionsdaten des mobilen Knotens und seiner Geschwindigkeit der Radius berechnet, in dem sich der mobile Knoten zum Ende des Messzeitraums befinden kann. Virtuelle Sensoren außerhalb dieses Bereiches werden aus dem individualisierten Messauftrag entfernt. Der so individualisierte Messauftrag wird dann an den mobilen Knoten versendet. Mobile Knoten, die keine virtuellen Sensoren erreichen können, erhalten keine Nachricht.

In der Optimierungsphase werden die Positionsdaten, wie in der Lernphase für die Individualisierung der Messaufträge genutzt. Zusätzlich werden sie allerdings auch bei der Auswahl von angefragten virtuellen Sensoren berücksichtigt.

In [DP13] wird ein Mehrundenansatz, mit einer alternativen Auswahl von virtuellen Sensoren beschrieben. Hierbei werden in der ersten Runde alle virtuellen Sensoren als verfügbar angenommen. Basierend auf dem berechneten Modell werden dann einzelne Sensoren ausgewählt und angefragt. Virtuelle Sensoren, für die keine Ergebnisse zurückgeliefert werden, werden als nicht verfügbar betrachtet und in den kommenden Anfragerunden nicht mehr angefragt. Stattdessen wird versucht, Messwerte einzuholen, aus denen Werte für nicht verfügbare virtuelle Sensoren berechnet werden können. Dies geschieht unter Zuhilfenahme des Modells und verfügbarer virtueller Sensoren.

Diese Arbeit verwendet zur Auswahl von Sensoren dieselben Algorithmen, die auch in [DP13] verwendet werden. Allerdings gibt es nur eine Anfragerunde. Die Verfügbarkeit der virtuellen Sensoren wird auf Basis der Positionsdaten berechnet. Hierzu wird für jeden virtuellen Sensor geprüft, ob er, auf Basis des euklidischen Abstands, innerhalb des

Messzeitraums von einem mobilen Knoten erreicht werden kann. Sensoren, die nicht erreicht werden können, werden für den generierten Messauftrag dann als nicht erreichbar markiert und aus der Auswahl von angefragten Sensoren ausgeschlossen.

## 5 Implementierung

Im vorherigen Kapitel wurden die Lösungsansätze, für die an die Arbeit gestellten Anforderungen vorgestellt. Dieses Kapitel beschreibt die konkrete Implementierung der Lösungen. Den Einstieg in dieses Kapitel bildet eine Übersicht über die Simulationsumgebung. Im Anschluss folgt ein Überblick darüber, wie eine Simulation abläuft. Der Überblick über das Gesamtsystem sowie die Änderungen, die für den oben vorgestellten Entwurf nötig waren, runden das Kapitel ab.

### 5.1 Simulationsumgebung

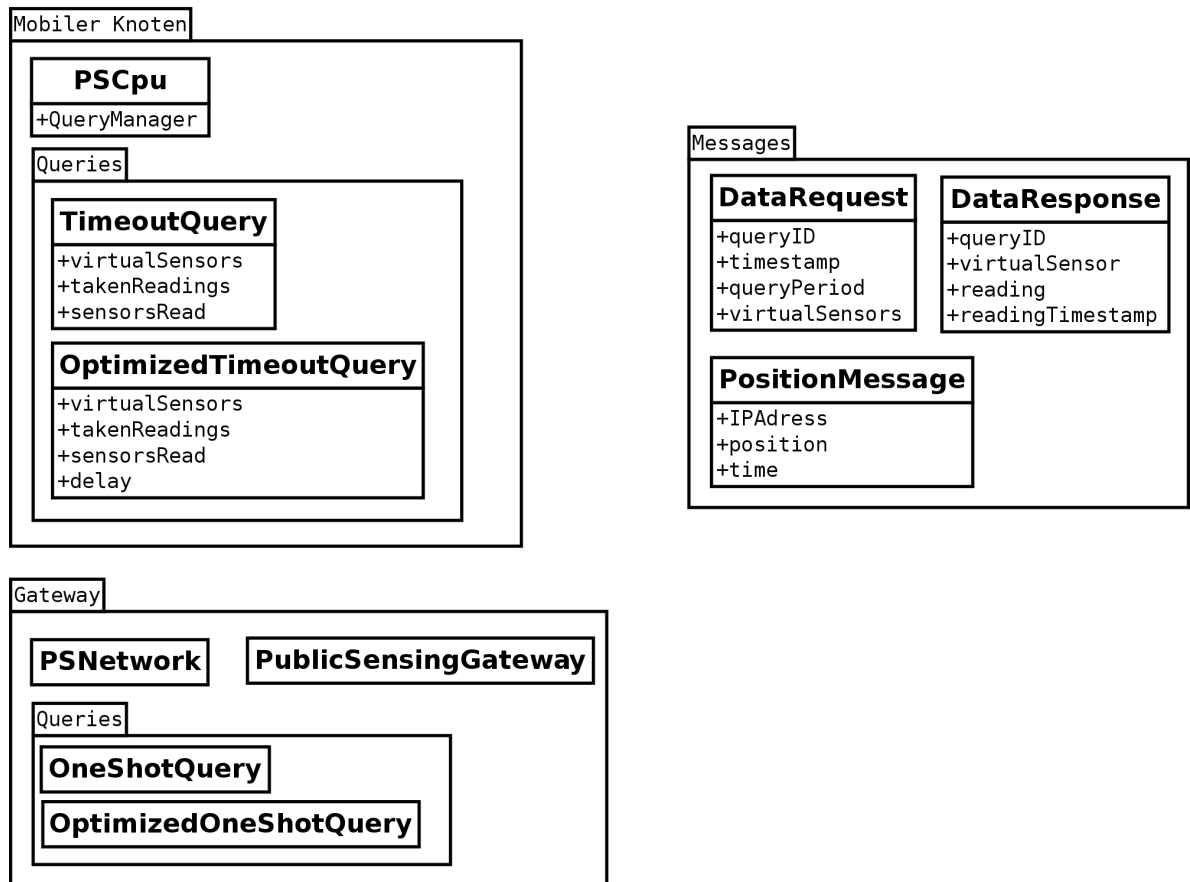
Die Implementierung wurde in einem bestehenden Public-Sensing-Simulator vorgenommen. Die Simulation ist in C++ geschrieben und nutzt den Netzwerksimulator OMNet++. Um die genutzten Netzwerkprotokolle zu emulieren, verwendet die Simulation inetmanet, eine für OMNet++ entwickelte Bibliothek.

### 5.2 Ablauf einer Simulation

Die Simulationen basieren auf einem festen Datenbestand. Das heißt die Situation, die an den virtuellen Sensoren zu einem bestimmten Zeitpunkt vorliegt, ist gespeichert. Bei jeder Simulation wird eine vorher festgelegte Anzahl an mobilen Knoten simuliert. Jeder mobile Knoten befindet sich an einer Position im Messgebiet. Die Position kann sowohl exakt, als auch mit Ungenauigkeit ausgelesen werden. Dadurch wird ein möglicher Lokalisierungsfehler von GPS emuliert. Die mobilen Knoten bewegen sich mit einer festen Geschwindigkeit zwischen den virtuellen Sensoren. Sobald ein mobiler Knoten einen virtuellen Sensor erreicht, wählt er einen neuen Sensor aus und bewegt sich direkt auf diesen zu.

Der Messwert an einem virtuellen Sensor kann von einem mobilen Knoten erfasst werden, wenn er sich seiner GPS-Positionsbestimmung nach, im Messbereich des virtuellen Sensors aufhält.

Für die Kommunikation zwischen den Komponenten des Simulationssystems, also zwischen dem Gateway und den mobilen Knoten, beziehungsweise zwischen den mobilen Knoten werden Nachrichten verwendet. Der Versand von Nachrichten zwischen einzelnen Komponenten ist eine Standardfunktionalität von OMNet++. Beim Nachrichtenversand werden Kommunikationsprotokolle emuliert, um realistischen Bedingungen möglichst nahe zu kommen. Genutzte Kommunikationsprotokolle sind TCP/IP und UDP. Für die Emulation dieser



**Abbildung 5.1:** Wichtige Komponenten des Public-Sensig-Systems

Protokolle, sowie der daran beteiligten Komponenten (zum Beispiel WLAN Chip oder UMTS Modul) wird inetmanet genutzt.

### 5.3 Aufbau des Public-Sensing-Systems

Dieses Kapitel beschreibt die konkrete Implementierung der wichtigsten Komponenten des Simulations-Systems. Das an UML angelehnte Diagramm 5.1 soll einen Überblick über wichtige Klassen geben, insbesondere solche, die für die Realisierung des Entwurfs erstellt oder verändert wurden. Weitere Klassen wurden für weitergehende Erklärungen aufgenommen. Die Strukturierung in Pakete im Diagramm ist nach Zugehörigkeit zu Komponenten oder logischen Bereichen geschehen und stimmt nicht mit der Paketaufteilung im Code überein.

### 5.3.1 Gateway

#### **PublicSensingGateway**

Die `PublicSensingGateway` Klasse implementiert die Logik des Gateways. Dies ist insbesondere das Initiieren neuer Messaufträge. Außerdem wertet sie Messwerte aus und hält Informationen darüber, ob ein Modell vorliegt oder nicht.

#### **OneShotQuery**

Die Klasse `OneShotQuery` erstellt den Messauftrag, für den Fall, dass kein Modell vorliegt, sich das Gateway also in der Lernphase befindet.

#### **OptimizedOneShotQuery**

Diese Klasse erstellt den Messauftrag, wenn ein Modell vorliegt, das Gateway sich also in der Optimierungsphase befindet. In dieser Klasse findet somit die Einteilung in verfügbare und nicht verfügbare virtuelle Sensoren statt.

#### **PSNetwork**

Die Klasse `PSNetwork` realisiert die Kommunikation mit den mobilen Knoten. Hier sind die IP Adressen aller mobiler Knoten bekannt. Messaufträge werden von dieser Klasse an die einzelnen Knoten adressiert und an sie versendet.

Außerdem empfängt die `PSNetwork` Klasse auch alle Nachrichten von den mobilen Knoten, liest die Messwerte aus und gibt diese an das `PublicSensingGateway` weiter.

### 5.3.2 Messages

Die hier aufgeführten Nachrichten sind die Nachrichten, die zwischen dem Gateway und den mobilen Knoten beziehungsweise umgekehrt ausgetauscht werden.

#### **DataRequest**

Ein `DataRequest` repräsentiert einen Messauftrag. Jeder Messauftrag hat eine eindeutige ID (`queryID`), durch die Messergebnisse eindeutig einem Messauftrag zugeordnet werden können.

Das Attribut `timestamp` ist der Startzeitpunkt des Messauftrags, die Länge des Messzeitraums wird durch das Attribut `queryPeriod` angegeben. Der späteste Zeitpunkt zu dem das Gateway Messergebnisse für einen Messauftrag entgegen nimmt, entspricht also dem Zeitpunkt `timestamp + queryPeriod`.

### **DataResponse**

Eine DataResponse repräsentiert ein Messergebnis. Der Versand von DataResponses erfolgt gekapselt als GatewayPacket. Dadurch ist der gleichzeitige Versand mehrerer Messergebnisse möglich. Eine DataResponse Nachricht besteht aus der queryID, um das Messergebnis eindeutig einer Anfrage zuordnen zu können, sowie aus den Informationen über den virtuellen Sensor zu dem der Messwert gehört (virtualSensor), dem Messwert (reading) und dem Zeitpunkt der Messung (readingTimestamp).

### **5.3.3 Mobiler Knoten**

Die Hauptklasse der mobilen Knoten ist die PSCpu Klasse. Sie ist für das Verarbeiten der Messaufträge zuständig, holt Messungen ein und sendet sie wieder ans Gateway zurück.

## **5.4 Implementierte Optimierungen**

### **5.4.1 Ausnutzung des kompletten Messzeitraums**

#### **Parameter: PSCpu.useTimeoutForReadings**

true: aktiviert die Ausnutzung des kompletten Messzeitraums

Erhält die PSCpu einen Messauftrag, so wird ein neuer TimeoutQuery erstellt und zum QueryManager hinzugefügt. Die Klasse TimeoutQuery speichert die angefragten virtuellen Sensoren und initiiert durch einen Timer, dass bei jedem Simulationsschritt, also jede Sekunde, die Erreichbarkeit aller virtuellen Sensoren geprüft wird. Befindet sich die PSCpu im Messbereich eines virtuellen Sensors und für den virtuellen Sensor wurde noch kein Messwert aufgenommen (sensorsRead[virtueller Sensor] = false), so wird ein Messwert aufgenommen und an das Gateway verschickt. sensorsRead wird für den entsprechenden virtuellen Sensor auf true gesetzt, damit keine weiteren Messwerte für diesen virtuellen Sensor aufgenommen werden.

### **5.4.2 Reduzierung von Positionsbestimmungen**

#### **Parameter: PSCpu.localOptimization**

true: Aktiviert die lokale Reduzierung von Positionsbestimmungen

Erhält die PSCpu einen neuen Messauftrag, so wird zunächst berechnet, welche im Messauftrag enthaltenen virtuellen Sensoren im Messzeitraum erreicht werden können. Alle anderen werden aus dem Messauftrag entfernt. Außerdem wird die Zeit berechnet, zu der der nächste virtuelle Sensor erreicht werden kann. Verbleiben virtuelle Sensoren im Messauftrag, wird ein neuer OptimizedTimeoutQuery erstellt und im QueryManager gespeichert. Der OptimizedTimeoutQuery hält die verbleibenden Messpunkte, sowie den Zeitpunkt, zu

dem der nächste virtuelle Sensor erreicht werden kann. Zu diesem Zeitpunkt wird mittels eines Timers ein erneutes Überprüfen aller verbliebenen virtuellen Sensoren initiiert. Nicht erreichbare virtuelle Sensoren werden aus dem Messauftrag entfernt. Gegebenenfalls werden Messwerte aufgenommen und an das Gateway gesendet. Im Anschluss daran wird wieder ein Timer gestartet, der zum nächsten Zeitpunkt, zu dem ein virtueller Sensor erreicht werden kann, die durchzuführenden Aktionen initiiert.

### 5.4.3 Optimierung der Kommunikation mit dem Gateway

**Parameter: PSCpu.useOptimizedTimeouts**

true: Messwerte werden lokal gespeichert und gesammelt ans Gateway übertragen

**Parameter: PSCpu.immediateSendOffset**

Differenz zum Ende des Messzeitraums in Sekunden. Messzeitraumende - immediateSendOffset ist der Zeitpunkt, zu dem die gesammelten Messwerte ans Gateway verschickt werden.

Diese Optimierung funktioniert sowohl mit dem TimeoutQuery als auch mit dem OptimizedTimeoutQuery. Diese speichern ihre genommenen Messwerte jeweils in takenReadings. Bei jeder erneuten Prüfung, ob Messwerte aufgenommen werden können, wird dann zusätzlich geprüft, ob die Messwerte ans Gateway versendet werden müssen.

### 5.4.4 WLAN Erweiterung für die Reduzierung von Positionsbestimmungen

**Parameter: PSCpu.wlanReadingDistribution**

true: Aktiviert die Optimierung

Ist diese Option aktiv, so speichert die Klasse OptimizedTimeoutQuery eine Liste mit allen virtuellen Sensoren, für die sie Messwerte aufgenommen hat (sensorsRead). Außerdem wird, nachdem ein neuer Sensor in die Liste aufgenommen wurde, eine Nachricht per WLAN Broadcast versendet, die die komplette sensorsRead Liste enthält. Als Nachricht wird die ReadingUpdateMessage verwendet.

Empfängt eine PSCpu Instanz eine ReadingUpdateMessage, so gibt sie alle darin enthaltenen virtuellen Sensoren an den OptimizedTimeoutQuery weiter. Die virtuellen Sensoren werden dann aus dem Messauftrag entfernt.

### 5.4.5 Modelloptimierung durch Mobilitätsvorhersage

**Parameter: PSNetwork.individualQueries**

true: virtuelle Sensoren, die von einem mobilen Knoten nicht erreicht werden können, werden für diesen mobilen Knoten aus dem Messauftrag entfernt.

**Parameter: Gateway.checkNodeAvailability**

true: das Gateway wählt nur virtuelle Sensoren aus, die als verfügbar markiert sind

Grundvoraussetzung ist zunächst, dass die PSCpu Instanzen ihre simulierte Position ans Gateway übermitteln. Hierzu versenden sie, wann immer sie Messwerte ans Gateway schicken eine PositionUpdateMessage mit. Diese Nachrichten werden ebenfalls vom PSNetwork empfangen und die darin enthaltenen Informationen ans Gateway weitergeleitet.

Die Überprüfung, ob ein virtueller Sensor von einem mobilen Knoten erreicht werden kann, wird mit Hilfe der Klasse PositionStatus berechnet. Für jeden mobilen Knoten wird eine Instanz dieser Klasse gehalten. Die Instanz für einen mobilen Knoten berechnet in der Funktion queryNode, ob eine gegebene Position bis zu einem gegebenen Zeitpunkt erreicht werden kann.

Die beschriebene Überprüfung kann von der Klasse OptimizedOneShotQuery durchgeführt werden, wenn diese die Verfügbarkeit von virtuellen Sensoren überprüft und von der Klasse PSNetwork, wenn diese die Queries an die mobilen Knoten verschickt.

## 6 Evaluation

In diesem Kapitel werden die durchgeführten Simulationen vorgestellt und ausgewertet. Insbesondere wird auf die in den Anforderungen vorgestellten Ziele eingegangen und es wird überprüft, inwiefern diese erreicht wurden.

### 6.1 Methodologie

Die Simulationen basieren auf den ersten 24 Stunden der Intel-Lab-Data Messdaten[bhg]. Diese Messdaten bestehen aus verschiedenen Werten für 54 angebrachte Sensoren. Die Positionen dieser Sensoren entsprechen den Positionen von virtuellen Sensoren, die in den Simulationen angefragt wurden. Die von den Knoten erfassten Messwerte entsprechen also denen, die in den Intel-Lab-Data Messdaten gespeichert sind.

Die Berechnungen der benötigten Energie wird für WLAN auf Basis des in [XSK10] vorgestellten Modells durchgeführt. Für die UMTS Kommunikation wird [BBV09] verwendet. Für eine Positionsbestimmung mittels GPS werden 75mJ Energie verbraucht. [BDR12]

Im folgenden werden die verschiedenen Optimierungen miteinander verglichen. Dabei werden zunächst die Optimierungen für die Verbesserung der erhaltenen Messwerte untereinander verglichen. Im Anschluss daran werden die Entwürfe für das Reduzieren des Energieverbrauchs ausgewertet. Da [PSA<sup>+</sup>13] der Ausgangspunkt dieser Arbeit war, wird jeweils der modellbasierte Ansatz als Standardfall verwendet. Dass der modellbasierte Ansatz gegenüber dem naiven Ansatz besser ist, wird in [PSA<sup>+</sup>13] gezeigt und wird in dieser Arbeit so weiterverwendet.

### 6.2 Metriken

In diesem Abschnitt werden die, für die Diagramme verwendeten Metriken, vorgestellt.

#### Anfragen ohne Ergebnis

Eine Anfrage an das Public-Sensing-System definiert eine Liste von virtuellen Sensoren, für die bestimmte Messwerte eingeholt werden sollen. Weiterhin definiert eine Anfrage Qualitätsanforderungen, die die vom Gateway zurückgelieferten Ergebnisse erfüllen müssen. Das Gateway generiert für jede Anfrage Messaufträge, die an die mobilen Knoten verschickt werden.

Die Metrik Anfragen ohne Ergebnis gibt an, für wie viel Prozent der Anfragen kein Messauftrag ein Ergebnis zurückgeliefert hat; es hat also kein mobiler Knoten einen Messwert für die Anfrage aufgenommen und ans Gateway zurückgeschickt. Das Gateway hatte keine Anhaltspunkte, um Ergebnisse für die Anfrage zu erzeugen.

### **Erfolgreiche Anfragen**

Die Metrik der erfolgreichen Anfragen definiert wie viel Prozent der Anfragen an das Public-Sensing-System Ergebnisse geliefert haben, die den gestellten Qualitätsanforderungen gerecht wurden. Anfragen, die kein Ergebnis zurückgeliefert haben, haben auf diese Metrik keinen Einfluss.

### **Energieverbrauch**

Für den Energieverbrauch wird jeweils der durchschnittliche Energieverbrauch pro mobilem Knoten berechnet. Je nach zu untersuchender Optimierung, werden die Durchschnitte relativ zueinander betrachtet.

## **6.3 Ergebnisse**

### **6.3.1 Ausnutzung des kompletten Messzeitraums**

In dieser Simulation wird der modellbasierte Ansatz (Modell) um die Ausnutzung des kompletten Messzeitraums (kompletter Messzeitraum) erweitert.

Simulationseinstellungen:

- Geschwindigkeit der mobilen Knoten: 1-2m/s
- Anzahl mobiler Knoten: 20, 40, 60, 80, 100, 140, 180, 220

Sowohl im Diagramm 6.1 Anfragen ohne Ergebnis, als auch im Diagramm 6.2 Erfolgreiche Anfragen ist eine deutliche Verbesserung des Ergebnisses zu erkennen.

Die Anfragen ohne Ergebnis sinken bereits bei 40 mobilen Knoten unter 1%. Rund 1% wird dagegen beim Ansatz ohne Ausnutzung des kompletten Messzeitraums erst bei 220 mobilen Knoten erreicht.

Die erfolgreichen Anfragen machen beim rein modellbasierten Ansatz bei 180 mobilen Knoten 68% aus. Dieser Wert wird bei Ausnutzung des kompletten Messzeitraumes bereits mit 20 mobilen Knoten erreicht. Schon mit 40 mobilen Knoten ist das Ergebnis deutlich besser als das Ergebnis für 220 mobile Knoten ohne Ausnutzung des kompletten Messzeitraums.

### **Diskussion**

Bei den Intel-Lab-Data Messdaten liegen die virtuellen Sensoren sehr nahe beieinander. Dadurch reichen bereits kleine Bewegungen der mobilen Knoten aus, um zum nächsten Messpunkt zu gelangen. In den Messzeiträumen können daher Messwerte an sehr vielen virtuellen Sensoren aufgenommen werden.

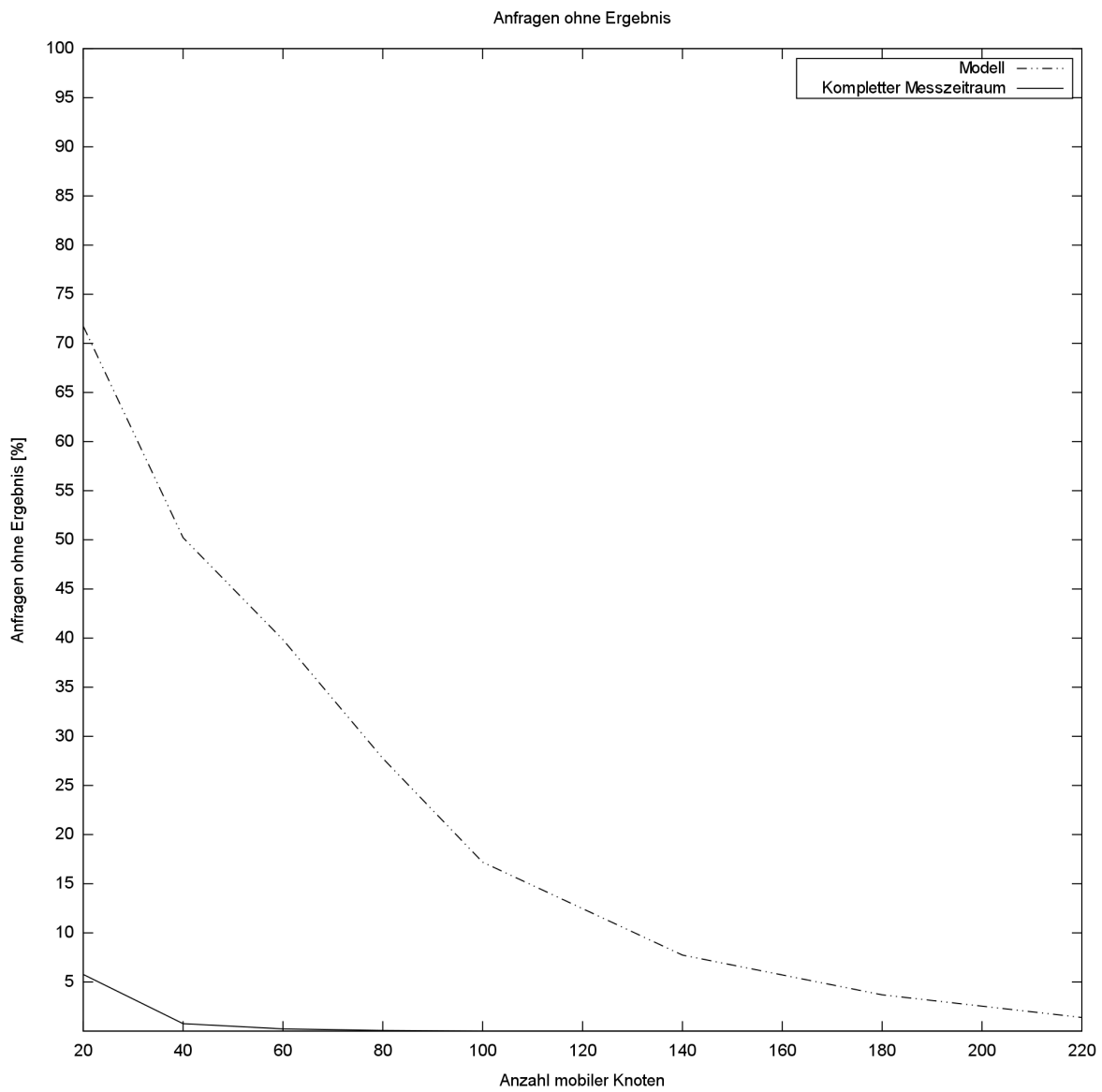


Abbildung 6.1: Anfragen ohne Ergebnis

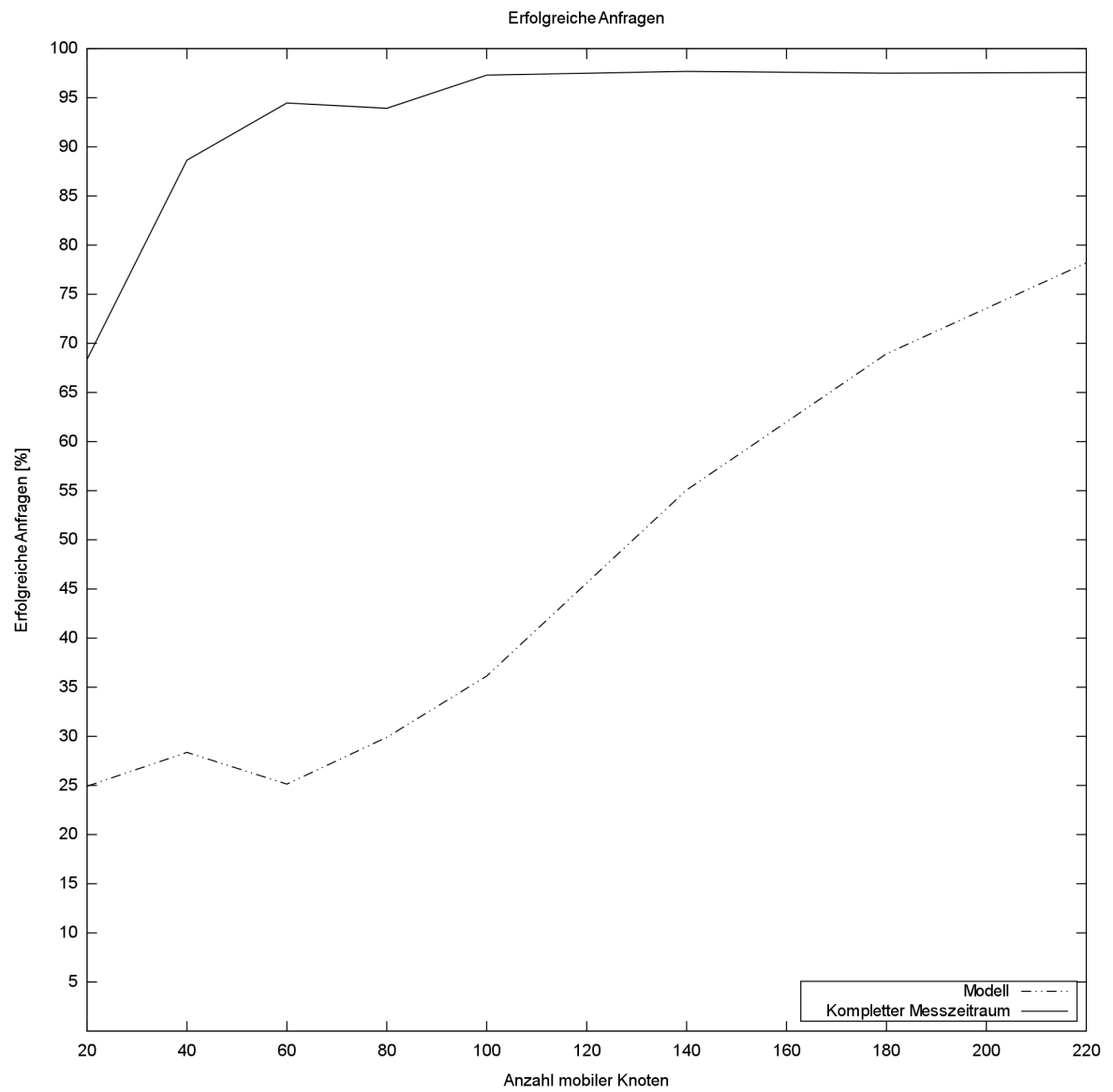


Abbildung 6.2: Erfolgreiche Anfragen

### 6.3.2 Modelloptimierung durch Mobilitätsvorhersage

In dieser Simulation werden die beiden oben aufgeführten Ansätze simuliert. Zusätzlich wurde die Modelloptimierung durch Mobilitätsvorhersage aktiviert.

Simulationseinstellungen:

- Geschwindigkeit der mobilen Knoten 0.5m/s
- Anzahl mobiler Knoten: 20, 40, 60, 80, 100, 140, 180, 220

Unter beiden Metriken ist wieder gut zu erkennen, dass das Ausnutzen des kompletten Messzeitraums zu großen Verbesserungen führt. Auch bei einer langsameren Fortbewegungsgeschwindigkeit der mobilen Knoten können starke Verbesserungen erreicht werden.

Durch die Auswirkung der Mobilitätsvorhersage ist eine Verbesserung im sehr kleinen Bereich erkennbar. Die Anfragen ohne Ergebnis sinken um ca. 2 Prozentpunkte. In etwa gleichen Maße steigen die erfolgreichen Anfragen. (Diagramme 6.3, 6.4)

### Diskussion

Auffällig ist, dass nur bei wenigen mobilen Knoten eine Verbesserung erreicht werden kann. Das Problem ist, dass jeder mobile Knoten sich von seinem Standort aus beliebig in jede Richtung bewegen kann. Dadurch wächst das Gebiet, in dem ein mobiler Knoten möglicherweise einen virtuellen Sensor erreichen kann, quadratisch zur Zeit, die der Knoten hat, um sich zu bewegen.

Darstellung: Ein mobiler Knoten bewegt sich pro Zeiteinheit (t) um eine Einheit (s) in eine Richtung. Das Gebiet (G), in dem sich der mobile Knoten aufhalten kann ist dann definiert durch  $G(t) = \pi * (t * s)^2$

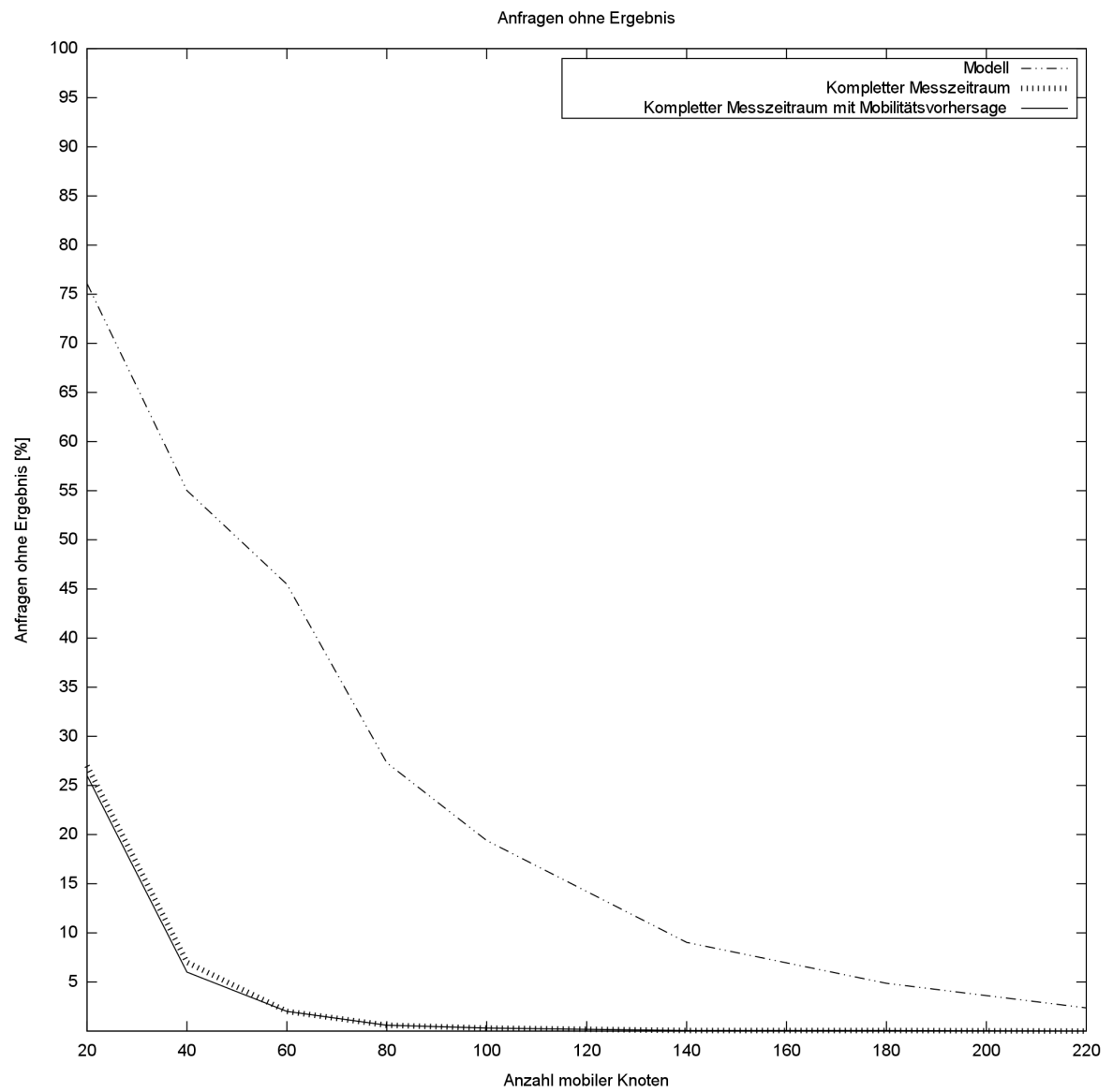
Jeder mobile Knoten, der im Messgebiet unterwegs ist, vergrößert also das Gebiet, in dem theoretisch virtuelle Sensoren erreicht werden können. Diese Gebiete sind jedoch nicht praktisch erreichbar.

Derselbe Grund ist auch dafür verantwortlich, dass die Qualitätszugewinne so gering ausfallen.

### 6.3.3 Optimierung der Kommunikation mit dem Gateway

Diese Simulation betrachtet den Energieaufwand je nach Zeitpunkt, zu dem Messwerte ans Gateway versendet werden. Da ein mobiler Knoten nur dann Messwerte für mehrere virtuelle Sensoren aufnehmen kann, wenn die Ausnutzung des kompletten Messzeitraums aktiv ist, ist diese Option ebenfalls aktiv.

Als maximaler Energieaufwand (100%) wird der Fall betrachtet, in dem ein mobiler Knoten



**Abbildung 6.3:** Anfragen ohne Ergebnis

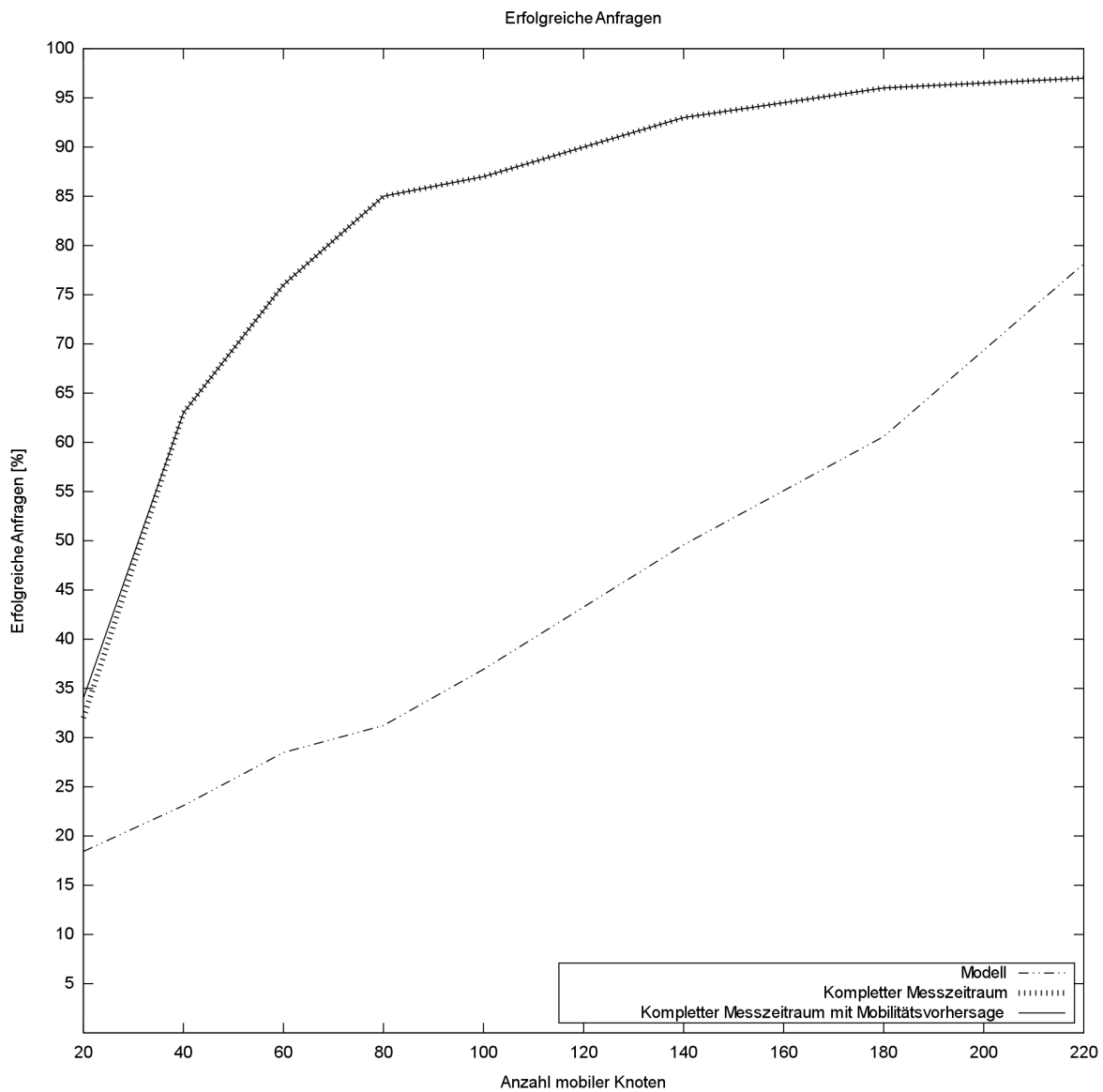
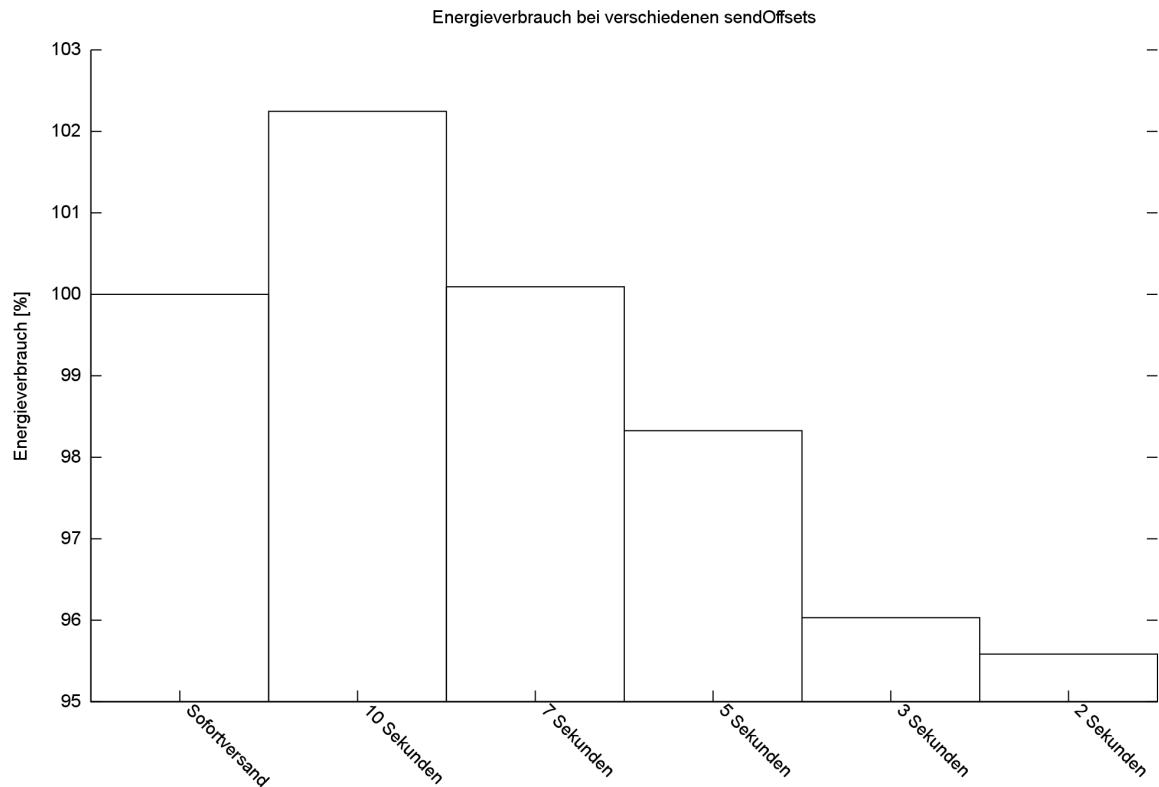


Abbildung 6.4: Erfolgreiche Anfragen



**Abbildung 6.5:** Erfolgreiche Anfragen

einen aufgenommenen Messwert sofort ans Gateway übermittelt. Weiterhin werden verschiedene Abstände in Sekunden zum Zeitpunkt hin betrachtet, zu dem der Messauftrag endet.

Simulationseinstellungen:

- Direktversand von Messwerten
- Sekunden vor Ende des Messzeitraums: 10, 7, 5, 3, 2

Das Diagramm 6.5 zeigt, dass durch einen kleinen Abstand zum Ende des Messzeitraums Energie gespart werden kann. Allerdings fällt auch auf, dass sich ein zu großer Abstand negativ auf die Energiebilanz auswirken kann.

### Diskussion

Logisch betrachtet gilt – werden Messwerte gesammelt und in einer Nachricht versendet, so werden in der Summe weniger Nachrichten versendet und somit Energie gespart. Dies bestätigt sich auch durch die kleinen Timeouts.

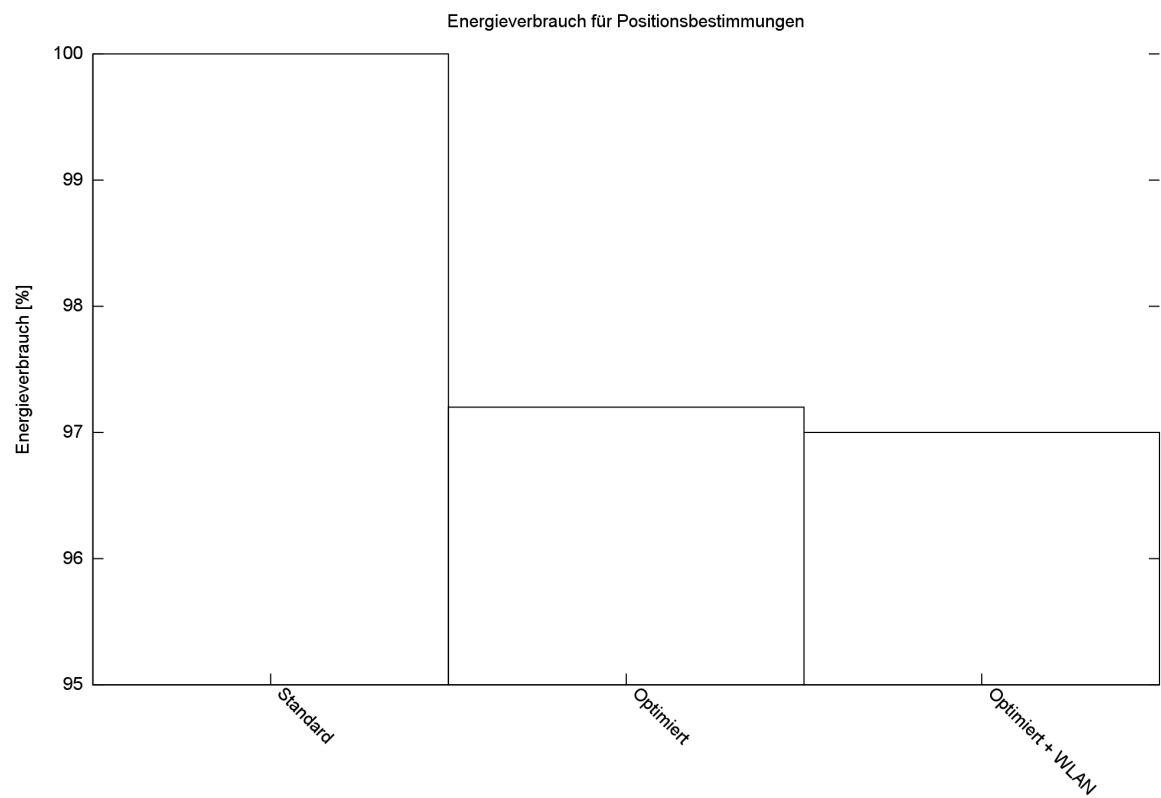
Es gibt jedoch auch 2 Werte, die dieser logischen Betrachtung widersprechen und daher näher betrachtet werden müssen. Die Simulation für 10 Sekunden benötigt deutlich mehr Energie und auch die Simulation mit 7 Sekunden weicht nach oben hin ab. Um diese Werte erklären zu können, ist es notwendig, die Funktionsweise des UMTS Chips näher zu betrachten.

Der UMTS Chip der mobilen Knoten unterscheidet zwei Stand-By Zustände; einen mit niedrigem Energieverbrauch (im folgenden „Low-Power-State“) und einen mit hohem Energieverbrauch (im folgenden „High-Power-State“). Zum Senden und Empfangen von Nachrichten muss in den High-Power-State gewechselt werden. Hierfür wird Energie benötigt. Außerdem bleibt der High-Power-State eine gewisse Zeit aktiv. Kann ein mobiler Knoten, aufgrund eines zu großen Abstands zum Messzeitraumende, nicht lang genug in den Low-Power-State schalten, kann es sein, dass die Energie, die benötigt wird, um in den High-Power-State zurückzuschalten, größer ist, als die Energie, die benötigt worden wäre, wenn der mobile Knoten dauerhaft im High-Power-State verblieben wäre und eine kleine Zahl an Nachrichten verschickt hätte.

#### 6.3.4 Reduzierung von Positionsbestimmungen

Im folgenden Diagramm 6.6 wird nur die Energie berücksichtigt, die für Positionsbestimmungen aufgewendet wird. Als Ausgangspunkt, also als 100% wird der Fall angenommen, in dem die mobilen Knoten regelmäßig ihre Position prüfen (Standard). Verglichen wird der Wert mit dem Fall, in dem die mobilen Knoten nur eine Positionsbestimmung durchführen, wenn sie rechnerisch einen virtuellen Sensor erreicht haben können (Optimiert). Als letztes wird der Wert für die Reduzierung von Positionsbestimmungen mit WLAN Erweiterung (Optimiert + WLAN) betrachtet.

Alle Werte sind erwartungsgemäß, also der Standardfall benötigt am meisten Energie, gefolgt vom Fall ohne WLAN Optimierung. Am wenigsten Energie wird für den Fall mit WLAN verwendet. Allerdings ist der letzte Wert nur ein theoretischer Wert, da nur die Energie, die für die Positionierungen aufgewendet wird, berücksichtigt ist. Hier muss normalerweise noch die WLAN Stand-By Energie mit eingerechnet werden. Rechnet man diese Energie mit ein, so lohnt sich die WLAN Optimierung nicht mehr. Allerdings könnte man die WLAN Energie amortisieren, indem man WLAN an weiteren Stellen zur Optimierung nutzt.



**Abbildung 6.6:** Erfolgreiche Anfragen

# 7 Zusammenfassung und Ausblick

## 7.1 Zusammenfassung

In dieser Arbeit wurden Methoden vorgestellt, durch die die Qualität von Messergebnissen in Public-Sensing-Systemen erhöht werden konnte. Die vorgestellten Verfahren waren sowohl die Ausnutzung des kompletten Messzeitraumes, als auch die Nutzung von Mobilitätsvorhersagen über die mobilen Knoten. Auch der Einfluss auf die verbrauchte Energie wurde dabei stets mit beachtet und konnte gegenüber den ersten Ansätzen verbessert werden.

Zusammenfassend kann gesagt werden, dass insbesondere die Ausnutzung des kompletten Messzeitraums zu deutlichen Verbesserungen führt. Dies führt zwar zunächst zu einer bedeutend schlechteren Energiebilanz, jedoch hat diese Arbeit auch Wege vorgestellt, um diesen negativen Auswirkungen entgegenzuwirken.

Es hat sich gezeigt, dass es für die Reduzierung der benötigten Energie insbesondere hilfreich ist, die Kommunikation über UMTS auf ein Minimum zu beschränken.

Der Versuch, die Zahl der Positionierungen mit Hilfe von WLAN zu reduzieren, hat sich als nicht sehr hilfreich herausgestellt, um die Energieeffizienz zu erhöhen. Die, durch das ständig aktivierte WLAN, verbrauchte Energie überschreitet die Einsparungen durch weniger Positionsbestimmungen bei Weitem.

Die Mobilitätsvorhersage weist in der implementierten Form das Problem auf, dass sich der Knoten in jede Richtung bewegen kann. Dadurch ist der Bereich, in dem sich ein mobiler Knoten zu einem späteren Zeitpunkt aufhalten kann, sehr groß. Als Folge werden zu viele virtuelle Sensoren als verfügbar markiert, die in Realität nicht verfügbar sind.

## 7.2 Ausblick

Ausgehend von dieser Arbeit, bieten sich weitere Untersuchungen an.

In [DP13] wird beispielsweise ein Mehrundenansatz vorgestellt, durch den die Messzeiträume verkürzt werden, dafür aber öfter die Verfügbarkeit von virtuellen Sensoren aktualisiert wird. Hierbei könnte untersucht werden, wie sich die Verkürzung des Messzeitraums, die in dieser Arbeit mit den größten Nutzen erzielt hat, auswirkt, wenn dafür die Mobilitätsvorhersagen exakter werden, da der Zeitraum, in dem sich die mobilen Knoten weiterbewegen können, verkürzt und sich damit die Unsicherheit wohin sich die mobilen Knoten bewegen verringert.

Das System könnte ebenfalls um realistische Bewegungspfade erweitert werden. Dann

könnten mobile Knoten sich nicht mehr in jede Richtung bewegen, sondern müssten bestimmten Routen folgen, was die Mobilitätsvorhersagen, zwar komplexer, aber zuverlässiger machen würde.

Eine weitere Möglichkeit ist es, die Verwendung von WLAN auszubauen, da WLAN im Dauerbetrieb weniger Energie benötigt als UMTS. Hierfür könnten Messaufträge nur an ausgesuchte mobile Knoten per UMTS versendet werden, die ihrerseits die Messaufträge per WLAN weiterverbreiten. Umgekehrt kann auf genau dasselbe Verfahren gesetzt werden, wenn es darum geht, Messwerte zum Gateway zurückzusenden. Statt dass jeder mobile Knoten über UMTS kommuniziert, kommunizieren nur einzelne mobile Knoten über UMTS. Diese erhalten die Messwerte anderer mobiler Knoten in ihrer Nähe per WLAN und leiten diese ebenfalls ans Gateway weiter. Durch ein solches Verfahren könnte die WLAN Stand-By Energie, die ein großes Problem dargestellt hat, amortisiert werden.

# Literaturverzeichnis

- [BBV09] N. Balasubramanian, A. Balasubramanian, A. Venkataramani. Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications. *IMC'09*, November 4-6, 2009. (Zitiert auf Seite 29)
- [BDR12] P. Baier, F. Dürr, K. Rothermel. PSense: Reducing Energy Consumption in Public Sensing Systems. *Proceedings of the 26th IEEE International Conference on Advanced Information Networking and Applications (AINA-2012)*, S. 136–143, 2012. (Zitiert auf den Seiten 10 und 29)
- [bhg] URL <http://db.csail.mit.edu/labdata/labdata.html>wie. (Zitiert auf Seite 29)
- [CLM<sup>+</sup>08] A. T. Campbell, N. D. Lane, E. Miluzzo, R. A. Person, H. Lu, X. Zheng, M. Musolesi, K. Fodor, S. B. Eisenman, G.-S. Ahn. The Rise of People-Centric Sensing. *IEEE Internet Computing*, 12:12–21, 2008. (Zitiert auf Seite 9)
- [DGM<sup>+</sup>04] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, W. Hong. Model-Driven Data Acquisition in Sensor Networks. *Proceedings of the Thirtieth international conference on very large databases*, 30:588, 2004. (Zitiert auf Seite 10)
- [DP13] F. D. K. R. Damian Philipp, Jaroslaw Stachowiak. Model-Driven Public Sensing in Sparse Networks. 2013. (Zitiert auf den Seiten 21 und 39)
- [KBLR09] E. Kanjo, J. Bacon, P. Landshoff, D. Roberts. MobSens: Making Smart Phones Smarter. *IEEE Pervasive Computing*, October - December:50–57, 2009. (Zitiert auf Seite 10)
- [KBP<sup>+</sup>08] E. Kanjo, S. Benford, M. Paxton, A. Chamberlain, D. S. Fraser, D. Woodgate, D. Crellin, A. Woolard. MobGeoSen: facilitating personal geosensor data collection and visualization using mobile phones. *Pers Ubiquit Comput*, 12:599–607, 2008. (Zitiert auf Seite 9)
- [LBD<sup>+</sup>05] B. Liu, P. Brass, O. Dousse, P. Nain, D. Towsley. Mobility Improves Coverage of Sensor Networks. *MobiHoc'05*, May 25–27, 2005. (Zitiert auf Seite 10)
- [LEM<sup>+</sup>08] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, A. T. Campbell. Urban Sensing Systems: Opportunistic or Participatory? *Proceedings of the 9th workshop on Mobile computing systems and applications*, S. 11–16, 2008. (Zitiert auf Seite 9)
- [PDR11] D. Philipp, F. Dürr, K. Rothermel. A Sensor Network Abstraction for Flexible Public Sensing Systems. *Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, S. 460–469, 2011. (Zitiert auf Seite 10)

- [PSA<sup>+</sup><sub>13</sub>] D. Philipp, J. Stachowiak, P. Alt, F. Dürr, K. Rothermel. DrOPS: Model-Driven Optimization for Public Sensing Systems. *Proceedings of the 11th IEEE International Conference on Pervasive Computing and Communications*, 2013. (Zitiert auf den Seiten 10, 13, 17, 18 und 29)
- [XSK<sub>10</sub>] Y. Xiao, P. Savolainen, A. Karppanen. Practical Power Modeling of Data Transmission over 802.11g for Wireless Applications. *e-Energy '10*, Apr 13-15, 2010. (Zitiert auf Seite 29)

### **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift