Quality-aware Coordination in Public Sensing

Von der Fakultät Informatik, Elektrotechnik und
Informationstechnik der Universität Stuttgart
zur Erlangung der Würde eines Doktors der
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Harald Weinschrott

aus Lugosch, Rumänien

Hauptberichter:	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Mitberichter:	Prof. Dr. rer. nat. habil. Pedro José Marrón
Tag der mündlichen Prüfung:	26. November 2012

Institut für Parallele und Verteilte Systeme (IPVS)
der Universität Stuttgart
2013

Acknowledgements

First of all, I want to thank my advisor, Prof. Dr. Kurt Rothermel, who has made this work possible in his group. I would like to thank him for the guidance and support he provided. I learned a lot about conducting scientific research and his ideas and advice gave me new insights into my work. Special thanks also go to Prof. Dr. Pedro José Marrón, who sparked my interest in the field of distributed systems, and kindly accepted to support this thesis as co-advisor.

I would like to thank the current and former members of the Distributed Systems group for the good working environment and the interesting discussions. Their encouraging, constructive, and valuable feedback helped me a lot to improve my work. Especially, I want to mention Frank Dürr, Dominique Dudkowski, Andreas Grau, Lars Geiger, Ralph Lange, and of course all the other members of the research group. Special thanks also to Julia Möhrmann for her valuable feedback and support.

I also want to thank all the students who have contributed to this work as part of their diploma or student thesis, or as student research assistants. Their effort and contribution to this project have made this thesis possible.

I would like to thank the German Research Foundation for their financial support through the Collaborative Research Center "Nexus", and the Baden-Württemberg Stiftung GmbH for partially funding this research through the "SpoVNet" project. This funding, which enabled my research in the first place, provided me the chance to present my results to the international research community.

Last but not least, I want to express my sincere thanks to my family and my friends. I want to thank my parents Anton and Anna-Eva for their continuous support and encouragement that has enabled me to fully focus on this work.

Contents

List of Abbreviations

1G	first generation cellular network
2G	second generation cellular network
3D	three-dimensional
3G	third generation cellular network
4G	fourth generation cellular network
AEROA	Adaptive Early Read Operation Avoidance
AN	associated node
AODV	Ad hoc On-demand Distance Vector
AWM	augmented world model
CP	context provider
CROA	Concurrent Read Operation Avoidance
CSS	Centralized Spatial Shaper
DHT	distributed hash table
DR	detection ratio
DROA	Distant Read Operation Avoidance
DSS	Distributed Spatial Shaper
EC	energy consumption
EDGE	Enhanced Data Rates for GSM Evolution
EEG	electroencephalography
EROA	Early Read Operation Avoidance
EROA/P	Proactive Early Read Operation Avoidance
EROA/R	Reactive Early Read Operation Avoidance
ETSI	European Telecommunications Standards Institute
FN	federation node
GHT	geographic hash table
GLONASS	Globalnaja Nawigazionnaja Sputnikowaja Sistema
GPRS	General Packet Radio Services
GPS	Global Positioning System

GPSR	Greedy Perimeter Stateless Routing
GSM	Global System for Mobile Communications
HSDPA	High-Speed Downlink Packet Access
ID	identifier
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPS	indoor positioning system
ISM	industrial, scientific and medical
LHT	Location Hierarchy Tree
LNG	Location Neighbor Graph
LTE	Long Term Evolution
MAC	media access control
MANET	mobile ad-hoc network
MSS	Minimum Subset Selection
NAVSTAR	Navigational Satellite Timing and Ranging
NCG	Node Connectivity Graph
NFC	near field communication
NHG	Node Hierarchy Graph
PC	personal computer
pdf	probability density function
POD	percentage of duplicates
psdf	partial spatial distribution function
PSS	Public Sensing Sever
RFID	radio-frequency identification
SAR	Symbolic Anycast Routing
SFB	Sonderforschungsbreich
SGR	Symbolic Geocast Routing
SLM	symbolic location model
SQL	Structured Query Language
TTL	time to live
UDEL	University of Delaware
UHF	ultra high frequency
UMTS	Universal Mobile Telecommunications System
UV	update validity

VANET	vehicular ad-hoc network
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	wireless local area network
WMAN	wireless metropolitan area network
WMN	wireless mesh network
WPAN	wireless personal area network
WSN	wireless sensor network
WWAN	wireless wide area network

Abstract

The evolution and proliferation of mobile sensing platforms such as mobile phones, enables services that analyze and adjust to the state of the real world. Billions of mobile phones around the globe seamlessly integrated into our life enable the vision of public sensing, i.e., monitoring and detecting a variety of physical phenomena by continuously collecting an abundance of sensor data. To exploit the enormous sensing capabilities, sensing may not interfere with normal operation of mobile phones. Furthermore, since mobile phones are battery powered, sensor data collection needs to be energy efficient and, thus, limited to the required data.

Therefore, this dissertation presents a public sensing approach that opportunistically collects sensor data. To specify and assess the quality of the data, spatial and temporal coverage metrics are devised. Virtual sensors are introduced as a data-centric abstraction to cope with the dynamic availability of mobile phones by decoupling applications from physical devices. More precisely, this dissertation addresses three major classes of virtual sensors that allow applications to request sensor data intuitively based on spatial, temporal, and quality requirements.

For each of the three classes of virtual sensors, this dissertation presents centralized and distributed algorithms for the selection and coordination of mobile phones according to the sensing requirements, while minimizing the energy consumption. In order to cope with the varying availability of physical sensors, the dissertation shows how to monitor the progress of sensing and how to adapt sensing to changes of movement. Moreover, this dissertation shows how to adapt the coordination mechanisms to the density of participating devices.

As a basis for the coordination algorithms, this dissertation presents basic group communication mechanism. These mechanisms allow to address specific devices based on their symbolic location. In essence, a routing structure that mimics the location model is created and proactively maintained. With a symbolic location model that matches the structure of the virtual sensors, this communication abstraction allows to easily identify and address nodes relevant for the coordinated data acquisition of the virtual sensors.

Zusammenfassung

Einleitung

In den letzten Jahren hat die Entwicklung mobiler Endgeräte und der Sensortechnologie dazu geführt, dass Milliarden leistungsfähiger Sensoren uns im täglichen Leben umgeben. Mittels dieser Geräte ist es möglich, den Zustand der realen Welt zu erfassen und Anwendungen entsprechend zu adaptieren. Die Umsetzung dieser Vision ist das Ziel des Forschungsgebiets *Public Sensing*. Durch die Ubiquität der mobilen Endgeräte werden Erfassungsszenarien möglich, die weit über jene der traditionellen Sensornetze hinausgehen. Zusätzlich zur Datenerfassung mit persönlichem Nutzen, sind unter anderem Szenarien in den Bereichen Forschung, Wirtschaft und Städteplanung denkbar.

Das Projekt BikeNet [EML$^+$09] setzt auf den persönlichen Nutzen der Datenerfassung, indem es Fahrräder mit diversen Sensoren ausgestattet hat, um Fahrten zu analysieren. Forscher haben aber auch vielfach auf den wissenschaftlichen Nutzen von Mobiltelefonen hingewiesen [ROE09, KP09]. Mittels mobiler Endgeräte lassen sich detaillierte Karten von Umweltphänomenen erstellen. So ist es beispielsweise möglich die Luftverschmutzung und deren Verteilung detailliert zu untersuchen, um geeignete Gegenmaßnahmen ergreifen zu können. Gleichzeitig sind solche Daten auch von großem öffentlichem Interesse, da sie es erlauben die Lebensqualität einzelner Gebiete einzuschätzen. Darüber hinaus erlaubt diese Technologie mobile Objekte zu verfolgen, um damit beispielsweise verlorene Objekte wiederzufinden [GBM08], oder detaillierte Verkehrsflussinformation bereitzustellen.

Die Verwirklichung dieser Szenarien hängt von der weiteren technologischen Entwicklung ab. Bereits heute sind *Smartphones* äußerst leistungsfähig. In Zukunft wird sich die Menge der Sensoren, die in diese Geräte integriert sind, noch weiter erhöhen. Darüber hinaus wird die Kommunikationsinfrastruktur vielfältiger werden und überall auf der Welt den Zugriff auf Sensordaten erlauben. Der Trend zur Partizipation an Projekten wie OpenStreetMap [HW08] zeigt, dass eine hohe Akzeptanz des *Public Sensing* zu erwarten ist.

Um jedoch großflächig *Public Sensing* betreiben zu können, müssen einige Herausforderungen bewältigt werden. Die Teilnahem an solchen Aktivitäten muss für die Besitzer der mobilen Endgeräte möglichst einfach sein und sie darf die primären Funktionen der Geräte nicht negativ beeinflussen. Dies erfordert in erster Linie eine energieschonende Datenerfassung und somit nur die Erfassung der tatsächlich benötigten Daten. Weiterhin bedeutet das aber auch, dass die Mobilität der Nutzer nicht kontrolliert werden kann. Darüber hinaus ist es notwendig die Qualität der erfassten Daten zu bewerten und gegebenenfalls auch Daten bestimmter Qualität anfordern zu können. Für beides sind entsprechende Metriken notwendig. Eine weitere Herausforderung ist die hohe Dynamik eines solchen Systems. Deshalb müssen Anwendungen von den physischen Geräten zur Datenerfassung entkoppelt werden.

Der Fokus dieser Dissertation lässt sich anhand mehrerer Dimensionen aufzeigen. Die erste Dimension ist die Größenordnung des betrachteten Systems. Hier liegt der Fokus auf großen Systemen, die die Kombination vieler Einzelmessung erfordern, um ein Phänomen als Ganzes abzubilden. Die zweite Dimension ist der Grad der Nutzerbeteiligung. Hier liegt der Fokus auf einem System das ohne aktive Teilnahme der Nutzer auskommt und somit die geringste Einschränkung für den Nutzer bedeutet. Die dritte Dimension betrifft das Kommunikationssystem. Da hier in Zukunft von hybriden Netzen auszugehen ist, legt diese Dissertation den Fokus auf eben diese. Diese Dissertation umfasst die Beobachtung von Umweltphänomenen und von mobilen Objekten.

Diese Dissertation liefert mehrere Beiträge zum Stand der Wissenschaft. Zunächst werden eine Schichtenarchitektur sowie eine datenzentrische Schnittstelle präsentiert. Kern dieser Schnittstelle sind die sog. virtuellen Sensoren, die es Anwendungen erlauben, Daten mit definiertem räumlichem, zeitlichem und qualitativem Bezug anzufordern. Der zweite Beitrag dieser Dissertation sind drei Metriken, um Qualitätsanforderungen zu spezifizieren: räumliche, zeitliche und räumlich-zeitliche Abdeckung. Der dritte Beitrag sind zentralisierte sowie verteilte Algorithmen zur Datenerfassung entsprechend der Anforderungen hinsichtlich der drei Metriken. Ziel dieser Algorithmen ist die effiziente Datenerfassung entsprechend der Anforderungen. Zum Umgang mit der Dynamik des Systems bietet die Dissertation Mechanismen zur Überwachung des Erfassungsfortschritts und zur Adaption der Erfassung. Darüber hinaus beschreibt die Dissertation Mechanismen zur Gruppenkommunikation basierend auf symbolischen Lokationsmodellen.

In den folgenden Abschnitten wird der Inhalt dieser Dissertation kurz zusammengefasst. Zunächst wird ein Überblick über die Architektur und die einzelnen Systemkomponenten gegeben. Schließlich werden die Mechanismen zur Gruppenkommunikation diskutiert, die die Grundlage für die Koordinationsalgorithmen sind, die im darauf folgenden Abschnitt beschrieben werden.

Systemübersicht

Wichtigste Anforderungen an das System sind die möglichst geringe Einschränkung der primären Funktionen der Endgeräte und damit auch die Energieeffizienz der Datenerfassung. Darüber hinaus soll auch die Kommunikation effizient sein. Die Datenerfassung soll entsprechend der Qualitätsanforderungen erfolgen und die Anwendungen sollen von den physischen Geräten entkoppelt werden. Darüber hinaus muss das System skalieren und mit der Mobilität der Geräte und den Dichteschwankungen umgehen können.

Das Systemmodell umfasst die mobilen Knoten sowie einen zentralen Server als Koordinator der Erfassung. Die mobilen Knoten verfügen über WiFi sowie Mobilfunkanbindung. Mittels WiFi können sie Ad-hoc Netze bilden und sich mit einem Mesh-Netz verbinden. Sie verfügen über einen GPS-Empfänger sowie diverse Sensoren: einen Sensor zur Erfassung von Umweltphänomenen, einen Annäherungssensor und zur Abfrage externer Sensoren eine Funkschnittstelle. Mobile Objekte sind mittels der Annäherungssensoren zu erfassen. Die externen Sensoren basieren auf RFID-Technologie und sind stationär. Die mobilen Objekte und die mobilen Knoten bewegen sich auf einem Straßennetz.

Ein symbolisches Lokationsmodell ist ähnlich definiert wie in [BD05]. Es besteht aus einer Menge von Lokationen. Zwischen diesen ist die Enthalten-sein-Beziehung definiert, woraus sich der *Location Hierarchy Tree* (LHT) ergibt. Zusätzlich definiert der *Location Neighbor Graph* (LNG) Nachbarschaftsbeziehungen zwischen Lokationen.

Das System hat eine geschichtete Architektur. Auf der untersten Ebene befindet sich die Kommunikationsschicht. Hier sind die Gruppenkommunikationsmechanismen und die Mechanismen zur Anbindung an die Infrastruktur angesiedelt. Die darüber liegende Schicht enthält Mechanismen zur Speicherung von Messwerten und anderen Daten. Insbesondere sind hier auch Mechanismen zur Erfassung des Zustands der Ressourcen, sowie der virtuellen Sensoren angesiedelt. Die darüber liegende Schicht ist die Schicht der virtuellen Sensoren. Diese Schicht enthält die Mechanismen zur koordinierten Datenerfassung.

Die Dissertation stellt die Integration dieses Systems in ein globales Overlay-Netz dar. Mittels eines Publish/Subscribe-Mechanismus wird der Nachrichtenfluss zur Anforderung der Daten sowie deren Bereitstellung erläutert.

Symbolisches und hybrides Routing

Symbolisches Routing erlaubt die Adressierung von Lokationen anhand symbolischer Bezeichner, wie etwa Straßennamen. Diese intuitive Adressierung eignet sich sehr gut zur Abstraktion physischer Knoten. Die Grundidee des entworfenen Routingverfahrens besteht darin, Routen entsprechend des Symbolischen Lokationsmodells aufzubauen. Somit dient das statische Lokationsmodell als Grundlage für das Routing. Die Routingstruktur besteht aus zwei Teilen: einer hierarchischen Struktur, die dem LHT nachempfunden ist und einer flachen Struktur, die dem LNG nachempfunden ist. Beide Strukturen sind Overlays, bestehend aus den sogenannten *Associated Nodes* (ANs). Die Knoten des physischen Netzes kennen Routen zu mehreren dieser Overlay-Knoten.

Das *Symbolic Anycast Routing* (SAR) basiert auf dieser Routingstruktur. Es liefert eine Nachricht zu einem Overlay-Knoten in der adressierten Lokation. Dazu wird eine Nachricht zunächst zum AN der Senderlokation geschickt. Von dort wird sie zu einem AN der hierarchisch übergeordneten Lokation gesendet. Dies wird solange wiederholt bis eine Lokation erreicht wird, die die Ziellokation enthält. Von dort wird die Nachricht schrittweise wieder zu hierarchisch untergeordneten Lokationen weitergeleitet bis das Ziel erreicht wird. Obwohl dieser Mechanismus effektiv ist, bietet er weiteres Potenzial für Optimierungen. Eine vorgestellte Optimierung ist das Ausnutzen von Abkürzungen in der Routingstruktur. Dabei wird eine Nachricht zu dem AN weitergeleitet, der der Ziellokation am nächsten ist.

Das *Symbolic Geocast Routing* (SGR) basiert auf SAR. Dieses Verfahren liefert eine Nachricht an alle Knoten aus, die sich in einer Lokation befinden. Die Grundidee hierbei ist es, zunächst mittels SAR eine Nachricht zu den ANs der Ziellokation zu schicken. Diese starten dann ein lokales Fluten der Nachricht innerhalb ihrer Lokation.

Der Aufbau der Routingstruktur basiert auf periodischen Nachrichten der ANs. Die Knoten schicken periodisch eine Nachricht, in der sie spezifizieren, wie hoch ihre Bereitschaft ist als AN ihrer Lokation zu fungieren. Nachbarknoten leiten diese Nachricht innerhalb der Lokation weiter, sofern ihre Bereitschaft niedriger ist. Durch diese periodischen Nachrichten werden Routen zu den ANs der lokalen Lokation aufgebaut. Darüber hinaus werden Routen zwischen ANs von Nachbarlokationen aufgebaut.

Wenn symbolische Lokationsmodelle nur partiell definiert sind, so ist die Übermittlung einer Nachricht von einer dieser Partitionen zu einer anderen mit einem rein symbolischen Routingverfahren nicht möglich. Für diesen Fall werden Mechanismen für das sogenannte hybride Routing skizziert.

Der erste vorgestellte Ansatz basiert auf der Verwendung des symbolischen Routings innerhalb eines symbolischen Lokationsmodells und der Verwendung von geographischem Routing außerhalb. Dazu werden die symbolischen Teilmodelle mit einer geographischen Koordinate annotiert. Darüber hinaus wird ein Lernverfahren eingeführt, um die Lokationen eines symbolischen Lokationsmodells zu bestimmen, von denen Knoten in Reichweite von Knoten außerhalb des Lokationsmodells sind. Um eine Nachricht nun von einer symbolischen Lokation in einer andere zu schicken, wird diese zunächst an eine Lokation geschickt von der Knoten außerhalb des Lokationsmodells erreichbar sind. Wird ein solcher Knoten erreicht, so beginnt dieser die Nachricht per geographischem Routing an die Position des Ziels weiterzuleiten. Sobald ein Knoten erreicht wird, der innerhalb des Lokationsmodells des Ziels ist, beginnt wieder das symbolische Routing.

Der zweite Ansatz basiert darauf, die einzelnen symbolischen Lokationsmodelle zu einem Modell zu integrieren. Basierend auf einer Grid-Struktur werden symbolische Lokationen im Bereich zwischen den Teilmodellen definiert. Auf diesem integrierten Modell ist dann durchgehend symbolisches Routing möglich.

Es wird gezeigt wie Mechanismen zur Datenverwaltung mittels der konzipierten Routingverfahren realisiert werden können. Dazu werden die Verfahren zur geographischen Datenverwaltung [Dud09] angepasst. Die Grundidee dieses Verfahrens ist es, Daten, die innerhalb einer Lokation erfasst werden, bei ANs dieser Lokation zu speichern. Eine Anfrage kann dann dadurch bearbeitet werden, dass sie per symbolischen Routings zu dieser Lokation geleitet wird.

Zur Evaluierung der vorgestellten Routingverfahren werden umfangreiche Simulationen durchgeführt. Dabei werden die Rate der erfolgreich ausgelieferten Nachrichten, die resultierende Routenlänge und der Aufwand zum Aufbau der Routingstruktur untersucht. Es wird gezeigt, dass in einem stationären Szenario nahezu alle Nachrichten erfolgreich ausgeliefert werden. Die Länge der Routen ist in den untersuchten Szenarien lediglich 23% höher als die optimale Route. In einem mobilen Szenario wird gezeigt, dass die Rate der erfolgreich ausgelieferten Nachrichten nur langsam mit der Reduktion des Intervalls des Routingstrukturaufbaus fällt.

Koordinationsalgorithmen

Virtuelle Sensoren sind eine datenzentrische Abstraktion für den Zugriff auf Messwerte eines bestimmten Typs, ohne die physischen Knoten kennen zu müssen. Jedem virtuellen Sensor ist ein räumliches Gebiet zugeordnet. Die Ausgabe eines virtuellen Sensors sind sogenannte virtuelle Messwerte. Diese setzen sich aus einer Menge physischer Messwerte, die innerhalb des Gebiets des virtuellen Sensors erfasst wurden, zusammen. Drei Parameter bestimmen die Qualität der Daten, die ein virtueller Sensor bereitstellen soll. Der erste Parameter ist V_δ. Er bestimmt in welchem Zeitraum die Erfassung eines virtuellen Messwerts abgeschlossen sein soll. Der zweite Parameter ist V_s. Er bestimmt mit welchem Intervall virtuelle Messwerte erfasst werden sollen. Schließlich bestimmt V_k die geforderte Redundanz der physischen Messwerte.

Das primäre Ziel der Algorithmen zur Realisierung punktförmiger virtueller Sensoren besteht darin, die geforderte zeitliche Abdeckung effizient zu gewährleisten und somit das spezifizierte Intervall zur Bereitstellung virtueller Messwerte einzuhalten. Die Algorithmen lösen dabei drei grundlegende Probleme. Zunächst werden nur dann Messwerte erfasst, wenn dies auch notwendig ist. Die Annäherung von Knoten an das Gebiet eines virtuellen Sensors wird effizient detektiert. Schließlich werden Kollisionen bei der Datenerfassung vermieden und gleichzeitig aber die Redundanz der Knoten ausgenutzt, um erfolgreich Daten zu erfassen.

Um die geforderte zeitliche Auflösung einzuhalten wird sowohl ein zentralisierter als auch ein verteilter Ansatz präsentiert. Das zentralisierte Verfahren beruht auf einem periodischen Notifikationsmechanismus. Das verteilte Verfahren beruht auf der Koordination der mobilen Knoten in einem Ad-hoc Netz. Der Zeitpunkt des letzten virtuellen Messwerts wird dabei den direkten Nachbarn von den Knoten mitgeteilt, die die Erfassung durchgeführt haben. Bevor ein Knoten nun einen virtuellen Messwert aufnimmt, schickt er eine Anfrage an seine direkten Nachbarn. Diese antworten mit dem ihnen bekannten letzten Erfassungszeitpunkt. Anhand dieser Information entscheidet dann der Knoten ob aktuell ein Messwert aufgenommen werden muss.

Um effizient die Annäherung an einen virtuellen Sensor zu detektieren, werden Informationen über die zu aktualisierenden virtuellen Sensoren einbezogen. Es wird die Entfernung bestimmt, die ein Knoten zurücklegen kann bevor er potenziell in den Bereich eines zu aktualisierenden virtuellen Sensors gelangt. Zur Bestimmung dieser Entfernung wird außerdem die Positionsunschärfe berücksichtigt.

Das Verfahren zur Kollisionsvermeidung basiert auf zwei Wahrscheinlichkeiten für die erfolgreiche Aktualisierung eines virtuellen Sensors. Die erste Wahrscheinlichkeit ist die eines einzelnen Knotens. Abhängig von der Entfernung zum virtuellen Sensor und der Positionsunschärfe, kann ein Knoten bestimmen, mit welcher Wahrscheinlichkeit er tatsächlich in Sensorreichweite ist. Die zweite Wahrscheinlichkeit ist die der Gruppe von Knoten, die aktuell in der Nähe des virtuellen Sensors sind. Im Ad-hoc Netz tauschen die Knoten die individuellen Wahrscheinlichkeiten aus, um so zu bestimmen, welche Knoten gleichzeitig Messwerte aufnehmen müssen, um am effizientesten zusammen eine gewisse Erfolgswahrscheinlichkeit zu erzielen. Um Kollisionen zu vermeiden, wird ein Slot-basiertes Leseverfahren eingesetzt.

Darüber hinaus wird ein Adaptionsverfahren vorgestellt, welches die Anzahl der bei der Datenerfassung partizipierenden Knoten reduziert. Durch Spezifikation einer Toleranz bei der Erfüllung der Qualitätsanforderungen lässt sich der Energieverbrauch senken.

Zur Erfassung von Umweltphänomenen mittels segmentförmiger virtueller Sensoren werden Algorithmen entwickelt, die die geforderte räumliche Abdeckung effizient gewährleisten. Zunächst wird hierfür die räumliche Abdeckung eines Segments definiert. Danach wird ein zentralisierter Algorithmus vorgestellt, der die Erfassung so koordiniert, dass die geforderte räumliche Abdeckung möglichst effizient erreicht werden kann. Danach wird ein verteilter Algorithmus eingeführt, der ad-hoc Kommunikation nutzt, um eine verteilte Koordination durchzuführen.

Die räumliche Abdeckung eines Segments ergibt sich dadurch, dass jeder Punkt des Segments abgedeckt ist. Ein Punkt ist abgedeckt wenn ein Messwert in Sensorreichweite des Punktes aufgenommen wird. Abdeckungslücken an einigen Punkten des Segments werden nicht durch mehrfache Abdeckung anderer Punkte kompensiert.

Das Ziel des zentralisierten Algorithmus ist es, diejenigen Knoten auszuwählen, die die geforderte Abdeckung erzielen können, und diesen dann zu erledigende Teilaufgaben zuzuordnen. Anhand der vorhergesagten Bewegung eines Knotens kann dessen mögliche Abdeckung abgeschätzt werden. Eine zentrale Instanz fordert diese Vorhersagen von den

Knoten an und bestimmt daraus welche Mindestmenge von Knoten an der Erfassung beteiligt werden sollen.

Sofern mehrere der gewählten Knoten gewisse Gebiete abdecken können, erfolgt eine eindeutige Zuweisung der Erfassung dieser Gebiete zu den Knoten. Da Knoten von der vorhergesagten Bewegung abweichen können, erfolgt eine dynamische Anpassung dieser Zuweisungen. Hierfür aktualisieren die Knoten ihre Vorhersage, wenn sie von dieser abweichen. Schließlich wird ein Mechanismus vorgestellt, um auf den Ausfall von Knoten reagieren zu können.

Der verteilte Algorithmus verfolgt das gleiche Ziel wie der zentralisierte. Allerdings erfolgt hier die Koordination verteilt mittels eines Ad-hoc Netzes. In einer ersten Phase tauschen die Knoten ihre Vorhersagen bezüglich der möglichen Abdeckung untereinander aus. Somit erhält jeder Knoten eine lokale Sicht auf die Knoten eines Segments. Anhand dieser lokalen Sicht bestimmen die Knoten jeweils lokal, welche Knoten Daten erfassen sollen. Die Knoten verifizieren relevante Selektionsentscheidungen, um Fehler durch Inkonsistenzen zwischen den lokalen Sichten zu vermeiden. Ebenso wie beim zentralen Ansatz, erfolgt auch hier die Überwachung des Erfassungsfortschritts.

Um zu vermeiden, dass Knoten kontinuierlich Daten erfassen während sie sich entlang eines Segments bewegen, wird ein Verfahren zur Reduktion der Anzahl der Leseoperationen entwickelt. Die Idee hierbei ist es, das zeitliche Intervall der Leseoperationen anhand der Geschwindigkeit der Knoten und der Sensorreichweite zu bestimmen.

Das Ziel der Algorithmen zur Realisierung segmentförmiger virtueller Sensoren zur Erfassung mobiler Objekte besteht darin, die räumlich-zeitliche Abdeckung zu maximieren, so dass die Wahrscheinlichkeit mobile Objekte zu finden erhöht wird. Zunächst wird die räumlich-zeitliche Abdeckung, die im Gegensatz zur rein räumlichen Abdeckung, die Geschwindigkeit der mobilen Objekte berücksichtigt, diskutiert. Aufbauend auf den definierten Metriken zur Abdeckung werden ein optimistisches und ein pessimistisches Verfahren zur Koordination der Erfassung vorgestellt.

Zunächst wird in der Dissertation die räumlich-zeitliche Abdeckung einer einzelnen Leseoperation in Abhängigkeit der angenommen Objektgeschwindigkeit diskutiert. Es wird gezeigt, dass eine Leseoperation zu einem gewissen Zeitpunkt nicht nur einen räumlichen Bereich abdeckt, sondern auch zu einem früheren oder späteren Zeitpunkt eine gewisse räumliche Abdeckung hat. Im Weiteren wird die Abdeckung durch überlappende Leseoperationen erläutert. Schließlich wird dargestellt wie detaillierte Informationen über die Geschwindigkeitsverteilung der Objekte dazu genutzt werden können, um eine Abdeckungsmetrik zu definieren, die es erlaubt anhand der Anzahl der detektierten Objekte auf die Gesamtanzahl der Objekte zu schließen. Mit dieser Metrik kann anhand weniger detektierter Objekte auf die Gesamtzahl der Objekte geschlossen werden.

Die räumlich-zeitliche Abdeckung hängt davon ab, ob sich zwei lesende Knoten treffen oder nicht. Ob es zu einem solchen Treffen kommt ist allerdings nicht im Vorhinein bekannt. Das Ziel des pessimistischen Ansatzes ist es, die maximale Abdeckung zu erzielen. Knoten deaktivieren deshalb nur dann die Datenerfassung, wenn sie sich in einem bereits abgedeckten Bereich bewegen. Um effizient festzustellen ob Knoten sich treffen wird ein adaptives Positionierungsintervall eingesetzt.

Der verteilte Algorithmus verfolgt einen optimistischen Ansatz. Knoten deaktivieren bereits dann ihre Erfassungsaktivität wenn Abdeckungsvorhersagen darauf hindeuten, dass sie sich in einem Gebiet bewegen, das wahrscheinlich abgedeckt sein wird. Um dies verteilt bestimmen zu können, senden Knoten ihre vorhergesagte Trajektorie an ihre direkten Nachbarn im Ad-hoc Netz. Auf diese Art bauen die Knoten eine lokale Sicht auf die Bewegung anderer Knoten auf dem Segment auf. Damit bestimmt jeder Knoten welchen Beitrag zur Abdeckung des Segments er leisten kann und deaktiviert gegebenenfalls seine Erfassungsaktivitäten.

Ähnlich dem Verfahren zur Verringerung der Leseoperationen, die für die räumliche Abdeckung notwendig sind, wird ein entsprechendes Verfahren für den Fall der räumlich-zeitlichen Abdeckung vorgestellt. Die Idee hierbei ist es, das zeitliche Intervall der Leseoperationen anhand der Geschwindigkeit von Knoten und Objekten sowie der Sensorreichweite zu bestimmen.

Zur Evaluierung der vorgestellten Algorithmen wurden umfangreiche Simulationen durchgeführt. Dabei bewegen sich die mobilen Knoten sowie die mobilen Objekte auf einem Straßengraphen. Für die Bestimmung des Energieverbrauchs der Knoten wird ein detailliertes Energiemodell eingesetzt. Zunächst präsentiert die Dissertation die Ergebnisse der Evaluierung der punktförmigen virtuellen Sensoren. Darauf folgen die Ergebnisse zum segmentförmigen virtuellen Sensor zur Erfassung von Umweltphänomenen, gefolgt von den Ergebnissen zum virtuellen Sensor zur Detektion mobiler Objekte.

Bei den punktförmigen virtuellen Sensoren wird zunächst die Effektivität untersucht. Die entwickelten Verfahren erzielen die gleiche zeitliche Abdeckung wie ein optimales Verfahren, bei dem die Knoten keine notwendige Leseoperation verpassen. Diese Effektivität ist unabhängig von der Anzahl der Knoten im System, der Geschwindigkeit der Knoten und der Positionsunschärfe gegeben. Die Effizienz in Bezug auf den Energieverbrauch der Knoten wird ebenfalls gezeigt. Die entwickelten Verfahren erzielen einen bis zu 95% niedrigeren Energieverbrauch als ein Vergleichsverfahren, welches Leseoperationen unabhängig von anderen Knoten durchführt. Eine ähnliche hohe Reduktion des Energieverbrauchs kann auch für unterschiedliche Werte von V_s beobachtet werden. Darüber hinaus wird auch die Effizienz in Abhängigkeit der Positionsunschärfe und der Anzahl der virtuellen Sensoren gezeigt. Danach wird die Anzahl der redundanten Messwerte bestimmt. Hier kann die Dissertation zeigen, dass die vorgestellten Verfahren nahezu keine Redundanz erzeugen. Dies wird für verschiedene Knotenanzahlen, für unterschiedliche Werte von V_S, für verschiedene Positionsunschärfen und unterschiedliche Knotengeschwindigkeiten gezeigt.

Für die segmentförmigen virtuellen Sensoren wird zunächst ebenfalls die Effektivität untersucht. Hier zeigen die Simulationsergebnisse, dass die vorgestellten Verfahren nahezu die gleiche räumliche Abdeckung erzielen wie ein optimales Verfahren bei dem Knoten kontinuierlich Daten erfassen. Diese Effektivität ist gegeben für unterschiedliche Anforderungen bezüglich V_k und unterschiedliche Knotendichten. Als nächstes wird die Energieeffizienz untersucht in Abhängigkeit von V_k, in Abhängigkeit von V_δ und in Abhängigkeit der Knotendichte. Die Simulationen zeigen, dass die vorgestellten Verfahren deutliche Energieeinsparungen ermöglichen.

Anschließend folgen die Ergebnisse der virtuellen Sensoren zur Detektion von Objekten. Die Simulationsergebnisse zeigen die Effektivität der vorgestellten Verfahren in Bezug auf die erzielte räumlich-zeitliche Abdeckung. Hier kann gezeigt werden, dass das pessimistische Verfahren die gleiche Abdeckung erzielt wie ein optimales Verfahren. Dies wird gezeigt für verschiedene Knotendichten sowie für unterschiedliche Objektgeschwindigkeiten. Das optimistische Verfahren erzielt eine leicht niedrigere Abdeckung, da Abweichungen der Knoten von ihrer vorhergesagten Bewegung zu Lücken in der Abdeckung führen können. Als nächstes wird die Effizienz der Verfahren verglichen. Hier zeigt sich, dass das pessimistische Verfahren nur dann zu einem verringerten Energieverbrauch führt, wenn die Kommunikationskosten niedrig sind. Das optimistische Verfahren erzielt einen noch niedrigeren Energieverbrauch als das pessimistische Verfahren und erreicht bis zu 63% Einsparung gegenüber kontinuierlichem Lesen.

Schließlich wird untersucht wie gut die räumlich-zeitliche Abdeckungsmetrik die Erfassungsquote von Objekten wiederspiegelt. Zunächst wird gezeigt, dass detailliertes Wissen über die Geschwindigkeitsverteilung der Objekte notwendig ist. Dann werden einzelne Ursachen für Abweichungen der Abdeckung von der Entdeckungsquote analysiert und der Umgang mit ihnen diskutiert.

1 Introduction

1.1 Motivation

Over the last few years, the evolution and proliferation of mobile sensing platforms, such as mobile phones, has brought services within grasp that take real world situations into account. Such services will not only adjust to the whereabouts and the profile of a user, as known from location-based services [HD09], but also to the state of the real world. In this vision, sensors that are seamlessly integrated into our everyday life through billions of mobile phones all around the globe, will monitor and detect a variety of physical phenomena by continuously collecting an abundance of sensor data.

It is widely believed that within the next years billions of instrumented mobile phones will implement what is today actively researched in the field of *public sensing*. In the literature, this is also often referred to as community sensing [KHKZ08], urban sensing [CHK08], or people-centric sensing [CEL+08]. Exceeding the scale of traditional wireless sensor networks, networks of mobile phones can easily address metropolitan-scale scenarios and provide information about environmental or social phenomena. Various scenarios are conceivable. Beyond sensing for personal use, applications proposed in the literature span fields such as research, commerce, urban planning, society, and the public interest.

One exemplary scenario for sensing with personal use is BikeNet [EML+09]. It relies on bicycles instrumented with various sensors to collect data about a cyclist's rides. Regarding research, scientists from various fields have proposed smartphones as a tool for research [ROE09, KP09] beyond using dedicated sensing hardware. Moreover, this enables citizens to be active scientists themselves [PHH08]. While, for instance, scientists can study the air pollution distribution in urban areas [KP09], the same map of air pollution can be used for urban planning. Based on spatio-temporal maps of environmental phenomena, city administrators can develop a city according to the behavior and the needs of its inhabitants. For instance, they assess the quality of living and take actions to improve the quality of living or even to prevent health risks. The same data might also be useful to the public. Citizens could estimate the surrounding noise level of accommodations. Moreover, tracking of mobile objects allows for applications ranging from lost-and-found scenarios [GBM08] to the tracking of objects, such as buses, to estimate accurate arrival times. Another scenario is the collection of traffic information, e.g., the number of cars on a street segment. Traffic control systems could use this information to reroute traffic or impose temporary speed limits.

The realization of these scenarios depends on future trends. Tremendous technological advances such as miniaturization of computing and sensing devices already allow for prototypical public sensing systems based on off-the-shelf smartphones. In the future, an increasing number of sensors in mobile devices will further boost the potential of public sensing. The trend towards global connectivity of mobile devices to the Internet through hybrid network structures consisting of WWANs, WMANs or WLANs will leverage instant access to global information provided by public sensing. Beyond the trend of people to participate in community-based projects such as OpenStreetMap [HW08] or Awekas [Awe11] that collect measurements and map the environment, the large potential of public sensing comes from the fact that the aggregate mobility of people carrying mobile phones lead to large opportunistic sensor coverage [LBD+05].

Several challenges need to be addressed in order to realize a large scale public sensing system that fully exploits the potential of the described trends. First, in order to benefit from the large opportunistic coverage, sensing must not be obtrusive, i.e., sensing must not require user interaction and control of user mobility. Furthermore, it must not interfere with the normal operation of the participating users' phones. Since mobile phones are battery powered, this implies energy efficient sensor data collection. Moreover, this means that sensing needs to be limited to the required data. The second challenge that follows from the opportunistic sensing is to incorporate quality information of the sensed data. On the one hand, such quality information allows for assessing the utility of the sensed data that is prone to spatial and temporal quality variations. On the other hand, it allows for specifying quality requirements and, thus, provide data with the requested quality. The third major challenge is the varying availability of phones and their mobility. Therefore, applications need to be decoupled from physical sensors [KZ07]. In essence, data requirements need to be specified independently from physical sensor sources, and dynamically mapped to available physical sensors.

This dissertation targets these challenges by a set of mechanisms and algorithms for selecting and coordinating sensors, in order to efficiently provide sensor data with the required quality and, thus, enable the presented scenarios. A more detailed view on the focus and contribution of this dissertation is provided in Section 1.2 and in Section 1.3, respectively. Finally, Section 1.4 outlines the structure of this dissertation.

1.2 Focus

The scope of this dissertation is classified along several dimensions. The first dimension is the *scale of sensing*. According to [CEL+08], this dimension ranges from personal sensing via social sensing to public sensing. While personal sensing and social sensing is focused on sensing for personal use or for the use of a social group, public sensing aims to monitor phenomena relevant to the public. With the goal of sensing for the public comes the need to compose individual measurements to an overall view of the observed phenomena, which might span large areas and periods of time. This dissertation addresses this challenge and focuses on the most complex class of the sensing scale, namely public sensing.

The second dimension is the *conscious human involvement*. Lane et al. [LEM+08] distinguish two basic roles of human interaction patterns: participatory sensing and opportunistic sensing. In a participatory sensing system, users actively pursuit sensing goals by consciously visiting sensing sites, or sensing requires some kind of user interaction. For instance, users might be prompted by their phone to take a picture. In contrast, opportunistic sensing does not require any conscious user interaction. Users participate only by providing resources of the mobile devices they carry around. Since strong human involvement negatively affects the willingness of users to participate, this thesis puts focus on opportunistic sensing to allow for detailed monitoring of large scale phenomena. Since opportunistic sensing does not allow for guarantees on the data quality of sensing, a best effort system is devised.

The third dimension is the *communication system*. The tremendous growth of mobile data traffic (predicted 240 times increase in the period 2007-2014 [RA10]) that accompanies the rapid proliferation of smartphones over the last years, requires mobile operators to extend the capacity of their networks. Moreover, mobile data traffic is expected to grow faster than the capacity of cellular networks through technological upgrades [Han09]. To increase the capacity for mobile traffic and to limit expensive investments, offloading cellular traffic is getting into focus of mobile operators and research [BMV10, LRL+10, RLBCE11]. Beyond offloading through access points of wireless local area networks, wireless metropolitan area networks, such as mesh networks [ACVS10], are considered. This development suggests that future wireless connectivity will be heterogeneous. Although most urban areas will be covered by cellular networks, large urban areas will also be covered by WiFi networks. Therefore, in this dissertation, hybrid communication networks are considered that consist of cellular networks, and WiFi networks such as mesh networks or ad-hoc networks.

In addition to these dimensions, the following properties characterize the scope of this dissertation. It addresses the observation of two basic types of physical phenomena, namely the monitoring of environmental data, such as noise, temperature or humidity, and on detecting mobile objects using proximity sensors based on, for instance, Bluetooth or radio-frequency identification (RFID). The data size of a measurement of such sensors is in the range of several bytes, and a sensor returns a measurement almost instantaneously when queried. The dissertation addresses the monitoring of stationary phenomena such as the environmental conditions at a specific location or within a certain region, and the detection of mobile objects.

1.3 Contributions

This dissertation has several contributions within the field of large-scale opportunistic public sensing in urban areas, and enables services that are aware of the real-world. More precisely, this dissertation has the following individual contributions.

1. A layered architecture and the data-centric interface to build and interact with such a public sensing system.

2. Based on the characteristics of the phenomena that are to be observed, this dissertation introduces three major classes of so called virtual sensors that allow applications to request sensor data intuitively based on spatial, temporal, and quality requirements.

3. Metrics for specifying the quality requirements and for measuring the resulting data quality. First, this dissertation introduces a quality metric for specifying and measuring the temporal coverage of sensing. This metric is complemented by a spatial coverage metric. Finally, for the case of observing mobile objects, this dissertation presents a spatio-temporal metric.

4. Algorithms for the selection and coordination of sensors according to sensing requirements. For each of the three specific classes of virtual sensors, this dissertation contributes centralized and distributed algorithms that aim for fulfilling the quality requirements while minimizing the acquisition costs in terms of energy consumption of the battery-powered mobile phones.

5. In order to cope with the dynamic availability of sensors, the dissertation shows how to monitor the progress of sensing and how to adapt sensing to changes in sensor density and sensor movement.

6. Basic group communication mechanism as basis for the coordination algorithms. These mechanisms allow to address specific devices based on their symbolic location. With a symbolic location model that matches the structure of the virtual sensors, this communication abstraction allows to easily identify and address nodes relevant for the coordinated data acquisition of a virtual sensor.

7. In extensive simulative evaluations, this dissertation shows the effectiveness of the proposed algorithms in a broad range of scenarios. Moreover, it shows the efficiency of the data acquisition, and the efficiency of the basic communication mechanisms.

1.4 Structure

The remainder of this dissertation is structured as follows. Chapter 2 presents information about the context of this work. It provides background information on context-aware systems and presents their relevance for this dissertation. It provides a brief introduction to public sensing, its challenges and its applications. Afterwards, this chapter discusses characteristics of positioning systems and wireless communication.

Chapter 3 presents an overview of the system. First, the system requirements are stated. Second, the system model is defined. Then, the layered architecture of this system is introduced and discussed. The relation of its different components is explained. Finally, the system interface is sketched.

Communication mechanisms that are the basis for discovering and notifying mobile phones are presented in Chapter 4. Beyond symbolic routing mechanisms, this chapter presents extensions to these mechanisms that support hybrid location models, where symbolic location information is only partially defined. The effectiveness and efficiency of these mechanisms is afterwards evaluated, and the related work to this contribution is shown.

Chapter 5 presents the coordination algorithms for the efficient acquisition of data with defined quality requirements. As a common prerequisite, the concept of virtual sensors is devised. Based on characteristics of the phenomena that is to be observed by these virtual sensors, different algorithms are presented. First, algorithms for coordinating the monitoring of phenomena at a specific location. Second, algorithms for monitoring of stationary phenomena along a road segment. Finally, algorithms that take the mobility of the phenomena or objects that are observed into account. The algorithms are evaluated and the chapter is concluded with a discussion of related work.

Finally, Chapter 6 concludes this dissertation with a summary, concluding remarks and an outlook on future research directions.

2 Background

In this chapter, relevant characteristics and requirements imposed on the mechanisms and algorithms developed in this thesis are highlighted and discussed. First, context-aware systems are briefly discussed in Section 2.1. Then, in Section 2.2, the field of public sensing is introduced, and sensing capabilities and characteristics are discussed. Positioning and its inherent inaccuracies are briefly discussed in Section 2.3. Finally, in Section 2.4 wireless communication, which is the basis for the connectivity of mobile phones in urban areas, is discussed along with its challenges.

2.1 Context-aware Systems

Context is the information that characterizes the situation of a user or an entity. This includes, for instance, its position and the environmental conditions of the surrounding. Dey defines context in [Dey01] as follows:

> *"Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."*

Context-aware systems adapt to the context of users. They monitor the context of users and provide relevant information and services. Since different users or applications might depend on the same context information, it is beneficial to manage context information in structured context models [BCQ$^+$07] that provide each user with a relevant subset of information. According to [SBG99, RDD$^+$03], identity, time and location are the most basic properties of entities and referred to as primary context. In contrast, secondary context comprises all other context information that describe other aspects of entities, such as temperature or humidity. Primary context is often used as an index when selecting information from context models.

Over the last decade, a large number of context-aware systems have been developed. For a detailed survey of these systems and their characteristics see [BDR07, Kja07]. For an overview of the challenges addressed in these systems see [LPL09a]. In the following, the Nexus project, in which parts of the research for this dissertation was conducted, is exemplarily presented.

2.1.1 Collaborative Research Center Nexus (SFB627)

The Collaborative Research Center 627 (Nexus)[1] aims for sharing and federating distributed spatial context models [LCG+09]. It investigates a broad range of fundamental mechanisms required to use context information at a global scale and, thus, enable the vision of a *World Wide Space*. The World Wide Space, in analogy to the *World Wide Web*, is seen as a common basis for future context-based applications. Each context provider can include its models into a virtual space, a digital representation of the real world, that forms the conceptual and technological basis for the integration and the access of context models.

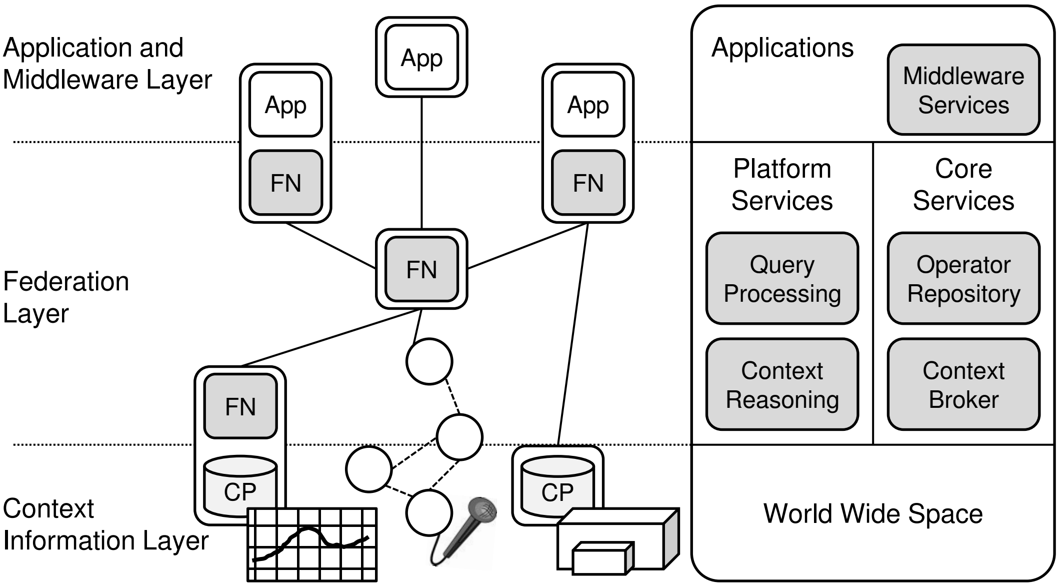

Figure 2.1: Simplified architecture of Nexus (adopted from [LCG+09])

The basic architecture of Nexus (cf. Figure 2.1) consists of three layers. At the top is the *Application & Middleware Layer*. It consists of context-aware applications and middleware services that facilitate application development. Applications and services of this layer rely on the context models provided through the *Federation Layer*. It provides a unified and federated view on the different context models provided by the bottom layer. The Federation Layer [GBH+05, CEB+09] consists of federation nodes (FN) forming a hybrid network that consists of infrastructure-based servers, as well as mobile devices. Core services running on these nodes form a distributed platform that enables services such as distributed query processing or context reasoning. The bottom layer is referred to as *Context Information Layer*. It consists of context providers (CP) [LR02, BBHS03] whose context

[1] German: Sonderforschungsbreich (SFB) 627

models include dynamic sensor data, static context, or data histories. Mobile sources are integrated on various layers into the infrastructure-based network [DWM08].

Nexus supports stream-based data processing [CEB+09] on the Federation Layer. Application queries are resolved on the nodes of the Federation Layer by continuously processing data in an operator graph [CSM11]. This operator graph considers the specific requirements provided through application queries, and integrates and transforms the context information of relevant providers to query results.

The extensible data model of Nexus, the *Augmented World Model* (AWM) [NM04], is based on object-oriented concepts. It defines a basic set of entities and is easily extended with models for specialized entities. A major goal of Nexus is to incorporate information about the quality of context into the context models and into context processing [GHLW09]. Beyond the accuracy of context information [LWG+09], inconsistencies between context models [Pet09] and reliability of individual context providers are modeled [Gut07].

2.2 Public Sensing

Creating a wide range of context-aware systems, i.e., services that are aware of the real world, requires detailed and large-scale context models. Although static context models such as city maps or building outlines are important context models, they only represent a small fraction of the context information describing the state of our world. Most aspects of our world are highly dynamic. This includes, for instance, the position of people, the temperature, and most other environmental parameters. A context model that provides information about these dynamic aspects requires continuous monitoring. However, the complexity of such models renders attempts to manually provide this dynamic information unfeasible.

The need for automated generation of context models is one of the driving forces behind the rapid advances in sensing technology, which has been experienced over the last few years. At the beginning of this development the goal was to build dedicated sensing systems for the autonomous monitoring of specific environmental conditions of the physical world. These sensing systems have drawn a lot of interest by a large research community, whose main research focus are wireless sensor networks (WSNs) [EGHK99, ASSC02, RM04, YMG08]. Special sensor devices, the nodes of these networks, have the ability to sense their environment, to process the measurements, and transmit the resulting information using wireless multi-hop communication without any infrastructure support to stationary or mobile sinks.

WSNs can be used for unobtrusive and autonomous monitoring of the physical world in a variety of specialized scenarios including, but not limited to, habitat monitoring [SOP+04], natural disaster relief [CEQM+04], and infrastructure health monitoring [KAB+05, KPC+07]. However, without infrastructure support, the deployment and the maintenance of battery-powered, dedicated sensing devices is expensive and, in public urban areas, it even requires compliance to administrative workflows that induce further efforts. There has been done some research on sensor nodes especially tailored to urban environments [BKM+04, MGT+07]. For instance, CitySense [MGT+07] proposes the use of stationary embedded PCs as nodes of an urban-scale sensing testbed. However, amongst other reasons [AAB+07], these approaches still require the deployment of a dedicated sensing infrastructure and, thus, prevent a wide deployment of WSNs in urban areas.

Complementary to WSNs, a trend towards the integration of a variety of cheap sensors into off-the-shelf hardware, such as mobile phones, can be observed. Since these powerful mobile devices, with advanced processing and communication capabilities, are omnipresent in populated areas, they form a rich ubiquitous [Wei91] sensing and communication infrastructure in urban areas. Steed and Milton [SM08] even show that it is feasible to create fine-grained maps of environmental parameters using tracked mobile sensors. Moreover, [CM06] shows that mobile phones acting as mobile sinks can be used to collect measurements from stationary sensors. A detailed overview of characteristics and challenges of mobile phone sensing provides [LML+10].

2.2.1 Sensing Technology

Current smartphones such as the Apple iPhone 4S[2] or the Galaxy Nexus[3] already include a large number of sensors as shown in Table 2.1.

Ambient light and proximity sensor are added to a mobile phone for simple context detection. The primary goal of a light sensor is to adjust the brightness of the display depending on the brightness of the surrounding. The proximity sensor is used to prevent unintended input on keys or touchscreen while the user is speaking on the phone. The primary intention for integrating accelerometers into smartphones was to rotate the screen. However, as several projects have shown, accelerometer data characterize the movement of the entity carrying the mobile phone.

Bao and Intille [BI04] rely on five biaxial accelerometers worn on different parts of the body to detect user activities such as walking, running, and bicycling. In more recent work [PLV+10], accelerometer data from smartphones is used to detect a user moving up- or downstairs and elevators in order to provide floor-level localization in buildings. The challenge when using accelerometer data from smartphones for user activity detection is that the relative position of a smartphone to its carrier is not fixed and varies over time. Recently, mechanisms have been developed that deal with this issue and robustly detect user

[2] Apple iPhone 4S – `http://www.apple.com/iphone/`

[3] Galaxy Nexus – `http://www.google.de/nexus/`

Table 2.1: Sensors integrated in current smartphones

Sensor	Apple iPhone 4S	Galaxy Nexus
ambient light	✓	✓
proximity	✓	✓
accelerometer	✓	✓
compass	✓	✓
gyroscope	✓	✓
barometer	×	✓
GPS	✓	✓
GLONASS	✓	×
dual cameras	✓	✓
microphone	✓	✓
Bluetooth	✓	✓
NFC	×	✓

activities [KLLK10, KWM10]. Besides user activity recognition, accelerometer data also proves to be a source for reliable pothole detection [EGH$^+$08]. By analyzing accelerometer data from smartphones in vehicles, possibly dangerous potholes can be identified and reported for the purpose of repair. Moreover, PhonePoint Pen [ACGC09] proposes to use a mobile phone like a pen and write short messages in the air. Using accelerometer data, the human writing is recognized.

The electronic compass in a smartphone allows for navigating in augmented-reality applications. Together with gyroscope and barometer it improves the accuracy of positioning and adds orientation information to a user's position. As an alternative, uDirect [HGT11] derives the direction of a walking user from accelerometer data. Besides the GPS receiver that is integrated in almost every smartphone, current smartphones start shipping with GLONASS receivers, as well. Moreover, future smartphones will also support the European satellite navigation system Galileo. A more detailed review of positioning systems can be found in Section 2.3. From position traces of mobile phones, significant places and activities of a person can be derived [LFK07]. In previous work, we developed mechanisms for the automatic road validation and correction based on collected traces from mobile phones [BWDR11]. The collected traces are processed by road generation algorithms such as [Rot08] to estimate the quality of roads.

With location being a primary context information, the importance of position sensors exceeds that of other sensors. Besides the direct usage of location data to derive context information, location data is often used to map other types of context information and measurements to the location where it was acquired.

Current smartphones often have two integrated cameras. One high-resolution camera at the back for taking pictures, and a low-resolution camera at the front, mainly intended for video telephony. However, several applications have been introduced that process this visual data, and derive context information from it. For instance, EyePhone [MWC10] tracks the eye movement as a means for controlling the smartphone. Besides using a phone's camera for novel human-machine interaction schemes, visual data can also be used to generate 3D models of objects or of the environment. While [VG06] is a web-based approach that processes images uploaded by users, ProFORMA [PRD09] allows for on-line 3D model generation in a stationary environment. In contrast, Ishikawa et al. [ITK+09] present a 3D indoor modeler that generates models semi-manually. Beyond the presented examples, visual data contains a variety of context information that can be extracted with computer vision techniques.

Probably the most ubiquitous sensor in mobile phones is the microphone. In addition to its use for telephony, a microphone can capture characteristics of its surrounding that provide a multitude of context information. In contrast to visual sensors, this does not require the users to take the phone out of their pockets. Early work [SMR06] relied on infrastructure-based processing of sound samples to recognize specific context information. In contrast, SoundSense [LPL+09b] is a general purpose sound sensor that runs on a mobile phone. It enables recognition of sound types such as music or voice, and specific sound events relevant to a user. A more specialized scope have [ACRC09] and [LBP+11]. The former uses fingerprinting techniques to recognize logical locations such as streets or pubs. The latter focuses on the detection of individual speakers.

NoiseTube [MSN+09] and NoiseSpy [Kan10] are two projects that measure the noise level of the environment. This basic context information is of public interest and, by adding these noise measurements to a global noise-level map of urban areas, the noise pollution characteristic of a city can be assessed.

Bluetooth and Near Field Communication (NFC) are wireless communication technologies supported by mobile devices (cf. Section 2.4). They can be used to detect mobile objects in their proximity. For instance, [FgSPO08] uses Bluetooth to detect meetings and interactions of people in urban areas. In addition, these communication technologies allow to easily connect a mobile phone to arbitrary Bluetooth- or NFC-enabled sensors. Thus, these technologies can extend the sensing capabilities of mobile phones by far. First, they allow to sense parameters of the environment otherwise not support by the mobile phones directly. For instance, a temperature sensor included in a mobile phone senses the temperature of the pocket rather than the temperature of the environment. Second, these technologies allow mobile phones to serve as communication relay and collect measurements from external sensor devices. In addition to Bluetooth and NFC, further communication standards such as ZigBee or long-range RFID might be supported by future phones, as well. It has been shown that RFID-enabled mobile phones support mobile lost-and-found applications [GBM08].

Whether external sensors will be connected to mobile phones or additional sensors will be integrated into mobile phones, a wide range of sensors can be expected to be accessible for future public sensing systems. For instance, air quality monitoring is an active topic and the focus of several projects. The N-SMARTS project [HBPW08] uses external air pollution sensors and integrated pollution sensors for air quality monitoring. The OpenSense system [ASC+10] proposes a utility-driven approach for air pollution monitoring that limits sensing to application requirements. At least for reagent-based sensors that rely on chemical reactions to detect specific compounds such as carbon monoxide or nitrogen oxide, the lifetime of a sensor depends on the consumption of its reagent [OD08, Dia04]. The more a sensor is used to detect a compound the faster its remaining lifetime shrinks, which suggest that unnecessary sensing should be avoided.

Other, more exotic sensors that might be a part of future smartphones are heart rate sensors [PKG+09] and neural signal sensors [CCH+10]. While the former relies on earphones to discretely measure the heart rate of a person, the latter uses a headset with integrated wireless electroencephalography (EEG) sensors to monitor neural signals as a hands-free means to control the mobile phone. For an overview of sensor classes and the type of phenomena that can be monitored see [Whi87].

Although individual sensors can be the source for a wide range of context data, the combination of multiple sensors might even provide additional information or more accurate information that could not be derived from single sensors. For instance, [RMB+10] combines GPS and accelerometer data to determine the transportation mode of a mobile phone.

A big challenge in the scenario of sensing with mobile phones is the calibration of the sensors, since these are often shipped uncalibrated [MLCOS08]. However, without calibration, sensors might produce misleading measurements. Beyond the error inherent to the sensor hardware, the phenomenon of sensor drift varies the error over time. Since manual calibration does not scale, [Hon07] and [MLCOS08] propose mechanisms for the automated calibration of sensors that exploit the fact that different sensors at the same location and the same time should report the same measurement.

2.2.2 Discussion of Participation Incentives

It is widely believed that large-scale ubiquitous sensor networks of mobile phones will be available for public sensing applications, although individual users remain in control of the devices. Several aspects support this prediction. First, application stores such as the

Apple App Store[4] or the Android Market[5] promise to quickly distribute sensing applications and attract a large number of participants [LML+10].

Moreover, the information provided by the sensors in mobile phones is of public and/or personal interest. For instance, user who use their mobile phone to monitor their mode of transportation based on accelerometer data may use this information to automatically update their social networking profiles. At the same time, the accelerometer data can be analyzed to detect potholes on roads. This example shows that users might provide information that is of interest to the public with only little additional overhead compared to the default use of their mobile phones. Moreover, the cost for data acquisition can be shared among a large number of applications that benefit from the same basic measurements.

In addition, a trend towards community based projects that aim for collaborative data acquisition and modeling can be identified. For instance, OpenStreetMap [HW08] aims for creating a free detailed world-map. Awekas [Awe11] aims for the collaborative creation of weather maps. In essence, this trend shows that a large number of people are willing to contribute their resources and participate in projects aiming for the good of the public.

As shown by CrowdSearch [YKG10], micro payments can be used as incentives for people to participate in crowdsourcing activities. CrowdSearch pays small amounts of money to people for validating image search results. It can be assumed that similar incentives would also be feasible for public sensing systems. Clearly, there are several scenarios with monetary and non-monetary incentives that support a wide participation of people in public sensing activities. However, mobile phones are battery powered and their primary function is communication rather than sensing. Thus, a low burden should be associated with public sensing.

2.2.3 Characteristics of Public Sensing Systems

The network of instrumented mobile phones is subject to several dynamics that directly correlate with the movement of people, i.e, the carriers of the mobile phones. First, the node density varies over time and space. For instance, at night, usually less people move along the streets compared to daytime. Beyond these macroscopic variations, movement of people is also subject to microscopic effects such as group movement or correlations with public transportation systems.

The second dynamic property of public sensing systems is the node movement itself. Movement of phone carriers ranges from highly structured patterns, for instance, road traffic, to less structured patterns of pedestrian mobility. On the one hand, this mobility leads to a potentially very high spatial and temporal coverage of urban areas [LBD+05]. On the other hand, since mobility of people is self-determined, no guarantees on the sensor

[4] Apple App Store – `http://itunes.apple.com/en/genre/ios/id36`

[5] Android Market – `https://market.android.com/`

availability and the quality of sensing can be given. Two main reasons prevent control of user mobility. Although participatory sensing approaches might achieve limited control over the node mobility by directing the people to specific areas of interest for sensing, large-scale control of user mobility is not feasible. Moreover, control over user mobility is also not aimed for to limit the burden for participation.

Directly related to the node mobility is also the huge diversity of the mobile phones in terms of sensing capabilities. A phone fits different sensing goals depending on parameters of its context like integrated sensors, location, and power level. Therefore, goal-based sensing requires knowledge about the state of the available sensing resources. For instance, temperature sensing of the environment only works if the phone is outside the user's pocket. While such context might be resolved by the intelligence of a user in case of participatory sensing, this is much more challenging in case of opportunistic sensing. The general challenge of monitoring the state of the sensor network and its resources is also stressed by [FH11].

2.2.4 Mobile Phone Sensing Systems

Over the last few years various prototypical mobile phone sensing systems have been developed. In this section, some of those systems are briefly introduced.

SenseWeb [KNLZ07] is a system for managing sensors and the data streams they produce. It allows for selecting suitable sensing resources that are contributed by entities all over the world. Several components form the architecture of SenseWeb. The central coordinator maps sensing requests to the sensors available to the system. Heterogeneous sensors connect to the system through sensor gateways that provide a uniform interface. So-called data transformers allow to easily extend the system with data processing capabilities for processing the data before it is presented to the user.

Similarly, Partisans [PRS+06] proposes a web service architecture as a platform to inject sensor data into applications. One focus of Partisans is selective sharing, i.e., automated control of the data that is transmitted from a sensor to an application. For instance, a node that provides a measurement might only be willing to share it with coarse-grained information about position and time of its acquisition. Moreover, discovery of sensor streams is proposed similar to a publish and subscribe mechanism.

CenceMe [MLEC07, MML+08] is one of the first personal sensing systems to use the sensors in mobile phones to infer user context, such as activity or mood, and share this information through social networks. In addition to position data and Bluetooth discovery information, CenceMe analyzes accelerometer and audio data. Based on this data it determines activities, detects whether a user is in a conversation, or derives social context such as proximity to buddies. This data is uploaded to an infrastructure node from where it is injected into social networks. The work of CenceMe is extended by VibN [MPL+11]. It focuses on analyzing how people and communities interact, and how these interactions relate to locations they inhabit.

MyExperience [FCC+07] extends the idea of using mobile phones to passively monitor context information for a system that allows for active user feedback collection. The main focus of this system is on collecting information about the usage of mobile phones.

BeTelGeuse [KLNA09] is a system to gather and process situational data on mobile phones. The main focus is on inferring high-level information from low-level sensor data. It can be easily extended with components that integrate external sensors or processing capabilities.

Riva and Borcea [RB07] propose the term Urbanet to describe the mixture of sensor networks and mobile ad-hoc networks that cover large urban areas and support people-centric sensing applications. They propose three types of interfaces for such networks to ease distributed programming and specification of sensing tasks. First, declarative programming that allows to access sensor data through an SQL (Structured Query Language) interface. Second, an imperative programming interface that allows to access the resources of an Urbanet as a virtual namespace. Finally, an interface that allows to specify services that migrate from node to node and perform sensing tasks.

Systems with a more specific focus are the following: UbiFit Garden [CMT+08] is a personal sensing system for healthcare support. On the one hand, this system monitors the physical activities of people. On the other hand it tries to encourage people to follow a planned activity schedule. A system with an urban scale application is GreenGPS [GPA+10]. This navigation service relies on fuel-efficient maps that are built from fuel consumption measurements of vehicles. The authors show that their navigation service allows for fuel savings compared to navigating along shortest or fastest routes. The goal of PEIR [MRS+09] is to provide personal environmental impact reports to individuals. Based on GPS data from mobile phones, PEIR determines transportation modes in a centralized way. It combines this data with other web-based context sources such as traffic or weather data to evaluate the environmental impact models.

2.3 Positioning and Localization

Recent years of development have brought a wide range of positioning systems. Probably most widely used is the Navigational Satellite Timing and Ranging - Global Positioning System (NAVSTAR GPS) or short GPS. Satellites continuously transmit signals that encode their position and the current time. Passive GPS receivers compute their own position based on the signal runtimes through trilateration. The accuracy of positioning depends mainly on the number of received satellite signals by a GPS receiver and, thus, satellite constellations. The uncertainty of GPS is often modeled as a normal distribution [Gur91].

Similar satellite navigation systems are the Russian GLONASS system and the European Galileo system. Since the satellite signals cannot reliably penetrate buildings, they are considered outdoor positioning systems.

The field of indoor positioning systems (IPS) is much more heterogeneous, and a lot of research has been done to investigate a wide range of different mechanisms for localization. Instead of reviewing all the research done in this field, focus is put on widely used techniques and novel mechanisms that benefit from the development of mobile phones and sensor technology. For a detailed survey of location systems see [HB01].

Today, WiFi-based or cellular-based localization is supported by a large number of mobile phones. For instance, the current navigation software Google Maps 6.0[6] uses signal strength from WiFi-routers and cellphone towers to triangulate the location of a phone. In some cases, this system achieves an accuracy that allows coarse-grained indoor positioning.

Sensewhere[7] follows a similar approach. To keep its database of location reference points up-to-date, it continuously corrects and updates its database with crowdsourced data from the devices that use the system.

A hybrid between WiFi- and Bluetooth-based positioning is WBroximity [ZCM+10]. It combines both technologies to achieve better positioning accuracy than simple WiFi-based systems. However, it does not require a dedicated beacon infrastructure. It collects the fingerprints that are needed to build a database for the positioning using participatory sensing similar to the Sensewhere approach.

The approach proposed by Parnandi et al. [PLV+10] does not require any infrastructure support. Instead, it uses accelerometer data from mobile phones to detect a user moving up or down stairs and elevators in order to provide floor-level localization in buildings. In contrast to the approaches that rely on an infrastructure, this approach is much less accurate.

Another approach that does not require any infrastructure is [LLC07]. It uses machine learning techniques to classify sensed data and identify known location footprints. Such known locations are referred to as ambient beacon locations. Nodes that are currently at such a location act as a beacon for this location, and allow other, more distant, nodes to perform beacon-based localization.

Recently, CSR, a manufacturer of satellite navigation receivers, has introduced the SiRFstarV architecture and the SiRFusion platform for indoor and outdoor navigation of mobile devices [CSR11]. The architecture aims for integrating sensors available in mobile devices, such as accelerometers, to improve satellite-based navigation. Besides supporting multiple satellite navigation systems, it integrates positioning based on WiFi and cellular signals. Moreover, it integrates data provided by accelerometers, gyroscopes, and com-

[6] Google Maps 6.0 - `http://www.google.com/mobile/maps/`

[7] Sensewhere - `http://www.satsis.com/`

passes. Especially indoors, this platform can improve positioning compared to systems that rely on satellite-based navigation only.

Positioning is subject to measuring errors, positioning tolerances, and approximations. A large number of specific approaches for specifying positioning uncertainties have been proposed [LWG+09]. In order to deal with the heterogeneity of positioning systems and their corresponding uncertainty models, we propose a generic uncertainty model that integrates the different approaches. We previously published a detailed overview of the model and the resulting interface for accessing spatial data [LWG+09]. In the following only a short overview is presented.

Three basic classes of specific uncertainty models are known. The first class uses probability density functions (pdf) to model the accuracy of position data, they are referred to as pdf-based models. For instance, the accuracy of GPS-positions is modeled as a pdf. The second class, are models which describe position data using shapes. For each shape that describes a spatial region, a probability value is associated which reflects the likeliness of the real position to be in the respective region. In contrast to pdf-based models, no probability density function is defined for the individual shapes. For instance, position information from beacon-based systems is modeled as shapes. Finally, uncertainty of positions is sometimes not modeled at all. In these cases, the position reported by the positioning system is modeled as a point.

All of these models enable mapping from geometric shapes to cumulative probability functions. Since this mapping may be defined only partially, it is referred to as partial spatial distribution function (psdf). Given any geometric shape, an upper bound p_{upper} and a lower bound p_{lower} for the probability that the actual position is within this shape can be determined. In case of a completely defined function, upper and lower bound are always equal.

Based on this model, an uncertainty-aware query interface is devised. It allows for accessing and processing position information from heterogeneous sources. The interface extends the three major types of spatial queries: position query, range query, and nearest-neighbor query.

In addition to the identifier of the object whose position is queried, the position query takes a parameter $p \in [0,1]$. The query deterministically returns a minimal area that contains the queried object with a probability of at least p. For instance, in case of a position that is normal distributed with $N(c, \sigma^2)$, the query for an object with $p = 0.75$ returns a circular area around point c with a radius of about $1.18 \cdot \sigma$. Depending on the tolerance of an application, this query type allows to retrieve the area that is likely to include the position of an object.

The range query is based on inside queries. An object is *inside* a range, i.e., a geometric shape, if the lower bound p_{lower} for the probability that the object is inside the range is higher than a given threshold p_{true}. If the upper bound p_{upper} for the probability of the object being outside the range is larger than a given threshold p_{false}, it is considered *outside*. Otherwise, the object may be in the range, and the query returns *maybe*. The inside query is easily extended to a range query.

The distance query returns bounds for the distance between two objects. Based on the minimum probabilities for the object positions, the minimum and the maximum distance between the resulting shapes is computed. The distance query can also be extended to a nearest-neighbor query, similar to the range query.

2.4 Wireless Networks

Mobile devices such as smartphones are equipped with a variety of wireless radio interfaces. First, they connect to cellular networks that span most inhabited areas all over the world. Second, most mobile phones that are shipped today are equipped with a WiFi radio, which enables local connectivity. Finally, various short range radios are integrated in current mobile phones to connect to external resources.

2.4.1 Cellular Networks

Cellular networks, also called mobile telephony networks, are infrastructure-based wireless wide area networks (WWAN). Areas, which are called cells, are served by fixed nodes known as base stations. The base stations are interconnected by a core network. Mobile nodes, i.e., mobile phones, directly connect to the base stations, and are transparently handed over to neighboring base stations as they move from cell to cell.

Starting several decades ago, the first generation (1G) of these networks was based on analog technology and had only few users. The first digital cellular networks are known as second generation (2G) networks. Today, the Global System for Mobile Communications (GSM), developed in the 1980s by the European Telecommunications Standards Institute (ETSI), is the most widely implemented standard of a 2G cellular network with billions of subscribers. Beyond its initial goal to serve for voice telephony, the GSM standard was extended by the General Packet Radio Services (GPRS) to support packet data transport. Growing demands for traffic bandwidth have afterwards led to EDGE (Enhanced Data Rates for GSM Evolution).

The third generation networks (3G), most prominently represented by the Universal Mobile Telecommunications System (UMTS) standard, aim for increased network bandwidth. UMTS networks are based on GSM. While the maximum downlink data rate per user in a UMTS network is 384 kbit/s, it is in the range of several MBit/s for the extended High-Speed Downlink Packet Access (HSDPA) standard.

The development of fourth generation (4G) cellular networks often referred to as Long Term Evolution (LTE) aims at further increasing data rates and reducing latencies [ADF+09]. Moreover, 4G networks are developed as all-IP networks, i.e., telephony and data services are provided on top of an IP network.

Despite the constant evolution of cellular network technology, its capacity growth is still outpaced by the tremendous growth of mobile data traffic [Han09]. Therefore, offloading cellular traffic through WLAN or WMAN networks is getting into focus of mobile operators and research [BMV10, LRL+10, RLBCE11].

WiMAX (Worldwide Interoperability for Microwave Access) is another technology that implements the 4G network specification [Ahm09]. Beyond its use as an alternative to wired broadband access technologies, it also supports mobile applications.

2.4.2 WiFi Networks

A WiFi radio interface is integrated in most mobile phones that are shipped today. It operates in the industrial, scientific and medical (ISM) bands, which do not require license fees. The Institute of Electrical and Electronics Engineers (IEEE) standards committee defines the IEEE 802.11 standards for implementing several modes of WLANs. The initial IEEE 802.11 standard was released in 1997. Several evolutions have led to the current IEEE 802.11n specification that achieves transmission rates of several hundred MBit/s.

A basic type of wireless network is based on an access point infrastructure. A node in access point mode connects WiFi-enabled nodes to, for instance, wired networks. Moreover, mobile nodes that can act as access points allow for discovering and direct communication between mobile nodes. The latter mode of operation is specified by the WiFi Direct standard, for instance.

Another mode of operation is ad-hoc communication. Especially in case of node mobility, the set of direct communication neighbors, i.e., the nodes in communication range, changes dynamically. Since communication links are established dynamically, the topology of the physical network changes dynamically, as well. In general, the communication between two nodes is multi-hop, i.e, it involves several intermediate nodes. Wireless communication is subject to much higher error rates compared to wired communication. Moreover, collision detection and collision avoidance mechanisms are challenging, due to exposed terminal and hidden terminal topology configurations. In the literature, mobile ad-hoc networks (MANETs) have been actively studied. Several mobility models [CBD02] have been proposed to model the movement of the nodes in such a network.

Sometimes considered a more general network type than MANETs are the wireless mesh networks (WMNs). In contrast to MANETs, WMNs are hierarchically organized. A set of stationary nodes, the mesh routers, form a relatively stable wireless core network. The topology of the core network is often planned and offers redundant paths. The mesh routers are assumed to be resource rich, i.e., they have large storage, powerful processing capability, and no energy constraints. Special mesh routers may act as gateways to other networks such as the Internet. Mobile nodes, the mesh clients, connect to a mesh router either directly or through intermediate mesh clients. The IEEE 802.11s standard defines a wireless mesh network based on WiFi technology.

As this short overview shows, technologies similar to WiFi can serve two main purposes. First, they can act as local access networks to an infrastructure such as the Internet. Although WiFi-based access networks do not achieve a coverage of urban areas that is comparable to that of cellular networks, it is widely believed that coverage will expand drastically in the future. For instance, Google has deployed a mesh network in Mountain View [ACVS10]. The second purpose of such technologies is spontaneous communication of nodes in close proximity. Thus, such technologies can provide communication with low delay on a local scope independent from an infrastructure.

2.4.3 Short Range Wireless Communication

Several short range wireless radio technologies can be used for accessing external resources. Most widely distributed in current mobile phones are Bluetooth radios. Bluetooth was initially developed as a cable replacement technology that operates in the ISM bands. As a wireless personal area network (WPAN), it offers lower data rates and lower communication range compared to WLAN technologies, however, much more energy efficient.

After its development in the 1940s [Sto48], RFID was mainly used in industry to identify objects. Two basic components are part of an RFID system: RFID reader and transponder. In essence, to identify an object, the reader creates an electromagnetic field at a specific frequency. The field powers transponders which can then transmit stored information, i.e, their ID. This mode of operation is often referred to as passive mode, since it does not require transponders to have a power source. Active RFID transponders rely on a local power source to communicate with the reader.

More recently, RFID readers have been integrated into mobile phones [GBM08, PRR08]. In this dissertation, the EPCglobal UHF Class 1 Generation 2 standard [EPC05] for RFID systems is considered. It operates in the UHF band and achieves a communication range of several meters [AD07, BW08]. Beyond simple object identification purpose, transponders can be integrated with sensors [DVS$^+$06]. The functionality of these transponders is reduced to a minimum, namely sensing and 1-hop communication based on passive long-range RFID technology. Without an expensive and complex radio interface and without powerful processing capability these transponders can be produced much cheaper compared to traditional sensor nodes. Even simple battery-less transponders are conceivable that sample environmental phenomena of their surrounding. Using such sensor nodes, disadvantages of WSNs such as cost and battery-depletion can be greatly reduced. Without the need to form a complex multi-hop sensor network, these sensors need to be placed only at points of interest. The transmission of measurements is then performed by mobile devices with an integrated RFID reader, as mobile “relays”, and the readily available mobile communication infrastructure.

A technology, more widely integrated into mobile phones is NFC, which is based on HF (high frequency) RFID technology. Thus, NFC technology is intended for very short range communication over a distance of only a few centimeters. In addition to Bluetooth, NFC, and RFID, further communication standards, such as ZigBee, might be supported by future phones, as well. While WiFi is the de facto standard for WLANs, WPAN technologies are much more heterogeneous.

3 System Overview

This chapter presents an overview of the public sensing system that is devised in this thesis. First, the system requirements are described, based on the discussions of the previous chapters, in Section 3.1. Second, the system model is defined in Section 3.2. Then, in Section 3.3, the layered architecture of this system is introduced and discussed. The relation of its individual components is explained. Finally, an overlay network is sketched in Section 3.4 as an interface to this system.

3.1 System Requirements

Based on the discussions in the previous chapters, this section describes the key requirements for the coordinated acquisition of sensor data, which is considered in this thesis.

unobtrusive operation: In order to benefit from the large number of sensors in urban areas and their potential, sensing needs to be opportunistic and involve or hinder people as little as possible. In essence, this means that manual tasks for users should be avoided. Especially, this means that the movement of people is not controlled.

energy efficiency: Related to the previous requirement, the efficiency of public sensing in terms of energy consumption is crucial for the participation of a large number of people. Inefficient sensing would drain the batteries of the mobile devices and, therefore, interfere with the primary operation of mobile phones: mobile communication.

communication efficiency: Large-scale sensing in urban areas at high rates leads to an enormous amount of measurements that needs to be transmitted to the infrastructure. Therefore, it is essential to prevent phones from transmitting unnecessary redundant measurements that would impact user experience.

quality-aware sensing: Quality of public sensing mainly depends on the coverage of sensing. Since people's mobility is not controlled, sensing is opportunistically done where people move. Thus, the resulting coverage highly depends on the mobility of people. Quality-awareness serves two goals: it allows to reduce sensing activities to the requested level, and it allows to quantify the resulting data quality.

hybrid system structure: Urban environments are characterized by the diversity of communication infrastructures. Therefore, a public sensing system needs to cope with the specifics of these technologies.

decoupling of applications and phones: Applications are interested in data related to specific geographic locations rather than data from specific mobile phones. Therefore, applications need to be decoupled from physical data sources using a data-centric abstraction.

node mobility: On the one hand, the inherent mobility of people in urban areas leads to a large sensor coverage. On the other hand, it means that sensing activities need to be monitored and adapted constantly.

node density: Similar to the previous requirement, public sensing needs to cope with and adapt to large variations in temporal and spatial node density. A uniform level of sensing performance should be provided independently from node density. Since no specific performance can be guaranteed in an opportunistic system, loss of performance compared to the inherent bound through node movement characteristics should be avoided.

scalability: Millions of mobile phones cover large areas of our globe. Therefore, scalability of public sensing is required with respect to the number of nodes and the size of the service area.

3.2 System Model

A *Public Sensing Server* (PSS) coordinates a network of mobile nodes to cooperatively collect sensor data from external sensors or using their integrated sensors (cf. Figure 3.1). Connectivity is given either through cellular networks or through a wireless mesh network. In the following, the individual components that comprise the system are described, and the assumptions are given.

Mobile Nodes Mobile nodes N_i are carried by mobile users moving in the service area. Their movement is uncontrollable. A graph representing a street network defines where nodes may move. The maximum node speed is v_{maxn}. The battery of the nodes is assumed never to deplete.

For ad-hoc inter-node communication, nodes have a WiFi radio interface, e.g., 802.11bg, with transmission range r_{tx}. This interface is used for the distributed coordination of sensing. In addition, nodes can communicate with the PSS in the infrastructure using either

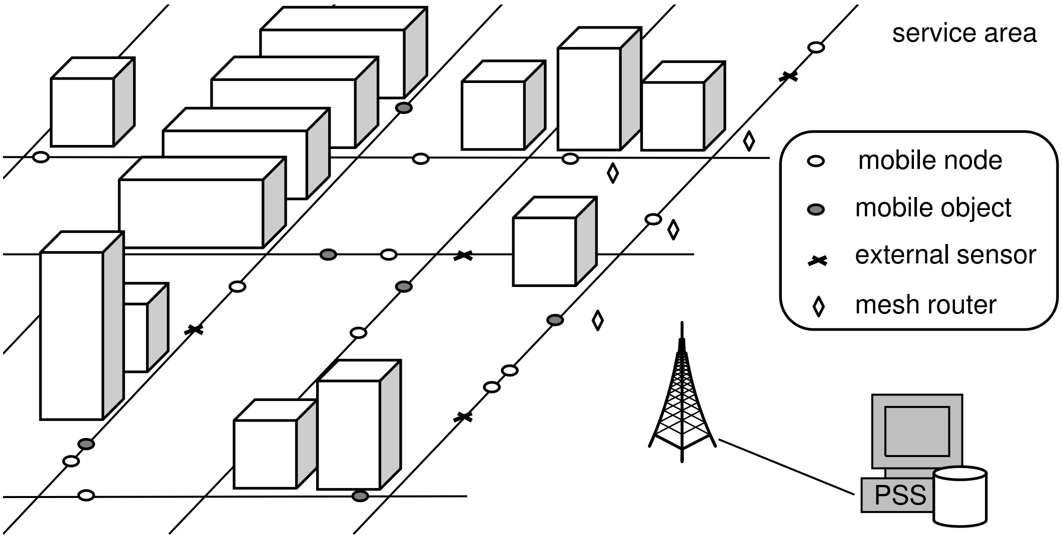

Figure 3.1: Components of the system model

WWAN networks such as the Global System for Mobile Communications (GSM) or Universal Mobile Telecommunications System (UMTS), or WMAN networks such as wireless mesh networks (WMN). Infrastructure connectivity is used for transmitting measurements from the nodes to the PSS, and for the central coordination of sensing.

Nodes are equipped with an outdoor positioning device such as a GPS receiver, which allows for deriving geographic trajectory information and which allows for synchronized node clocks. The positions reported by the positioning device are assumed to be inaccurate. In essence, it is assumed that the actual position is uniformly distributed within a radius r_{accuracy} around the reported position. Nodes can map their geographic position to a symbolic location according to a given symbolic location model or they can directly determine their symbolic location with a symbolic positioning device such as IR-beacons.

Nodes can collect sensor data from internal or external sensors. Internal physical sensors can capture environmental phenomena such as noise level and air pollution or they can detect objects in the proximity based on technologies such as RFID, Bluetooth, or ZigBee. Such proximity sensors can identify mobile objects with a specific probability p_{detect} if the object is within the sensor range r_{range}. The detection is based on unique object identifiers that are reported by the proximity sensors.

Based on wireless communication interfaces, mobile nodes can collect sensor data from external sensors. For instance, with passive RFID technology [BW08], an RFID reader integrated in the mobile node powers an external sensor so that it can transmit its current measurement. This reduces the required energy of an external sensor to a minimum, however, also the sensor range r_{range} is reduced to a maximum of about 5 m assuming UHF RFID technology. δ_{read} denotes the time it takes to acquire a measurement. An RFID reader as wireless sensor interface is prone to collisions [HES06], i.e., sensing is not successful if multiple readers try to read an external sensor concurrently.

The sensor range is denoted by r_{range}. This range is either, e.g., in case of a RFID reader, derived from a data sheet or, e.g., in case of an air pollution sensor, specified by an application according to its spatial resolution requirements. The output of a sensor is denoted as measurement M. It is associated with the time of its acquisition $time(M)$, and the position of its acquisition $pos(M)$. In order to assign measurements to positions on road segments, mobile devices rely on map matching techniques such as [CJNP04]. Each of these sensors reports a measurement instantly when accessed. Note that this assumption is valid for a wide range of sensors, e.g., thermometers or RFID readers. Sensing, the process of acquiring a measurement, is also referred to as reading.

Mobile Objects Similarly to mobile nodes, mobile objects O move self-determined on the street network. Their speed is limited to v_{maxo}. Each object has a unique identifier, for instance, a Bluetooth MAC address or an ID of an RFID tag.

External Sensors External sensors E are based on RFID technology that provide dynamic sensor values. The energy for sensing operations [DVS^{+}06] is either provided by the mobile node through the electromagnetic field (passive RFID) of its wireless sensor interface during sensing or by an extra battery (semi-active RFID). The sensors are placed stationary along the roads of the service area within sensor range of passing by mobile nodes. Each sensor has a unique identifier.

Street Network The street network is modeled as a graph G. Edges represent segments S of a street. Their weight denotes the length of the respective segment. Nodes and objects enter or leave the street network or transit to a different segment only at a vertex. Such vertices may be junctions, where objects transit to another segment, or doorways, were objects enter or leave a building.

Symbolic Location Model A symbolic location model (SLM), similarly defined in [BD05], consists of a set of symbolic locations with unique identifiers. An SLM defines two relations. First, the inclusion relation $\subset$ defines the *Location Hierarchy Tree* (LHT). The notation $L \subset P$ describes that the geographic area of location P covers the geographic area of location L. Location P is called a parent location of L if $L \subset P$ and there is no location M that satisfies $L \subset M \subset P$. Moreover, the sub-location L is referred to as a child location. Each location L is assigned a hierarchy level $level(L)$ according to the depth of the respective location in the LHT. The transitive extension of parent and child relations are referred to as k-parent for ancestors and k-child for decendants respectively, where k refers to the distance of the levels in the LHT. The non-symmetrical hierarchical distance

d_{hier} between two symbolic locations S and T is defined as the number of hierarchy levels that are between the location T and the smallest common ancestor P of location T and location S.

$$d_{\text{hier}}(S,T) = level(T) - level(P) \tag{3.1}$$

The second relation is defined by a graph that models the neighborhood relation of the leaves in the LHT. This information is derived from a floor plan, where adjoining locations are defined to be neighbors. On this *Location Neighbor Graph* (LNG) the geographic distance d_{geo} between two locations is defined as the length of the shortest path between them. All edges of the LNG have a weight of one.

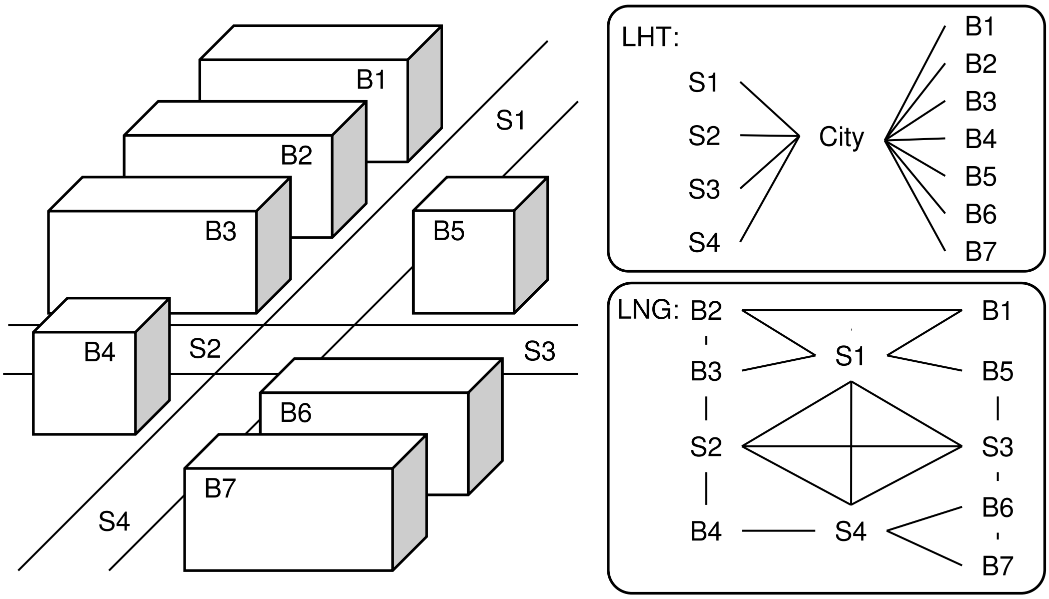

Figure 3.2: Symbolic location model (SLM)

Figure 3.2 shows a typical city region and the corresponding location model. Location City is the parent location of the buildings B1 to B7 and the streets S1 to S4. City and all of its sub-locations, compose the set of locations. Adjoining locations, i.e, connected locations in the LNG, are for instance building B2 and street S1.

Wireless Mesh Network The wireless mesh network (WMN) consists of stationary mesh routers and mobile nodes acting as mesh clients. The topological distance d_{topo} between two nodes is defined as the number of hops between them. Both types of mesh nodes are equipped with a WLAN interface for inter-node communication, e.g., a 802.11bg interface. The transmission range of this interface is denoted by r_{tx}. The mesh network covers defined parts of the service area and provides a communication infrastructure for the communication between mobile nodes and PSS. Mesh networks may only partially cover the service area of the PSS. The terms mobile node and mesh client are used interchangeably.

Public Sensing Server A *Public Sensing Server* (PSS) is either an infrastructure-based node or a mesh router responsible for managing the measurements of the mobile nodes in a specific geographic region (service area). In addition, the PSS stores the graph representing the street network and the symbolic location model. Moreover, it stores profiles of the virtual sensors in its service area, and it is the central coordinator of sensing operations.

3.3 System Architecture

Figure 3.3 shows the conceptual architecture of the public sensing system. Each component of this figure and their relation to the requirements will be sketched in this section. A detailed description of each component will be introduced in the following chapters of this dissertation.

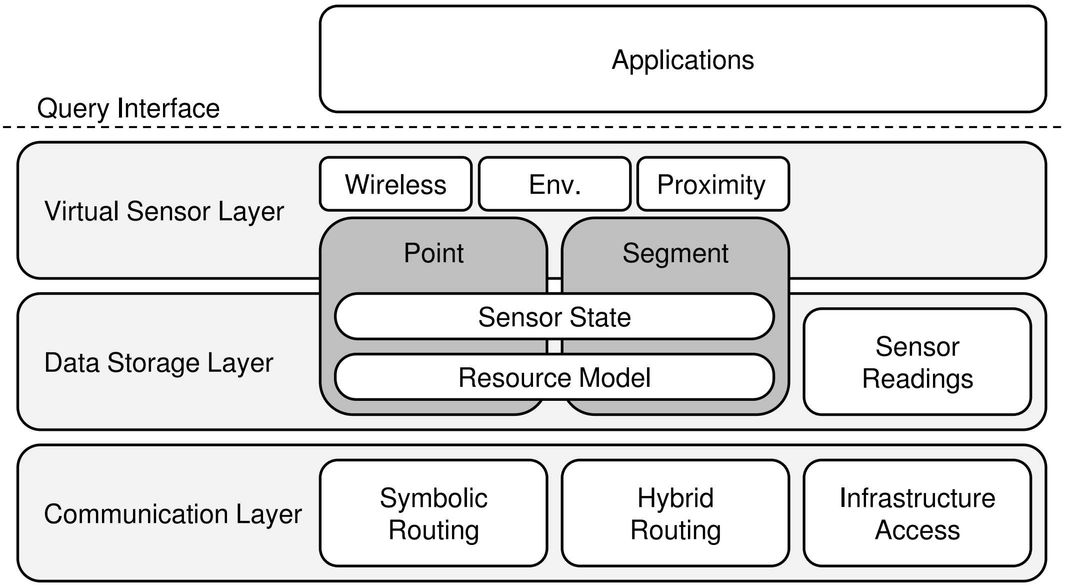

Figure 3.3: Conceptual architecture of the public sensing system

The *Communication Layer* contains fundamental routing mechanisms for the scalable communication between the infrastructure-based PSS and mobile nodes. Beyond the abstraction from communication technologies (Infrastructure Access), it mainly provides communication primitives for addressing nodes at specific symbolic locations (Symbolic Routing and Hybrid Routing). This functionality is the basis for the algorithms in the data storage layer. It mainly addresses the requirement to deal with hybrid system structures, and the location-based communication allows for decoupling applications from mobile nodes. Main parts of the mechanisms of the communication layer were previously published. Symbolic routing was published in [WDR10b], while an extension of these mechanisms supporting hybrid networks was proposed in [DWR10].

The *Data Storage Layer* serves two main goals. First, it provides mechanisms to manage and retrieve measurements (Sensor Readings). This goal is mainly fulfilled by relying on the mechanisms provided by the communication layer. Second, it provides mechanisms to distribute and manage control information needed to coordinate sensing. Besides the basic management of virtual sensor configurations, this layer provides mechanisms to monitor the current state of virtual sensors (Sensor State) and resources such as mobile phones (Resource Model). For efficiency reasons, these mechanisms are tightly coupled with mechanisms of the virtual sensor layer.

The *Virtual Sensor Layer* provides a data-centric abstraction for the physical network of mobile nodes and the sensor data they provide. It includes mechanisms to coordinate sensing in order to efficiently achieve defined sensing goals. Two basic types of virtual sensors are provided: Point sensors and Segment sensors. Point sensors sample environmental phenomena at a geographic position. In contrast, segment sensors sample environmental phenomena on one-dimensional segments along road segments. The virtual sensor layer includes mechanisms that use integrated sensors in mobile devices and external sensors. Moreover, it allows monitoring of environmental phenomena and detection of mobile objects. Virtual sensors are defined more formally in Section 5.1. Main parts of the algorithms for coordinating sensing of mobile nodes and the corresponding mechanisms of the data storage layer were previously published in [WDR09, WDR10a, WWDR11].

3.3.1 Query Interface

The query interface serves for accessing sensor data using a data-centric approach similar to Riva and Borcea [RB07]. In addition to specifying the required sensor data, the query interface allows to specify the requested quality of the data. Table 3.1 lists the parameters of such a query Q. First, $\mathfrak{S}$ denotes the spatial region for which sensor data is requested. Second, $\mathfrak{T}$ denotes the type of the sensor data to be acquired. Third, $\mathfrak{F}$ denotes the set of filters that are applied to the measurements. For instance, this allows for filtering all measurements whose value exceeds a certain threshold. The period $\mathfrak{P}$ of the sampling rate specifies the time between sampling phases. The duration $\mathfrak{D}$ denotes the duration of such a monitoring phase. Finally, $\mathfrak{Q}$ specifies the quality requirements. Note that these requirements depend on the type of the requested sensor data.

A PSS maps the queries it receives to the virtual sensors in its service area. If required, the PSS sets up new virtual sensors based on the parameters of the received queries. By

Table 3.1: Query interface

Query: $Q = (\mathfrak{S}, \mathfrak{T}, \mathfrak{F}, \mathfrak{P}, \mathfrak{D}, \mathfrak{Q})$
$\mathfrak{S}$: Spatial region to be monitored
$\mathfrak{T}$: Type of sensor data
$\mathfrak{F}$: Filters on sensor data
$\mathfrak{P}$: Period of sampling rate
$\mathfrak{D}$: Duration of acquisition
$\mathfrak{Q}$: Quality requirements

preventing applications to directly set up and configure virtual sensors, PSS operators have the possibility to apply sophisticated mapping policies.

3.4 PSS Overlay

As stated in the system model (cf. Section 3.2), a PSS is an entity that manages and coordinates sensor data acquisition for a specific spatial region. While the remainder of this dissertation focuses on the management and coordination tasks of such a PSS, this section briefly sketches an inter-PSS network. It offers a federated view on data acquisition and management similar to the federation of context providers proposed in the Nexus project (cf. Section 2.1.1).

Each PSS manages sensing activities within its service area. Thus, management and coordination can be considered independent from other PSS. In essence, each PSS acts as gateway and provides access to the mobile nodes in its service area and their measurements. The *Public Sensing Servers* form a structured overlay network and offer a publish/subscribe service similar to [PRS$^+$06, ALM11, TKK$^+$11]. Using this spatial partitioning approach, scalability is achieved.

Nodes discover the PSS that is assigned to their location by connecting to the PSS overlay and publishing a discovery message that contains their current position. Every PSS subscribes to discovery messages for its service area. The PSS that matches a discovery message then directly connects to the nodes. Using this mechanism, mobile nodes can dynamically connect to the PSS associated with their current location.

Similarly, applications can connect to the overlay and publish their requests for sensor data (cf. Section 3.3.1). Using the overlay and the publish/subscribe service, applications are decoupled from individual PSS, i.e., they can specify the data they require rather than dealing with data providers. The publish/subscribe service routes these requests to those PSS with matching subscriptions. Note that a PSS subscribes for requests for the sensor data it can provide. Based on the set of requests a PSS receives, it determines sensing goals, sets up virtual sensors, and coordinates the nodes in its service area accordingly.

Measurements and corresponding quality information are published by the PSS. The publish/subscribe system routes this data to the subscribers, i.e., the applications. Besides this stream-based processing model, a distributed system for managing sensor data can be easily built on top of the PSS overlay. The peers of such a system subscribe to the sensor data that needs to be managed and provide, for instance, aggregation services in a request-response model.

4 Hybrid Routing in Wireless Mesh Networks

Addressing nodes and information using location information as a key is highly beneficial in a dynamic scenario such as public sensing. It allows for querying information or a mobile node that covers a specific region without the need to know an individual physical entity. Thus, dynamic changes in the region of interest do not require a notification of the remote communication partner. In order to provide users with an intuitive addressing scheme, symbolic location information [BD05], e.g., street names or building numbers, is supported as an addressing concept. Although symbolic and hybrid routing approaches for infrastructure-based systems have been studied [DR08], they cannot be applied to wireless mesh networks (WMNs) that have emerged as a cost-efficient means to build medium-scale networks in urban areas.

This chapter presents a novel routing protocol for WMNs supporting anycast and geocast routing with symbolic and hybrid addressing. A simple approach to implement symbolic routing would be to introduce a location server to which each node sends its current position and which resolves symbolic addresses to node addresses. The latter can be used by a common routing algorithm, such as AODV [PBRD02] to establish a route to the target. However, this simple approach suffers from the introduced indirection, which requires sending possibly frequent position updates to the location server. Therefore, this dissertation presents an approach that integrates location management and routing, in the sense that symbolic location information is managed in a distributed and scalable manner, and nodes are enabled to forward messages based on symbolic location information.

The basic idea is to proactively build a routing structure that mimics the symbolic location model (SLM). Having such a structure, the SLM can be used for directed routing. Moreover, to cope with the discrete nature of symbolic addresses, which denote areas like streets or buildings rather than point coordinates, special mechanisms to forward messages through such areas are devised.

Main parts of this chapter, especially the symbolic routing mechanisms, were previously published in [WDR10b]. The discussion of the hybrid extension of these mechanisms was published in [DWR10]. The remainder of this chapter is structured as follows. First, an anycast as well as a geocast routing algorithm based on symbolic location models is presented in Section 4.1. This approach is extended in Section 4.2 to support routing based on hybrid location models. Then, it is briefly sketched how these routing mechanisms can be used to build a location-based data storage service in Section 4.3. Afterwards, in Section 4.4, the evaluation results are given. Finally, in Section 5.6, some related work is discussed.

4.1 Symbolic Routing

Symbolic routing is a network service that allows sending a symbolically addressed message to a location representing an area defined by the SLM (symbolic location model). The basic idea of the symbolic routing approach is to proactively build a routing structure that mimics the structure of the symbolic location model. More specifically, routes between locations that are connected by an edge either in the *Location Hierarchy Tree* (LHT) or in the *Location Neighbor Graph* (LNG) are established. Since connectivity of mesh nodes usually correlates with their geographic distance, a geographic routing structure resembling the SLM leads to short network paths. Since the SLM is static in contrast to the physical network topology, using the SLM as directional hints for routing is also beneficial in terms of network overhead. Each node stores a copy of the SLM.

This structure enables forwarding of messages from any location to any other location stepwise through a chain of intermediate locations, similar to an overlay network. Each step possibly involves multiple intermediate mesh nodes as relays. The proposed mechanisms are robust against several challenges resulting from sparse node densities or coarse grained location information. First, empty locations are bridged. Second, partitions within locations, which permit forwarding through that location, are resolved. Finally, the proposed mechanisms are also effective if the directional information provided by the SLM is ambiguous.

The routing structure is discussed in Section 4.1.1. Based on this structure, two routing primitives are introduced. *Symbolic Anycast Routing* (SAR), presented in Section 4.1.3, is a routing primitive that delivers a message to any node at the addressed location. In contrast, *Symbolic Geocast Routing* (SGR), presented in Section 4.1.4, delivers a message to all nodes at the addressed location. Finally, in Section 4.1.2, the mechanisms for building and maintaining the routing structure are discussed.

4.1.1 Routing Structure

This section states how the elements of the SLM are used as a “template” for building the routing structure. In essence, this routing structure is an overlay structure, where communication between neighbors in the overlay possibly involves multiple nodes in the underlay as relays. The details of its maintenance are introduced in Section 4.1.2.

The basic idea of the routing structure is to elect, for each location of the SLM, nodes that organize in a structured overlay. A link in the overlay is established between nodes of neigbor locations in the LNG or in the LHG. Ideally, the structure of the overlay completely resembles the structure of the SLM. However, depending on the characteristics of the underlay, the overlay may not contain nodes for every location of the SLM, and it may not contain links for every neighbor relation in the LNG or the LHT. For instance, locations may be empty, i.e., no node at that location can be elected to participate in the overlay. Section 4.1.3 will discuss the routing mechanism and show how it uses the SLM to compute overlay routes that are used to forward meassages despite of differences between the SLM and the achieved overlay structure.

In the following, the ideal routing structure is explained in detail. For each location of the SLM, a node participates in the overlay. Such nodes are referred to as associated nodes. A node that is associated with location L is referred to as AN_L. The AN_L is located at location L. If no node is located at location L, the overlay does not contain an AN_L.

The overlay consists of two structures. The first structure is the *Node Connectivity Graph* (NCG). Links of the NCG are set-up according to the links of the LNG (cf. Section 3.2), i.e., underlay routes between ANs of direct neighbor locations are maintained.

The second overlay structure is the *Node Hierarchy Graph* (NHG). Links of the NHG are set-up according to the links of the LHT (cf. Section 3.2). An AN of a location maintains a route to the AN of its parent location and to the ANs of its child locations. Note that child locations may be empty, i.e., not represented in the overlay. Moreover, an AN also knows what links its child ANs maintain, i.e., it knows through which child which sub-location can be reached. This information is encoded in the *reachability summaries*. The routes between parent and child ANs form a layered hierarchic structure.

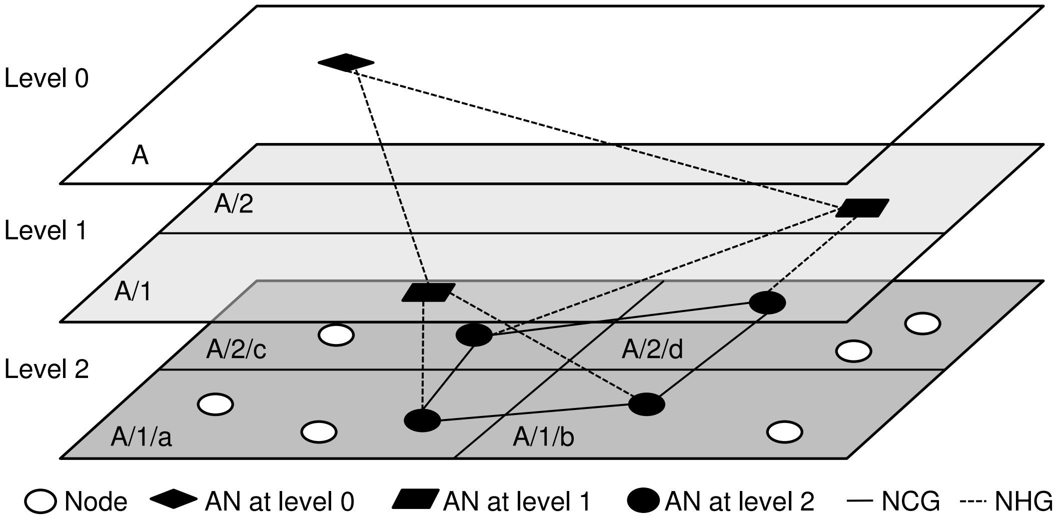

Figure 4.1: Routing structure consisting of NCG and NHG

Figure 4.1 shows a network, where each location of the three level hierarchy has one AN, and the overlay links between these ANs. The ANs on the lowest level form the NCG, while the NHG is formed by all ANs.

Every node knows underlay routes to several ANs that are its entry points to the NHG and NCG structures. First, each node knows at least one route to the AN of its current location. In addition, every node knows a route to the AN for each of its ancestor locations.

Each node manages a routing table where it stores entries for the underlay routes to ANs of the NCG and NHG structures. An entry of the routing table consists of the symbolic location representing the destination, the link layer address of the AN at the destination, the topological distance d_{topo} to this AN, i.e., the number of hops, and the link layer address of the next hop node towards the destination. Routing entries are discarded according to the soft-state principle. The mechanism to fill and refresh these tables are discussed in Section 4.1.2.

4.1.2 Routing Structure Maintenance

The algorithm for maintaining the routing structure as described in Section 4.1.1 consists of three mechanisms. First, a mechanism for electing the ANs to reduce the number of nodes actively participating in the routing structure maintenance. Second, a mechanism to build the *Node Connectivity Graph* (NCG) of ANs associated with neighboring locations and, finally, a mechanism to build the *Node Hierarchy Graph* (NHG). All of these three mechanisms are based on periodic advertisement messages and are described in the following sections.

Election of Associated Nodes

The selection of nodes that form the routing structure is crucial for the effectiveness and efficiency of the routing algorithm. The selection is done using an election mechanism that allows to consider various aspects such as stability or load. In essence, nodes specify the tuning parameter e, i.e., their “eagerness” to be associated with a location. The node with the highest eagerness is elected as AN. The election is based on periodic advertisements. Nodes advertise their eagerness if they do not receive advertisements for their location for more than δ_{adv} from a node with higher or equal eagerness. An AN that receives an advertisement for its associated location, stops sending advertisements for this location if its eagerness is lower than or equal to that of the advertisement. Moreover, an AN stops sending advertisements when it leaves the location to which it is associated. Several criteria can be used to determine or set the eagerness of nodes.

mobility: To keep ANs more stable, stationary nodes increase their eagerness.

load: To reduce their load, and to prevent bottlenecks, nodes decrease their eagerness.

value of location: Nodes that are at a strategic location, e.g., a central location, increase their eagerness to reduce the overall path length in the network.

number of associations: Nodes that are already associated with some locations, increases their eagerness to profit from synergies through combined advertisements, i.e., one node instead of several nodes has to send advertisements.

gateway: Nodes that are directly connected to the infrastructure and, thus, can act as gateway between WMN and the infrastructure, increase their eagerness.

user defined: A user or an application can indicate whether this node should be integrated tightly into the routing structure or not.

Advertisements are sent as periodic broadcast messages with a period of δ_{adv} to all direct topological neighbors and are forwarded with a certain TTL by every node at the advertised location. Outside the advertised location, the advertisements are discarded. The value for δ_{adv} is chosen according to node mobility, which influences the probability for a route to break, and the available bandwidth. To prevent the concurrent sending of advertisements, nodes randomly delay sending advertisements.

The TTL is correlated with the location size and set to the number of hops that are needed for a message to traverse a location. The size of a location is specified by the SLM. As a trade-off between overhead and resilience, the number of ANs of a location can be adjusted. For load distribution, a smaller hop value can be chosen resulting in multiple nodes associated with a single location. Duplicate forwarding is suppressed and multiple advertisements of a single node are combined into a single message.

Depending on the hop limit of an advertisement and depending on partitions within locations, multiple ANs may represent a single location. A node that receives an advertisement adds a route to the sender. Sophisticated schemes can be incorporated to select a route if multiple advertisements are received. However, such schemes that, for instance, prefer stable routes, are beyond the scope of this dissertation. Thus, the shortest route is selected.

To allow for efficient processing, advertisements only include location identifiers instead of location names. For instance, the location /BuildingA/Floor2/Room2.223 can be simply represented by the unique identifier 2578 within the scope of the SLM. Such a mapping can be defined, e.g., by the lexicographical order of the location names of an SLM. Hereby, the size and processing complexity of an advertisement is reduced. It includes a set of location identifiers, a TTL value, an eagerness value, and its sender.

NCG Building

The mechanism to build the NCG structure is based on the forwarding of advertisements to ANs of the direct neighbor locations of the sender. With this mechanism, advertisements of an AN reach ANs of neighboring locations in the LNG.

To prevent flooding in neighbor locations, an optimized forwarding mechanism for advertisements in neighbor locations is introduced. A link between two ANs of neighbor locations L and N can be established if the AN_N receives at least one advertisement from AN_L. The idea of the optimization is to stop forwarding as soon as an advertisement reaches the AN_N. Therefore, direct neighbors of AN_N include the location N in the advertisement before forwarding to signal that they deliver the advertisment to the AN_N. Nodes that are at one of the locations included in an advertisement suppress forwarding assuming that the advertisement already reached AN_N.

NHG Building

The mechanism to build the NHG structure establishes routes between parent and child ANs. When a child AN receives an advertisement of a parent, it replies with its own advertisement on the reverse path. The reply is randomly delayed to reduce the risk of message loss through collisions. The reply also includes information about links to descendants, i.e., the reachability summaries.

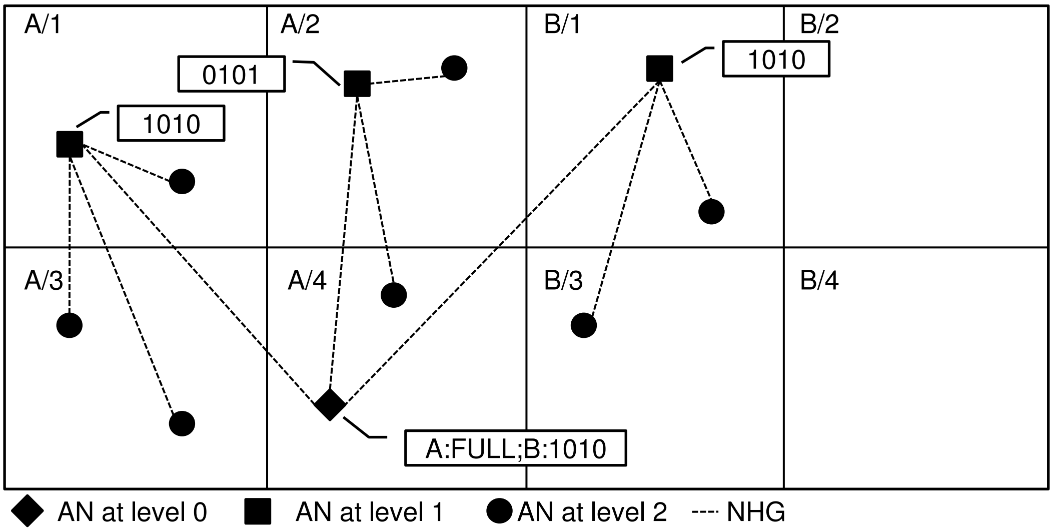

Figure 4.2: Encoding of reachability summaries

The summaries are efficiently encoded using bit-vectors. A node sorts its direct child-locations according to their unique identifier, for instance in lexicographical order. The i-th child-location is assigned to the i-th bit of the vector. The case when a node knows a route to all of its child-locations can be encoded by setting a single flag (FULL) and skipping the bit-vector. A reachability summary of an AN at locations L includes recursively the summaries of the ANs of decendant locations of L.

To further limit the size of summaries at level n, locations below level $n+k$ can be omitted, leading to false positives that need to be resolved at a lower level. The summary size is limited by the number of locations in the SLM, e.g., in case of 1000 locations the size is limited to 125 Bytes.

Figure 4.2 depicts a sample for this encoding. In the figure, 1 indicates a route to the respective child-location. For instance, the AN_B knows a route to its children at B/1 and B/3. Thus, the first and the third bit of its bitvector are set.

In order to detect sibling ANs, i.e., ANs of the same location, an AN that receives an advertisement for a location for which it already knows another valid route, forwards this advertisement to the AN of this route. With this mechanism, sibling ANs can maintain routes to each other. Especially, ANs at the root level need to know their siblings, because no higher level AN is available.

4.1.3 Symbolic Anycast Routing

The *Symbolic Anycast Routing* (SAR) is a network service primitive that enables sending a message to any node at a specific symbolic location, i.e., the target location, representing an area of the SLM. More precisely, SAR delivers a message addressed to target T at a node associated with T. Note that any relay of the message within T could easily deliver the message locally to fulfill the anycast requirement. However, in the following, SAR message are delivered to an associated node AN_T. A message is discarded if connectivity permits delivery. Next, the basic routing along the NHG structure is presented. Although effective, an optimization to increase the efficiency of message forwarding is presented afterwards.

Basic Anycast Message Forwarding

The basic anycast algorithm forwards a message along the links of the NHG overlay. First, the sender forwards it to the AN of the sender location. From there, the meassage is forwarded to the parent AN. This process is repeated until an AN of a location is reached that covers the target location. Then, the message is forwarded to the child AN that maintains a route to the target location. This process is repeated until the target location is reached.

Although routing towards a higher level AN of the NHG is simply done by following the route towards the parent AN, routing towards a lower level AN is more challenging since possibly multiple nodes are associated with a single child location and, therefore, can be chosen for forwarding. In such a case, a route is said to be branching. Three basic classes of topologies need to be considered when routing down the hierarchy (cf. Figure 4.3).

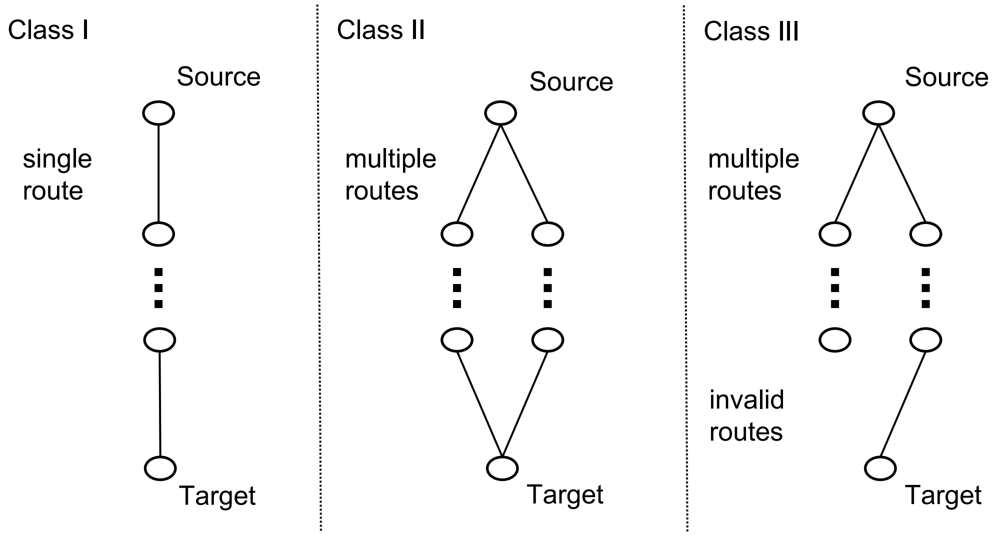

Figure 4.3: Routing Structure Classes

Class I is the simplest case where only one node is associated with a child location. Class II represents the case where multiple nodes are associated with a child location, and each of these nodes knows a route to the target location. Although forwarding a message to any of these nodes is effective, the resulting route length may differ. Class III represents structures that occur when multiple nodes are associated to a child location, and only a subset of these nodes knows a route to the target location. Such structures are typically the result of partitions within single locations, where ANs of a location are only connected through higher level ANs. In this case, the message needs to be forwarded to a node that knows a route to the target location to avoid backtracking.

The proposed routing algorithm tackles the challenges introduced by branching routes (Class II and III) using two mechanisms. First, it uses the reachability summaries to determine which of the ANs of the child location knows a route the target. Second, it heuristically chooses from those ANs, the one that results in the smallest route length. In essence, the algorithm forwards a message to the associated node of the child location for which the geographic distance d_{geo} to the target is minimal and that maintains a route to the target. If multiple ANs have the same geographic distance to the target, the message is forwarded to the AN with minimal topological distance in the underlay.

Backtracking is used to resolve cases, when a target location is not included in the reachability summary of a node, i.e., none of its direct child ANs knows a route to the target. In essence, a node forwards the message to one of its parent ANs, since these have more global information. The parent selection is performed based on d_{geo} as introduced in the previous paragraph.

With this mechanisms, a message that cannot be delivered would finally be routed to a root AN. However, depending on the number of root ANs they might become bottlenecks. Therefore, the sender of a message can set a limit for the maximum number of hops in the underlay to take, for reaching a higher level AN. This allows for trading-off between the probability for successful delivery and routing overhead. A small limit may prevent that gaps between nodes of a partitioned location are bridged by forwarding via nodes of higher level locations.

Exploiting Shortcuts for Forwarding

Although forwarding along the NHG is effective, shortcuts in the routing structure can be exploited to reduce the overall path length, e.g., a direct route between two ANs can be used skipping several intermediate ANs. Moreover, this optimization also relies on the NCG, i.e., routes between neighboring locations, to get closer to the target location, rather than taking indirections via NHG routes.

The idea is to greedily decrease the geographic distance d_{geo} to the target while limiting the effort for recovery by forwarding along the hierarchy if the greedy route turns out to be a "dead end" in the next (greedy) routing step.

Algorithm 4.1 SAR: Exploiting Shortcuts

```
Require: $\mathbb{L}$ list of routing entries, $C$ current location, $T$ target location
  next ← $\text{SAR}_{\text{hier}}(C, T)$    // next overlay hop
  for all $r \in \mathbb{L}$ do
    if $d_{\text{hier}}(\text{pos}(r), T) \leq d_{\text{hier}}(C, T)$ then
      if $d_{\text{geo}}(\text{pos}(r), T) < d_{\text{geo}}(\text{pos}(\text{next}), T)$ then
        next ← destination($r$)
      end if
    end if
  end for
```

Therefore, the cost to reach the target along a route of the NHG is estimated based on the hierarchical distance d_{hier} between locations. A destination is only considered in the greedy algorithm if its distance d_{hier} to the target does not exceed that of the current node. If greedy routing fails, recovery is performed using the basic anycast message forwarding algorithm SAR_{hier}. Algorithm 4.1 shows the algorithm in detail.

Figure 4.4 shows two examples of optimized forwarding based on possible routes for this sample topology. Instead of routing along the NHG, a message from source S_1 to the target location B/3, can be directly routed to the root AN. From there, a direct route to the AN of the target B/3 allows for skipping an intermediate AN for location B. A message from source S_2 to A/3 is forwarded to A/4, which is closer to the target and, finally, the message is directly forwarded to the target.

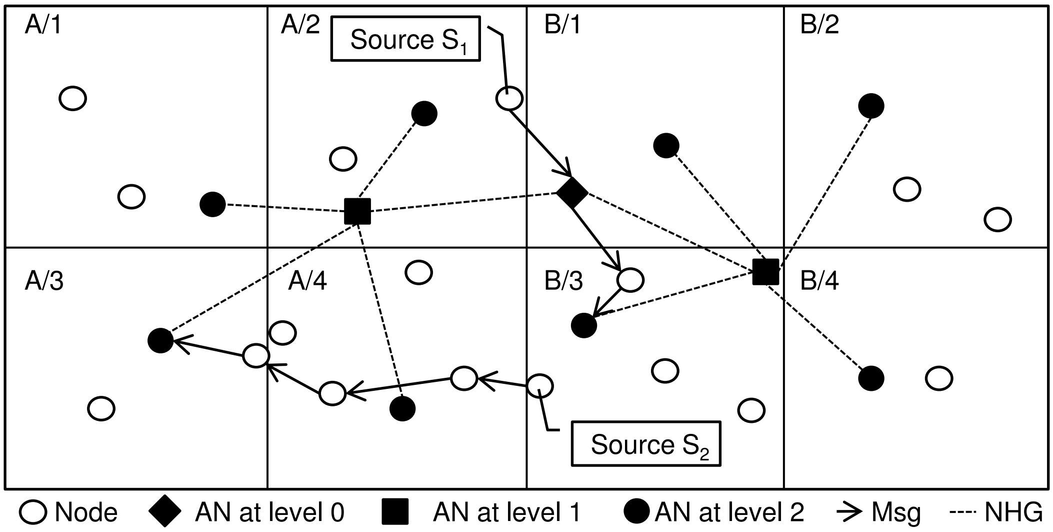

Figure 4.4: Shortcuts for path length optimization

4.1.4 Symbolic Geocast Routing

The *Symbolic Geocast Routing* (SGR) is a network service primitive that enables sending a message to all nodes within the target location. The basic idea is to use SAR for routing to the target location and then distribute the message within this area.

However, simple message distribution within the target area, using scoped flooding restricted to the target location, fails in case of a partitioned target area (Class III; cf. Section 4.1.3). When a message is at an AN of one branch it has to be routed via a parent AN to reach the AN in the second branch. Therefore, SGR routes a message to a node that is associated with the k-parent location of the target using the SAR algorithm. Every AN that receives such a message checks if it knows a route to the target location using its reachability summaries. If so, it sends a message along every route it knows to the target. Duplicate forwarding is suppressed. The ANs of the target location initiate a scoped flooding in their location on receiving a geocast message. As long as at least one route to each part within the target location is known, the message can be delivered to the complete location.

The selection of k influences the effectiveness of the SGR algorithm, and offers a trade-off between efficiency and effectiveness. A small value may result in the delivery of the message to only a subset of the ANs that represent the target, e.g., when a partition within the target is only resolved at a higher level of the hierarchy. Therefore, the sender of a message specifies k. Typically, a value of one is sufficient and a higher value is only needed in few special cases where large locations are partitioned.

4.2 Hybrid Routing

The symbolic routing algorithm presented in the previous section manages to deliver messages within a single local SLM. However, a WMN might cover several non-connected SLMs, e.g., each building of a campus might be modeled by a separate SLM. However, symbolic routing between these buildings fails because there is no direct connection between the locations of different SLMs. This section discusses two approaches to routing between SLMs. Section 4.2.1 presents a hybrid routing approach integrating geometric routing with symbolic routing. Section 4.2.2 propose a mechanism to build a single SLM for routing from the local SLMs.

4.2.1 Hybrid Location-based Routing

The basic idea of hybrid location-based routing is to use symbolic routing within the area covered by an SLM and geometric routing to forward messages where only geometric coordinates are available. For routing within an SLM, the routing algorithm of the previous section can be used unmodified. For geometric routing, any position-based geometric routing algorithm from the literature like GPSR [KK00] can be used.

In order to use geometric routing, geometric positions, i.e., simple point coordinates, are assigned to the SLMs. To assure correct routing, this position, which is easily assigned manually to each SLM, must be located within the SLM.

Routing from a symbolic location L_1, in the SLM of building B1, to another location L_2, in the SLM of building B2, requires three phases (cf. Figure 4.5). In the first phase, symbolic routing from L_1 to a node at the border of the building knowing a geometric coordinate is used. In the second phase, geometric routing from this border node to another node at the border of building B2 that has a neighbor node within building B2 is used. In the third phase, symbolic routing within building B2 to L_2 is performed. Note that this general case also includes the case of routing a message from outside an SLM to a symbolic location within an SLM, and the case of routing from a symbolic location to a geometric coordinate.

The first phase can be achieved in different ways. One simple solution is to let the sender issue a query for a node with a geometric coordinate outside the building by using an expanding ring search, i.e. sending broadcasts with increasing time to live values (TTL) until such a node is found, and the query result is received by the sender. If multiple nodes are found, the sender can choose the one with a coordinate closest to the position of the target building. This solution assures that there is a route to the discovered node following the reverse path of the query result. However, this solution induces high message overhead for the expanding ring search.

Another possibility is to learn geometric coordinates of *border locations* of the building and store them with the SLM. For instance, when a node enters a building at a certain symbolic location, it can assign the last known GPS coordinate to this location. Then the sender can directly send messages via symbolic routing to a border location of its SLM with a geometric position close to the target building. If a node at this border location has a neighbor outside the building, it will start the second phase. Otherwise, another border location is chosen.

In the second phase, the message is routed geometrically towards the position assigned to the SLM of the target (cf. Figure 4.5). Phase three starts as soon as a node with a symbolic coordinate within the target SLM is found. Since the geometric position is within the building, it cannot be reached using greedy geometric forwarding. Thus, routing algorithms such as GPSR use perimeter forwarding when greedy forwarding fails, i.e., no neighbor node with a position closer to the target than the current node can be found. In the example, this mode is triggered at the border of building B2. Perimeter forwarding relays the message to nodes at the border of the building (cf. Figure 4.5). If one of these nodes has a neighbor inside the building, it will forward the message to this node and start the third phase.

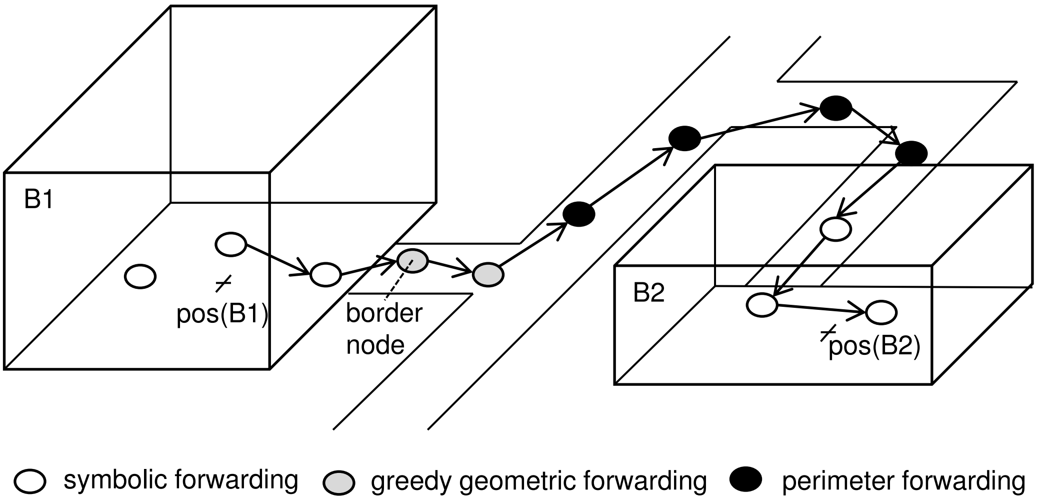

Figure 4.5: Hybrid routing approach

Finally, in phase three, unmodified symbolic routing is used within the building.

4.2.2 Integrated Symbolic Location Model

The second approach constructs a single SLM from separate SLMs. A simple approach would be to link the root locations of the separate SLMs as children of a newly inserted root location. Although effective, this might lead to sub-optimal routing structures because the resulting LNG is partitioned, i.e., there is no flat path between the SLMs. Therefore, only hierarchical routing can be used.

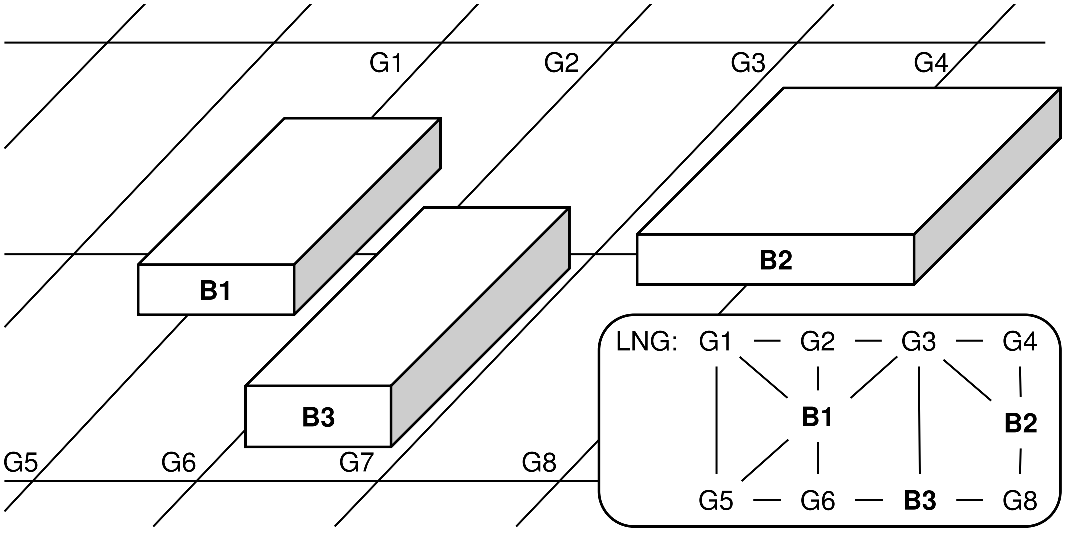

Figure 4.6: Integrated symbolic location model with learned LNG

Therefore, not only a new root location is added, but also additional *symbolic bridge locations* to define flat paths between the separate SLMs. A regular grid structure is the base for the bridge locations. Grid cells are interpreted as symbolic child locations of the root location. A function that is known to all nodes allows mapping of geographic coordinates to grid cells.

In order to allow for flat routing, the LNG includes links between adjacent grid cells and the locations of the SLMs. A fine-grained grid structure is proposed to preserve the accuracy of geographic position information, and to learn additional links between grid cells based on the typical network connectivity. Consider two nodes within 1-hop communication range, say N_1 and N_2, located in two different – not necessarily neighboring – grid cells. Since these two nodes are able to communicate, a link between these two locations is added to the LNG. With this method, the links of the LNG reflect typical network connectivity and, thus, serve as hints for routing. For instance, in Figure 4.6 a node at B1 has a direct neighbor at G3 and, therefore, the LNG is added a connection between these two locations in addition to the connection between adjoining locations. At the same time, the risk of reaching a dead-end is reduced since paths that have a higher chance of network connectivity are preferred.

4.3 Data Storage Service

Based on the symbolic and hybrid routing mechanisms, this section sketches a location-based data storage service. This mechanism for the management of dynamic data such as measurements or control information is similar to the service for location-based data storage based on geographic coordinates [Dud09]. As also stated in previously published work [DWM08], the mechanisms based on geographic coordinates can easily be adapted to support symbolic and hybrid coordinates as well.

The basic idea of this service is to preserve the locality and store location-based data close to the location where it was acquired or where it is required. This way, a location based-query can be resolved by addressing the respective location.

In essence, this principle is adapted by routing data that is to be stored to the symbolically addressed target location. At the target location, the data is delivered to an associated node (AN) of that location, where it is stored. Note that data must not necessarily be stored at the ANs. Instead, dedicated nodes for data storage might register with the ANs of the location for which they manage data. The ANs then can relay updates or queries for location-based data to these dedicated storage nodes.

To query the data that is mapped to a specific symbolic location, a node sends a query to this location using symbolic routing. Each AN of the target location resolves the query locally or by forwarding it to the registered storage nodes. The results are returned on the reverse path to the source of the query.

4.4 Evaluation

In this section, the performance of the *Symbolic Anycast Routing* algorithm (SAR) is evaluated. Since *Symbolic Geocast Routing* (SGR) is essentially SAR with subsequent flooding, it is not evaluated independently. Since hybrid routing sequentially relies on symbolic and geographic routing, its performance is the result of the individual performance of symbolic and geographic routing. Details about the performance of a geographic routing protocol can be found in [KK00]. The following variations of symbolic routing were implemented using the network simulator ns-2, and evaluated:

SAR: The mesh nodes send advertisements and build up the routing structure as described in Section 4.1.

FLAT: In contrast to SAR, advertisements are only sent for leaf locations, i.e., no NHG is established and, therefore, routing is done in a greedy way as explained in Section 4.1.3. If the greedy mode fails, the message is discarded.

These approaches were evaluated with respect to packet delivery ratio, routing overhead, and path length.

Packet delivery ratio: The ratio between successfully delivered messages and the number of initiated message transfers.

Routing overhead: Average number of routing control messages sent to build up the routing structure per node and second. This metric includes sending advertisements, sending replies to advertisements, and forwarding of advertisements to siblings.

Path length stretch: The average underlay path length of successfully delivered messages divided by the minimum path length.

To determine the performance according to these metrics, each node sends a message without any hop limit, every ten seconds, to a randomly chosen destination location within its own partition. The payload size is set to 100 Bytes representing for instance short location-based notifications. To prevent this traffic from interfering with the mechanisms for routing structure maintenance, collisions in message transmission are not simulated. Therefore, message delivery is only affected by errors in the routing structure, which allows for measuring unbiased routing performance.

The symbolic location model for the experiments is derived from the floor-plan of the computer science building of the University of Stuttgart, which has a size of 75 m x 75 m. The floor-plan is divided into four quadrants which are in turn divided into 151 leaf locations in total. On the lowest level, this three level model consists of locations of different sizes: small rooms, medium-sized floors, and four large inner courtyards. The LNG is modeled based on adjoining locations. The mesh nodes store a copy of the SLM and they know their current position. Unless stated differently, nodes randomly select a destination location and move with pedestrian speed towards it. Then, after a pause between one and five minutes, they select another destination and move towards it.

The ns-2 extension of a 802.11b interface is configured to a bandwidth of 11 MBit/s and the maximum transmission range r_{tx} is 15 meters. All simulations have a duration of 600 seconds and the reported values are averaged over at least 15 different simulation runs.

4.4.1 Stationary Scenario

In this section, the performance with stationary nodes is studied to get results that are not biased by node mobility. Therefore, the percentage of delivered messages and the path length stretch for different numbers of nodes in the network are measured. Although no periodic retransmission of advertisements is necessary for the effectiveness, δ_{adv} is set to 32 seconds to get averaged results that are more expressive. In addition, the reported values are the average of 100 simulation runs.

Figure 4.7 shows the delivery ratio for different numbers of nodes in the network. SAR achieves a packet delivery ratio of more than 95% of the messages. This number increases

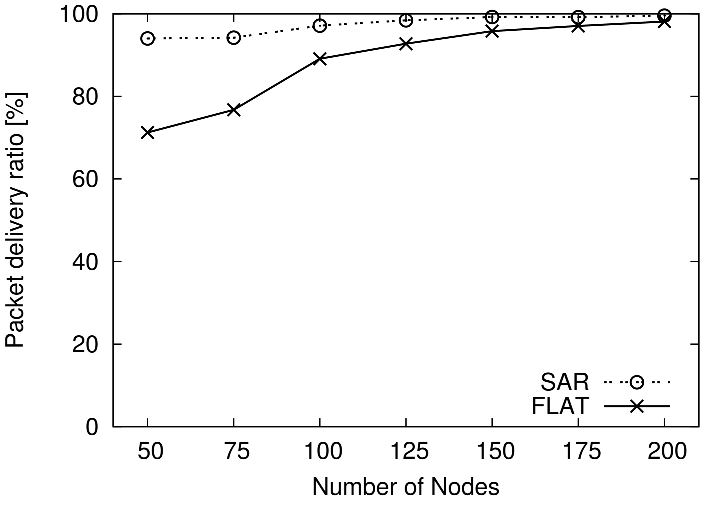

Figure 4.7: Percentage of delivery in stationary scenario

to nearly 100%, when the node density is higher. Not every message is delivered due to the unreliable transmission of advertisements which leads to anomalies in the routing structure. When the node density is low, redundant routes between locations are more unlikely. In that case, if the routing structure is broken due to undelivered advertisements, a message cannot be forwarded on an alternative route.

The performance of FLAT is, as expected, below that of SAR, because greedy routing suffers from void areas in the network. Since perimeter routing is not applicable due to the inaccuracy of position information, a message is discarded if no neighbor is at least as close to the target as the current node. In particular FLAT suffers from low node density, since greedy forwarding is likely to fail. Although the delivery rate increases with increasing number of nodes in the network, FLAT still performs worse than SAR.

Figure 4.8 depicts the path length stretch compared to the minimum path length for different numbers of nodes in the network. FLAT achieves a lower path stretch compared to SAR. This is due to the property of the greedy forwarding: if it successfully delivers a message it achieves this on an almost direct path. In contrast, SAR establishes a routing structure to effectively deliver messages in case of arbitrary network topologies. Although routing along the hierarchy potentially leads to a high path length stretch, the simulation results show that optimized forwarding achieves to limit the stretch ratio to a 23% bound of the minimum path length in this scenario. The reason for both approaches to perform better with fewer nodes in the network is the reduced redundancy of paths. That is, in case of fewer alternative paths it is more likely that the shortest path is chosen.

In order to study the effect of a limited message lifetime, i.e., a hop limit for messages, Figure 4.9 plots the delivery ratio and the path length stretch in a scenario with 50 nodes

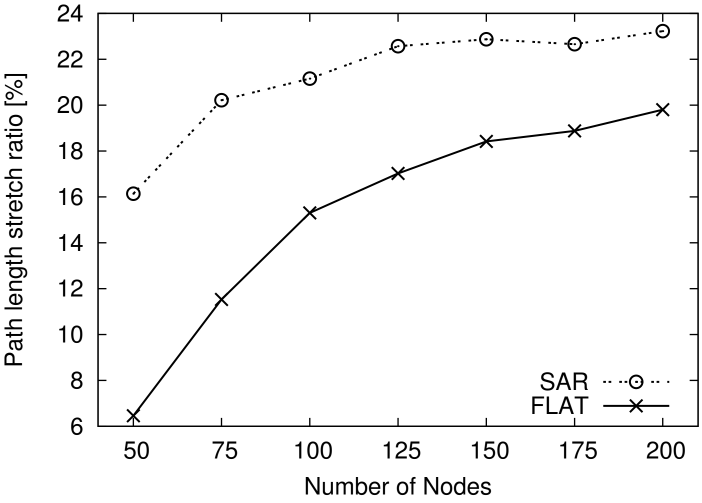

Figure 4.8: Path length stretch in stationary scenario

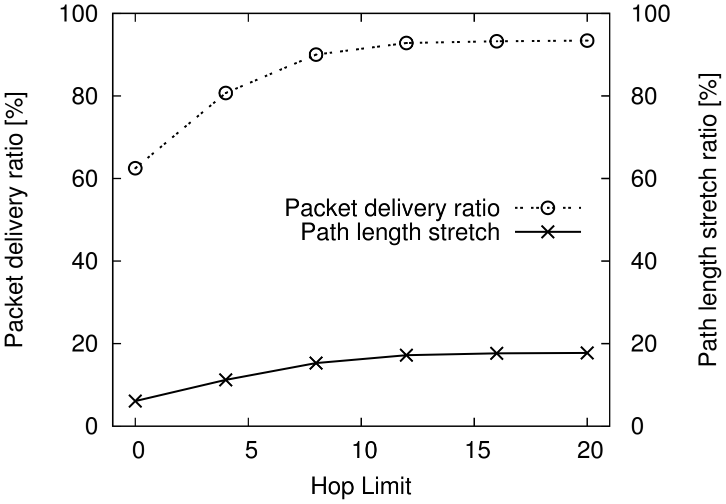

Figure 4.9: Effect of hop limit in stationary scenario

for different values for the message lifetime. In this scenario, a high number of locations are partitioned. Thus, it allows to study the robustness of the routing depending on the selected hop limit. Obviously, the delivery rate suffers from low limits. With a limit of zero, the delivery rate is more than 30% lower than with a limit of 20. The distance of an AN to its parent grows, on average, exponentially with its level in the hierarchy. Therefore, the delivery rate depends logarithmically on the hop limit.

The path length stretch shows a similar behavior. When the hop limit is low, fewer messages are forwarded through possibly long detours within the hierarchy. In addition, when the delivery rate is low, the average path length is reduced resulting in a reduced path length stretch.

4.4.2 Mobile Scenario

This section deals with the performance of the routing when it is subject to highly dynamic node topologies. Although nodes move at pedestrian speed, the low transmission range of 15 meters leads to frequent changes of the topology. Thus, this scenario shows the behavior of routing under challenging conditions. The effect of the advertisement interval on the delivery rate, the routing overhead and the path length is evaluated. The number of nodes is set to 100.

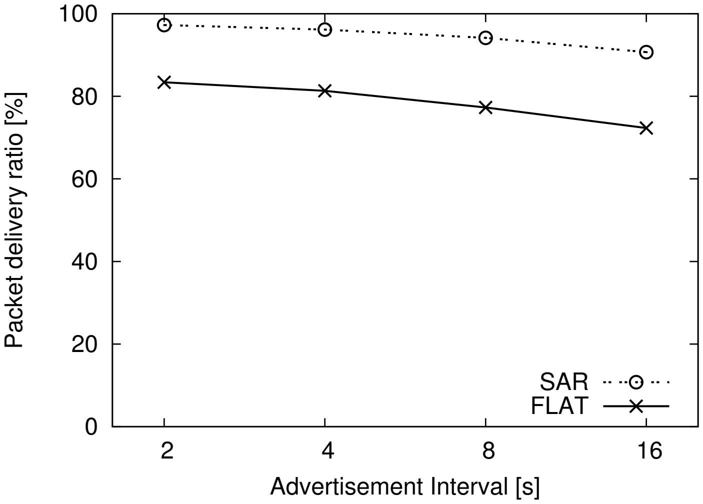

Figure 4.10: Percentage of delivery in mobile scenario

Figure 4.10 depicts the delivery rate. Similar to the results of the previous section, SAR performs better than FLAT. Both approaches depend on the advertisement interval. An interval of two seconds is small enough to almost fully compensate the mobility, i.e., the delivery rate is only slightly lower than in the experiment with stationary nodes. The delivery rate drops when ANs send advertisements at a lower rate.

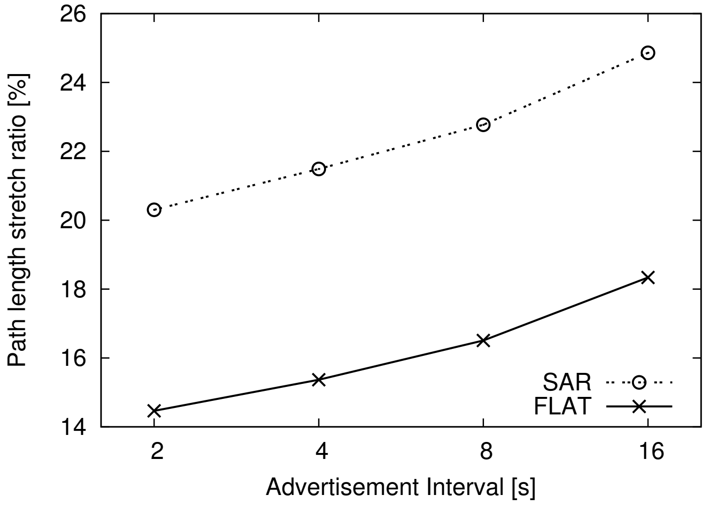

Figure 4.11: Path length stretch in mobile scenario

The effect of the advertisement interval on the path length stretch is depicted in Figure 4.11. The gap between SAR and FLAT is caused by the same reasons as in the stationary experiment. More interesting is the behavior that the path length stretch increases with the interval. This is due to the increased probability of route breaks. Consequently, longer alternative routes were chosen by the algorithm. This effect allows to trade cost of proactive routing overhead for the cost of reactive message forwarding. This effect also explains the small drop of the delivery rate (cf. Figure 4.10).

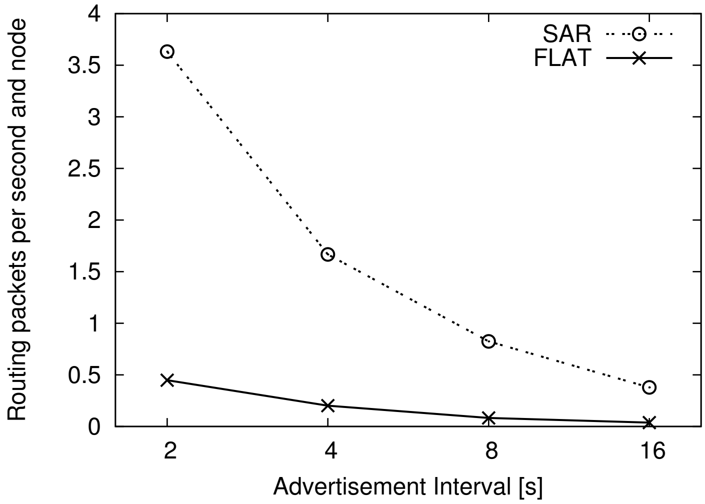

Figure 4.12: Routing overhead in mobile scenario

Figure 4.12 shows that the routing overhead increases with the advertisement rate.

However, FLAT sends less management messages, because advertisements are only sent for the leaf locations and because no hierarchy needs to be maintained. Although the routing overhead of SAR is higher than that of FLAT, it achieves a performance nearly as good as without mobility with an interval of two seconds and at a cost of only about 3.5 messages per node and second. With an interval of 16 seconds SAR still achieves to deliver more than 90% of the messages with a routing overhead of less than 0.5 messages per node and second. With this small proactive overhead the number of expensive reactive flooding-based route discoveries can be significantly reduced.

4.5 Related Work

There have been numerous works on routing protocols for ad-hoc networks [CEM+08]. These protocols can be divided into two main classes: topological routing and geographic routing. Topological approaches such as AODV [PBRD02] are not well suited for location-based services, since they do not include means for geographic addressing. In contrast, geographic routing protocols could be utilized to forward location-based messages since they utilize geographic information for routing and are highly scalable due to forwarding based on local geographic knowledge only.

Greedy Perimeter Stateless Routing (GPSR) [KK00] is a well-known representative of this class and uses perimeter routing when greedy routing fails. Other representatives such as [LBB05] further improve the performance of routing. However, geographic routing approaches assume devices to know their geometric coordinates (longitude, latitude). Especially indoors or if nodes are not equipped with a positioning device like GPS, these are not known. Moreover, as studies show [KLH04], geographic routing significantly suffers from inaccuracy of location information. Therefore, these protocols are not applicable for routing on symbolic coordinates. However, they can be used to bridge areas where only geometric location information is available, i.e., in case of hybrid location models.

Approaches that rely on virtual coordinates [CCDU05] allow for geographic routing without the need for physical position information. However, these approaches suffer from high overhead for updating these coordinates in case of network dynamics. Furthermore, they introduce the unsolved problem of mapping these coordinates to a symbolic location model. The routing protocol [LWKZ08] relies on a hierarchical structure similar to the proposed structure in this dissertation. However, it does not support symbolic addressing and it relies on a central node that manages global topology information.

Routing on symbolic coordinates has been previously considered in wireless sensor networks. Due to the limited resources of sensor nodes, the protocol in [GMH+08] is based on source routing, where a powerful node computes a source route in a centralized way. The message is then forwarded from location to location based on local neighborhood information. However, no elaborate recovery strategy is presented to deal with network dynamics. Besides this dissertation, no other approaches for symbolic routing in WMNs are known.

Several geocast routing protocols [Mai04] have been proposed, which mainly rely on dedicated routing structures or on flooding-based mechanisms and cannot benefit from unicast routing capabilities. Some protocols like GeoTORA [KV03] establish a unicast route to the target area and then initiate a scoped flooding in this area as in the approach presented in this dissertation. However, GeoTORA is a reactive protocol that relies on flooding-based route discovery. Moreover, the aim of this dissertation is to reach every partition of nodes in the target area. In [DR08, Dür10], mechanisms for symbolic geocast in Internet-based overlay networks also relies on hierarchical structures. Although characteristics of mesh networks require new routing concepts, overlay-based approaches can be used where nodes are directly connected to the infrastructure.

In the field of location-based data management [RKS+03], mechanisms for managing context information in mobile ad-hoc networks [DMR06, Dud09] have been well studied. Since these mechanisms are limited to geographic coordinates, they are adapted and support for symbolic coordinates is added by this dissertation.

4.6 Summary

This section presented basic group communication mechanisms as a basis for the coordination mechanisms. The anycast and geocast communication mechanisms allow for addressing specific nodes based on their symbolic location. The proposed routing algorithms are tailored for WMNs. In essence, a hierarchic routing structure (NHG) and a connectivity graph (NCG) between dedicated nodes, structured according to a simple symbolic location model, is created and proactively maintained. Message forwarding is done greedily along paths of the connectivity graph and, if this fails, through the hierarchic routing structure. It was shown that routing achieves high message delivery rates at low routing overhead in terms of routing messages and path length stretch. More precisely, routing achieves nearly 100% delivery rate in a stationary network, and shows only slightly reduced delivery rates in scenarios with node mobility. The achieved underlay path length of the symbolic routing exceeds the minimal path by at most 25% for a wide range of scenarios.

5 Coordination Algorithms for Public Sensing

This chapter discusses and presents the algorithms and mechanisms for the efficient data acquisition according to data quality requirements. Depending on the type of a virtual sensor, different classes of physical phenomena can be monitored. Main parts of this chapter have been previously published in [WDR09, WDR10a, WWDR11].

The concept of virtual sensors is introduced and formalized in Section 5.1. Based on characteristics of the phenomena that is to be observed by these virtual sensors, different algorithms for the coordinated sensing are presented. First, algorithms for coordinating the monitoring of phenomena at a specific location (cf. Section 5.2). Second, algorithms for monitoring of phenomena along a road segment (cf. Section 5.3). Finally, algorithms that are especially tailored for the monitoring of highly dynamic phenomena, such as mobile objects (cf. Section 5.4). Afterwards, these algorithms are evaluated by extensive simulative studies (cf. Section 5.5). The chapter is concluded with a discussion of related work (cf. Section 5.6).

5.1 Virtual Sensors as Data-centric Abstraction

A virtual sensor V is a data-centric abstraction that provides access to measurements of a certain type $\textit{type}(V)$ without the need to refer to a physical ID or address of a node. A virtual sensor is logically associated with a spatial region $\textit{area}(V)$. This dissertation considers two basic types of virtual sensors based on the characteristics of the assigned area (cf. Figure 5.1).

Virtual point sensors: The area assigned to a virtual point sensor is a geographic point. In analogy to traditional wireless sensor networks, where nodes are stationary placed at locations for which data needs to be gathered, virtual point sensors provide access to measurements from a specific location. Such measurements can be acquired either directly by sensors integrated in mobile nodes or collected from external sensors placed at the location of the virtual sensor.

Virtual segment sensors: The area assigned to a virtual segment sensor is a line segment, i.e., a part of a road. In essence, a virtual segment sensor is a data-centric

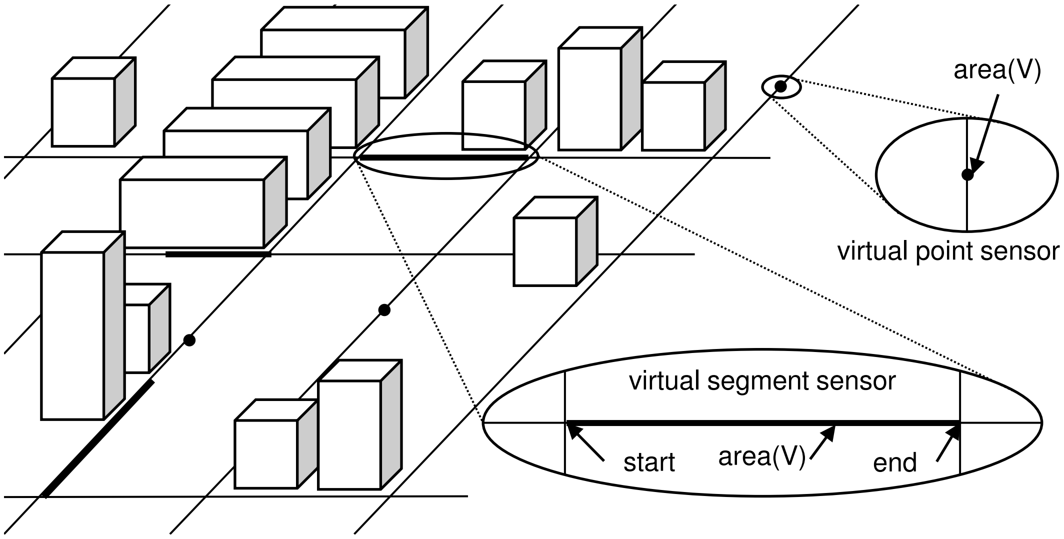

Figure 5.1: Virtual point sensor and virtual segment sensor

abstraction providing measurements acquired along a specific road segment. In contrast to point sensors, it enables generating detailed maps of the observed phenomena.

Note that the concept of virtual sensors can be easily extended to sensors assigned to arbitrary spatial regions. However, main focus of this dissertation are segment and point sensors, since these are most relevant in urban areas, where the movement of nodes is restricted to a street network.

In analogy to a physical sensor, a virtual sensor outputs virtual readings with a sampling interval of V_s. A virtual reading is a collection of measurements acquired by mobile nodes with well-defined spatial and temporal scope. A virtual reading is temporally mapped to the point in time when its acquisition is finished. In detail, the measurements of a virtual reading fulfill the following two requirements.

- The position $pos(M_i)$ of each measurement is in $area(V)$.

- The maximum temporal distance between any pair of measurements of a virtual reading does not exceed V_δ, which, in analogy to a physical sensor, is the maximum duration for the virtual sensor to acquire a virtual reading.

V_δ can also be interpreted as the maximum tolerable time difference between measurements of one virtual reading to be considered a snapshot of $area(V)$. Especially segment sensors require multiple measurements to cover $area(V)$, in general. Thus, such snapshots cannot be acquired instantly. Figure 5.2 shows the relation between V_δ and V_s.

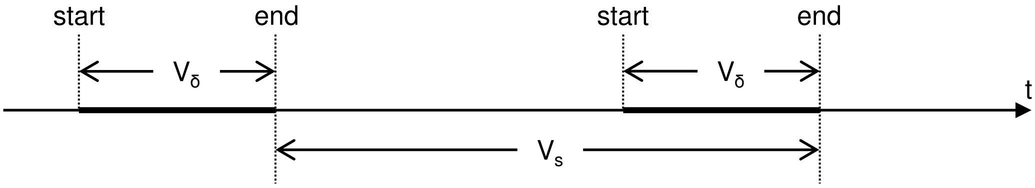

Figure 5.2: Temporal bounds of a virtual reading

The value of a virtual reading for an application increases with the coverage of the reading. However, the coverage of a virtual reading, as well as the coverage of a single measurement, depends on the spatial and temporal dynamics of the observed phenomena.

Environmental phenomena, such as temperature, humidity or the average noise level of an area, change only slowly over time and, thus, monitoring such phenomena with a sampling interval (V_s) of several minutes suffices a wide range of applications. When monitoring such phenomena, the quality requirements of many applications also tolerate temporal deviations (V_δ)of individual measurements, which compose a virtual reading, in the order of a few minutes.

In contrast, monitoring of highly dynamic phenomena such as mobile objects requires fine-grained sampling to capture object trajectories in detail. Since an opportunistic public sensing system has limits to its ability for fine-grained sampling, this dissertation also considers monitoring of higher level information that allows for reduced sampling requirements. For instance, counting the number of mobile objects that pass through an area during a specific time period does not require detailed monitoring of the objects' trajectories.

This dissertation, defines and exploits the metric of spatio-temporal coverage of virtual readings to monitor, i.e., count, mobile objects. A virtual reading spatio-temporally covers an area during a time period V_δ if every object is discovered that is at least once in this area during V_δ.

Especially when monitoring the spatial characteristics of environmental phenomena the spatial coverage of a virtual reading needs to be considered. A virtual reading spatially covers an area, if the distance of every point in this area to at least one measurement is smaller or equal to r_{range}. This reflects the fact that a measurement might not capture effects beyond the sensing range. For instance, an RFID reader is limited to a few meters.

Moreover, an application might desire more than one measurement for each point of an area. For instance, to cope with inaccurate and possibly uncalibrated sensors, redundant measurements at a certain location are necessary to allow for quality-improving sensor fusion. In case of an audio sensor, e.g., a high redundancy is needed to allow for filtering of noise peaks. Therefore, the concept of k-coverage is applied. A point is k-covered if there are k measurements within distance r_{range}. An area is k-covered if every point of it is k-covered. The value of V_k specifies the application requirement regarding spatial k-coverage of virtual readings.

A profile of each virtual sensor is stored at the *Public Sensing Server* (PSS) and disseminated to mobile nodes. Such a profile includes the associated area *area*(V), the type of the sensor *type*(V), and the parameters for specifying the required quality of its virtual readings: V_δ, V_k, and V_s.

5.2 Algorithms for Point Sensors

In analogy to traditional wireless sensor networks, where nodes are stationary placed at locations for which data needs to be gathered, this dissertation provides mechanism to build virtual point sensors (cf. Section 5.1). Analogous to a physical sensor, such a virtual sensor is a data-centric abstraction to access sensor data from a specific location. Such data can be acquired either directly by sensors integrated in mobile nodes or collected from external sensors placed at the location of the virtual sensor. Main parts of this section were previously published in [WDR09].

The *Public Sensing Server* (PSS) specifies the sampling rate V_s, i.e., the requested update interval, for each virtual sensor. The parameter is derived from application requirements. In addition to the V_s values, the PSS manages the position and the time of the last update for each virtual sensor.

The presented approach achieves two main goals. The first goal is to provide applications with new measurements with a sampling interval of V_s. It should be clear that in an opportunistic sensing system the achievable sampling interval depends on the density and availability of the mobile nodes. Thus, no hard guarantees for the actual sampling interval can be given. There might be situations where no node is in sensing range of a virtual sensor for which the PSS requires a new measurement. Therefore, a best effort service is devised where a node acquires a new measurement as soon as it enters the sensing range of a virtual sensor for which the PSS requires a new measurement.

The second goal is to achieve the first goal with as little effort as possible in terms of energy. In particular, unnecessary sensing and communication operations, which are not necessary to achieve the required sampling interval shall be avoided. Formally, if the virtual sensor V has been updated at time t, then further sensing in the interval $(t, t + V_s)$ should be avoided. Note, that this also reduces the load on the PSS and on the mobile nodes, since it reduces the number of updates. Since the mobility of nodes is uncontrollable, no guarantees about coverage and data quality are possible. In order to determine the quality, i.e., coverage, of the sensed data that is returned by a virtual point sensor, the temporal coverage metrics is defined in Section 5.2.1.

The straight-forward approach to acquire measurements reactively whenever they are queried is not effective, since it introduces a potentially high delay as no mobile node might be in sensor range at the query time. Therefore, the proposed algorithms follow a proactive approach to provide new measurements with a sampling interval of V_s. Thus, this approach decouples query processing from update processing.

Unnecessary sensing is the cause for high energy consumption of mobile nodes. Such unnecessary read operations can occur in three situations. First, a node tries to acquire a measurement without being in sensor range of a virtual sensor and, thus, fails to cover the area of the respective virtual sensor. Second, a virtual sensor is read before an update

Algorithm 5.1 Process that runs concurrently with EROA

```
while true do
    doPositionFix()
    if closeToTag() $\wedge$ needUpdate() then
        CROA()
    else
        sleep(DROA())
    end if
end while
```

is required w.r.t. the requested update interval V_s. Third, depending on technological issues, concurrent sensing of nodes within sensor range may lead to collisions. For instance, reading external RFID-based sensors is prone to collisions. In the following, this challenging case of external RFID-based sensors co-located with the virtual sensors is considered. Note that the case of reading external RFID-based sensors includes the case of sensing that is not prone to collisions, e.g., using built-in microphones to measure the noise level. The solutions to avoid these three problems make mobile nodes aware of the location of virtual sensors in their proximity, their respective update times, and concurrently sensing nodes.

The proposed algorithm consists of two concurrently running processes. The first process, *Early Read Operation Avoidance* (EROA) provides the update time of virtual sensors (cf. Section 5.2.2). The second process is listed in Algorithm 5.1. When a node is near to a virtual sensor, the algorithm for concurrent read operation avoidance (CROA) is executed (cf. Section 5.2.4). It coordinates nodes such that a measurement is effectively and efficiently acquired. Otherwise, the algorithm for *Distant Read Operation Avoidance* (DROA) is executed (cf. Section 5.2.3). DROA determines the time a node can deactivate positioning until it may reach a virtual sensor. In the following sections, EROA, DROA, and CROA are discussed in detail.

5.2.1 Temporal Coverage

The goal of the temporal coverage metric is to compare the achieved sampling rate of a virtual sensor with the requested sampling rate V_s. The value of an individual sample, i.e., a virtual reading, is application-dependent. For instance, an applications might require k-coverage with $k = 3$, but accept virtual readings with a minimum achieved value of $k = 2$ to be valid. Only valid virtual readings, which conform to the application requirements, are considered for the temporal coverage.

The temporal coverage quantifies the effective reduction of the sampling rate. It is defined as follows:

Definition 1. *A point in time is temporally covered if at least one valid virtual reading precedes this point by at most V_s.*

Definition 2. *The achieved temporal coverage c_t of a set of virtual readings is the ratio of covered time and the whole monitored time period.*

5.2.2 Early Read Operation Avoidance (EROA)

The idea of *Early Read Operation Avoidance* (EROA) is to make the mobile nodes aware of the next time a virtual reading needs to be acquired to avoid sensing while the server still has a fresh value. To calculate the next sampling time of a virtual sensor, the node needs to know the last update time and the requested update interval V_s. The latter is announced to a mobile node by the PSS when it registers, i.e. connects, with the PSS. In the following, registration is assumed to be done when a node enters the service area of a PSS.

The dissertation introduces two approaches to inform nodes of the last update time of a virtual sensor. The first – proactive management – is based on the idea that nodes cooperatively form an ad-hoc network to manage the update times of virtual sensors. In contrast, the second – reactive management – is based on the idea that the PSS in the infrastructure notifies nodes if an update is needed.

Proactive Early Read Operation Avoidance (EROA/P)

The basic principle of this algorithm is to store the update time of a virtual sensor at least at some nodes in transmission range r_{tx} of the respective virtual sensor so that nodes in proximity of the virtual sensor can query the update time when needed. Note that the sensor range r_{read} is assumed to be much smaller than the transmission range r_{tx}.

At first, the proactive algorithm for managing update times is explained for the case of a single virtual sensor V. Afterwards, the details of the general case with an arbitrary number of virtual sensors are explained. The mechanism to manage update times is based on one-hop broadcast messages and on a locally managed list of virtual sensors at each node. An entry consists of static information such as the ID of the virtual sensor, its position and the requested update interval V_s. Additionally, an entry of this list also includes the update time t_{update}, which is the time of the most recent update of the sensor. Based on t_{update}, a node can determine the time when the PSS requires an update as $t_{\text{update}} + V_s$.

When a node enters the service area it announces its presence to the PSS, which replies with a list of virtual sensors in the service area. From this list, the node initializes a local list of virtual sensors, and fills it with static values like identifier, position and requested update interval.

When a node successfully acquires a measurement that covers *area*(V) it sends the measurement to the PSS. In addition, it also signals t_{update} to its 1-hop neighbors via an

Info message and updates t_{update} locally. As the sensor range r_{read} is small compared to the transmission range r_{tx}, the node that sends the update is close to the center of the disc around V with radius r_{tx}. Thus, the *Info* message may reach further nodes outside the sensor range. All neighbors that receive this *Info* message locally refresh t_{update}. This mechanism, with minimal cost of a single 1-hop ad-hoc broadcast message, prevents nodes within r_{tx} of the virtual sensor from performing an update as long as the server has a fresh update. Therefore, nodes send an *Info* message along with every update.

However, due to mobility, two problems may arise. First, nodes that did not receive the *Info* message, because the distance between sender and node was greater than r_{tx}, will possibly enter the sensor range of the virtual sensor. At the earliest, this happens after the time $\delta_{\text{cover}} = r_{\text{tx}}/v_{\text{maxn}}$, where δ_{cover} is the time a node needs to cover the transmission range at maximum speed. If a node comes into sensor range before the next update is due, i.e., $\delta_{\text{cover}} < V_s$, then it should not sense. Second, a node that received the initial *Info* message may move out of transmission range and thereby miss a duplicate update. Thus, a node needs to check the validity of t_{update}. The refresh mechanism described in the following handles these two problems.

The basic idea of this refresh mechanism is to store and keep the update time at nodes within transmission range r_{tx} of the virtual sensor rather than only broadcasting t_{update} once when the virtual sensor is read. Nodes in sensor range r_{read} of the virtual sensor then can query the update time with 1-hop broadcast messages. To assess the validity of the local t_{update} and to decide whether it needs to be refreshed, nodes manage the time t_{com} of the last communication related to a virtual sensor.

The anticipated update time t_{aut} specifies the earliest point in time for an update of the virtual sensor, as locally seen by a node. It is defined as follows:

$$t_{\text{aut}} = t_{\text{update}} + V_s \tag{5.1}$$

The refresh mechanism is triggered by DROA, when the node gets into sensor range of the virtual sensor. Details about the DROA algorithm are explained in Section 5.2.3. A node verifies if t_{aut} is a future value. In addition, the node checks whether it was in transmission range of the virtual sensor since t_{com}. If so, it assumes that it received all update messages and, thus, its update time is valid. A node can assume that it was in transmission range if the time since t_{com} is smaller than δ_{cover}. Otherwise, the node may have missed a duplicate update while outside transmission range. In this case, it queries the 1-hop neighbors for the most recent update time by sending a *Query* message.

In the *Query* message, a node specifies the identifier and the value of t_{update}. All nodes that receive a *Query* compare their local entry to the received update time. If a node has a more recent update time it sets a timer to send an *Info* message as reply. The reason to postpone the reply is twofold. First, a random jitter reduces collisions through simultaneous replies. Second, the *Info* with the most relevant information, i.e., most recent time, should be sent. Therefore, a node chooses a small random jitter between $[0, j]$ if it knows that no update is needed. The value of j is chosen according to the read interval used by CROA. Otherwise, it chooses a larger random jitter between $]j, 2j]$. On receiving an *Info*, a node cancels its own timer if it cannot contribute a more recent update time. With this mechanism few messages are sent per refresh cycle and the number of cycles for a specific update interval is limited by $V_s/\delta_{\text{cover}}$.

Although it is effective to apply the algorithm described above to each individual virtual sensor, it is more efficient to bundle multiple update times into a single *Info* message to reduce the message overhead and outdated information. Instead of replying immediately, a node that receives a *Query* for a virtual sensor sets a timer and adds its reply to a list. If the node then receives another *Query* while the timer is already set, it adds the reply to the list of replies. On receiving an *Info* message with a more recent t_{update} it removes its own reply from the list. It only cancels its timer if its list is empty. When the timer expires, it sends all valid update times with a single *Info* message. The size of the reply message is limited by the number of virtual sensors on the disc centered at the node with radius r_{tx}.

The cost, i.e., the number of messages, to manage t_{update} with the proactive ad-hoc algorithm increases with node density and with the update interval. The reactive algorithm for managing t_{update}, addressed in the following section, especially suits large V_s values. The problem of high node density is addressed in Section 5.2.5.

Reactive Early Read Operation Avoidance (EROA/R)

In contrast to the proactive EROA/P algorithm, EROA/R relies on the management of update times at the PSS, which notifies nodes within a maximum radius of the virtual sensor to perform an update when needed. It periodically repeats this notification until an update is performed. Such notifications can be sent using, for instance, the symbolic routing approach presented in Section 4.1. With this mechanism nodes ignore virtual sensors if not explicitly notified to sense. In addition, this algorithm guarantees that no node reads while the PSS does not require an update. In contrast, EROA/P may cause early read operations if the time of the last update is lost.

A notification message *Notify* includes the identifier, the position, and the notification radius r_{notify} for the corresponding virtual sensor V. A node ignores the *Notify* message if its distance to V is larger than r_{notify}. Otherwise, V is marked as active and $t_{\text{aut}}(V)$ is set to the current time.

The notification interval δ_{notify}, i.e., the time between two successive notifications, is selected as the minimum time a node moving at v_{maxn} needs to cover a distance of r_{notify}. This ensures that a new notification is only sent when nodes possibly come into sensor range that have not been notified already.

A node sets the state of a virtual sensor to inactive if one of the following conditions is fulfilled.

- The latest *Notify* was received more than δ_{notify} ago.
- The distance of a node to the virtual sensor exceeds r_{notify}.
- An *Info* message was received indicating that an update has been performed. Such a message can be sent either directly by the node performing the update, or by the PSS.

The cost of this reactive algorithm is independent of the required update interval V_s. In contrast, the cost of the proactive algorithm EROA/P grows with the required update interval. Results of a detailed simulative evaluation of the energy consumption of both algorithms are discussed in Section 5.5.

5.2.3 Distant Read Operation Avoidance (DROA)

This section discusses the approach to prevent distant read operations that occur whenever a node tries to read while being outside the sensor range. The basic idea is that nodes determine their position using, for instance, a GPS device. By comparing this position to the known positions of virtual sensors, a node can determine its proximity to virtual sensors. However, a positioning technology like GPS has two challenging characteristics: high energy consumption and inaccurate positioning. The algorithm presented in this section reduces the number of position fixes by computing the time a node can deactivate positioning before it may pass by a virtual sensor. This calculation has to be done carefully such that nodes may not pass by a virtual sensor unnoticed. In particular, this means that the inaccuracy of node positions needs to be considered.

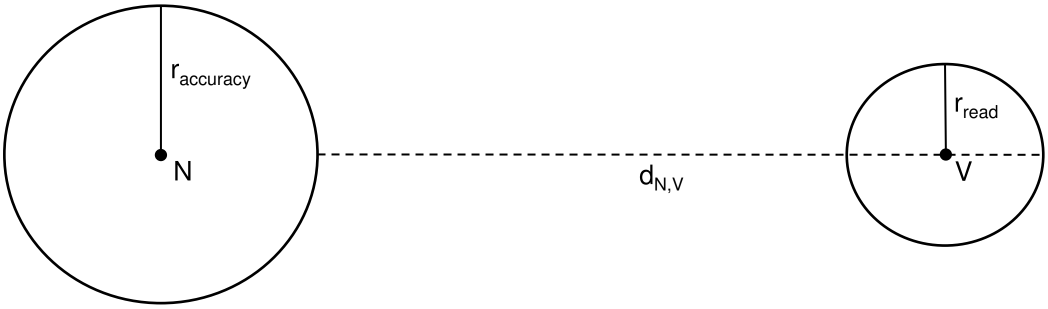

Figure 5.3: Distance metric $d_{N,V}$

The distance a node N needs at least to move before passing by the sensor range of a virtual sensor V is defined as:

$$d_{N,V} = |pos(N) - pos(V)| + r_{\text{read}} - r_{\text{accuracy}} \tag{5.2}$$

While *pos*(N) denotes the GPS position of N, *pos*(V) is the real position of V. A node adds r_{read} to the distance to V, because sensing is possible as long as the node is in sensor range after passing by a virtual sensor. Moreover, due to the inaccuracy of GPS, N might actually be r_{accuracy} closer to V (cf. Figure 5.3).

This definition allows for exceptional cases where nodes are in sensor range of a virtual sensor but do not notice. This can happen when nodes turn while they are in sensor range of a virtual sensor and when nodes move past the virtual sensor. While both cases are only relevant for nodes that almost move at v_{maxn}, the latter is, in addition, unlikely since virtual sensors are assumed to be placed directly where nodes pass by. The evaluation (cf. Section 5.5) shows that these effects are insignificant.

Algorithm 5.2 shows how to determine the time of the next position fix of a node depending on the proposed distance metric.

$$\delta_1 = d_{N,V} / v_{maxn} \tag{5.3}$$

δ_1 is the time a node needs to pass the distance of $d_{N,V}$ at maximum speed. Node N can turn off positioning after a position fix at t_{fix} until $t_{\text{fix}} + \delta_1$. If the next virtual reading is scheduled at $t_{\text{aut}} > t_{\text{fix}} + \delta_1$, N can postpone the next fix even longer. How t_{aut} is derived is explained in Section 5.2.2.

$$\delta_2 = \max(t_{\text{aut}}(V) - t_{\text{now}}, 0) \tag{5.4}$$

$$\delta_{\text{fix}} = \max(\delta_1, \delta_2) \tag{5.5}$$

δ_2 is the time until the next update is needed. δ_{fix} is the maximum of δ_1 and δ_2. It defines the time span to turn off positioning for the respective virtual sensor. Note that $t_{\text{aut}}(V)$ might change when another node updates V. In this case, a node updates t_{aut} accordingly without performing a position fix.

The minimum δ_{fix} value of all virtual sensors determines the time period a node can switch off positioning (cf. Algorithm 5.2).

When V_s is large, δ_{min} is mainly determined by δ_2. Otherwise, it is mainly determined by δ_1. Moreover, the most restrictive virtual sensor determines the position fix interval. These considerations influence the CROA algorithm for efficient sensing as presented in Section 5.2.4.

Due to position inaccuracy, a node cannot definitely determine whether it is in sensing range of a virtual sensor. The *target area* A_{target} of a virtual sensor is defined as the area where the probability of being in sensor range is larger than zero. This area is a disc cen-

Algorithm 5.2 DROA: Computation of position fix interval

```
$\delta_{\text{min}} \leftarrow \infty$
for all Virtual_Sensor V do
    $\delta_1 \leftarrow d_{N,V} / v_{\text{maxn}}$
    $\delta_2 \leftarrow \max(t_{\text{aut}}(V) - t_{\text{now}}, 0)$
    $\delta_{\text{fix}} \leftarrow \max(\delta_1, \delta_2)$
    if $\delta_{\text{fix}} < \delta_{\text{min}}$ then
        $\delta_{\text{min}} \leftarrow \delta_{\text{fix}}$
    end if
end for
return $\delta_{\text{min}}$
```

tered at *pos*(V) with radius $r_{\text{target}} = r_{\text{read}} + r_{\text{accuracy}}$. When a node detects to be in A_{target} and an update is needed, it switches to the CROA algorithm (cf. Section 5.2.4). It switches to the DROA algorithm when it is outside A_{target}.

5.2.4 Concurrent Read Operation Avoidance (CROA)

When multiple nodes read concurrently, read collisions might prevent successful sensing or, at least, lead to redundant reading. For instance, external RFID-based sensors are prone to read collisions. However, high node density can be exploited to increase the probability of successfully reading an external sensor. The basic idea of CROA is to use cheap coordination messages in the ad-hoc to limit sensing to those nodes that have the highest probability of being in sensor range. Moreover, it uses a slot-based scheduling to prevent collisions.

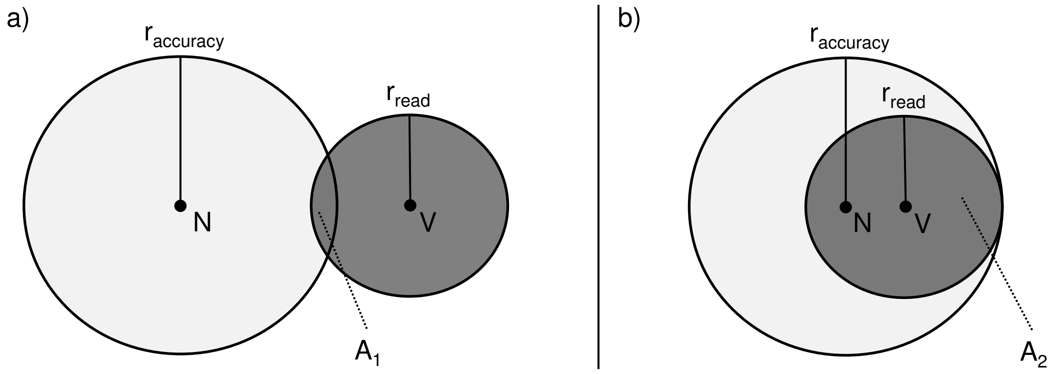

Figure 5.4: Probability of successful reading

After each position fix, a node determines the set of virtual sensors that need to be updated and whose target areas A_{target} cover its position. Then, it checks if it needs to participate in sensing for any of these virtual sensors V_j. Therefore, a node computes its indi-

Algorithm 5.3 The *Concurrent Read Operation Avoidance* algorithm

```
while doPositionFix() $\in A_{\text{target}} \wedge t_{\text{aut}} > t_{\text{now}}$ do
    $p_{\text{success}} \leftarrow$ successProbability(INDIVIDUAL)
    $p_{\text{group}} \leftarrow$ successProbability(GROUP)
    if $p_{\text{success}} > p_{\min} \wedge p_{\text{group}} < p_{\max}$ then
        send(BEACON)
        if read() = SUCCESS then
            sendUpdate()
        end if
    end if
    sleep($\delta_{\text{pause}}$)
end while
```

vidual probability p_{success} of being in sensor range of V_j and the probability p_{group} that at least one of the currently sensing nodes is in sensor range. A node senses if $p_{\text{success}} > p_{\min}$ and $p_{\text{group}} < p_{\max}$ is fulfilled for at least one virtual sensor. A node periodically performs these tasks with an interval δ_{pause}. This interval depends on the maximum node speed and allows to trade-off effectiveness and efficiency of sensing. When the interval is large, nodes risk passing through the target area without sensing; when the interval is small, nodes sense twice at almost the same position.

p_{success} is the probability of a node being within r_{read} of a virtual sensor, i.e, the overlapping zone as indicated in Figure 5.4. In Figure 5.4a, node N has a fairly low probability of being in sensor range of V. Whereas in Figure 5.4b the probability is maximal. Note that the maximum may be below one according to the accuracy.

The equation to compute the probability p_{success} for a specific inaccurate position A_{pos} and the reading area of a virtual sensor A_{read} is the following:

$$p_{\text{success}} = \frac{A_{pos} \cap A_{\text{read}}}{A_{pos}} \tag{5.6}$$

Note, that this idea is applicable to other models for position uncertainty as well, and not limited to uniformly distributed positions.

To compute the success probability p_{group}, a node sends, directly before each read operation, a *Beacon* message as 1-hop broadcast to its neighbors. This message includes the position of the node and, implicitly, the time when the node reads. By sharing this information, every receiver can compute p_{success} for the respective node. Moreover, a node can schedule sensing so that no collisions occur. A node computes p_{success} for all nodes of which it is aware, and that have a higher individual probability than itself. A node is only aware of other nodes that recently signaled their sensing with a *Beacon* message. According to the following equation, a node then computes the probability p_{fail} that none of these k nodes is in sensor range:

$$p_{\text{fail}} = \prod_{i=1..k} 1 - p_{\text{success}}(i) \tag{5.7}$$

When the probability $p_{\text{group}} = 1 - p_{\text{fail}}$ is lower than $p_{\max}$ the node sends a *Beacon* message and senses. The full algorithm is listed in Algorithm 5.3. A node switches to the DROA algorithm when an update is performed or when it leaves A_{target}.

5.2.5 Adaptive Early Read Operation Avoidance (AEROA/P)

Since the physical node movement is not controllable in an opportunistic system, the effectiveness of sensing depends on movement characteristics and node density. In a system with high node density, sensing can fulfill the sampling requirements of the PSS.

Especially in case of high node density, coordination is crucial for efficiency of sensing as it limits redundancy while maintaining effectiveness. However, since the coordination overhead grows with the number of nodes that need to be coordinated, reducing the number of nodes for coordination also increases the overall efficiency of sensing. Thus, the idea of the AEROA/P mechanism is to adaptively exclude nodes from coordination if the coordinated nodes effectively fulfill the requested sampling interval V_s. Thus, reducing coordination overhead and energy consumption of nodes.

AEROA/P assigns only a subset of all virtual sensors to individual nodes. Thus, it reduces the number of nodes that need to be coordinated. If V is not assigned to a node, the node can ignore the respective virtual sensor in the DROA algorithm (cf. Section 5.2.3). Since this algorithm computes the time to deactivate positioning based on the most restrictive virtual sensor, ignoring sensors allows for longer periods of deactivated positioning and, therefore, energy savings.

The PSS monitors the average update interval δ_{average} for each virtual sensor as indicator for the effectiveness of sensing. To inform the nodes about which virtual sensors can be ignored, the PSS adds an *ignore* flag to each entry of the list that is distributed to the nodes as they enter the service area (cf. Section 5.2.2). This flag specifies whether the corresponding virtual sensor is assigned to nodes. Its value is determined as follows:

$$\text{ignore} = \begin{cases} \text{true} & \text{if } \delta_{\text{average}} < V_s + \mathit{delay}_{\text{th}}, \\ \text{false} & \text{else.} \end{cases} \tag{5.8}$$

This mechanism can be used to adapt sustainably but inertly the number of nodes that need to be coordinated at a specific virtual sensor. Virtual sensors are not removed from the list to allow nodes to further participate in the EROA/P and CROA algorithms, because energy consumption for these algorithms is fairly low compared to positioning in sparse sensor environments. Moreover, EROA/P and CROA benefit from high node density.

The number of nodes in the service area changes over time, since nodes continuously enter and leave this area. This mechanism only affects those nodes entering the service area. To adapt rapidly, all nodes that enter the service area ignore a virtual sensor if the *ignore* flag is set. This is necessary because V_s is fairly small compared to the average time a node stays in the service area. The value of $\mathit{delay}_{\mathrm{th}}$ specifies the maximum accepted delay of updates and, thus, allows for trading timeliness of updates for saving energy through extended intervals of deactivated positioning.

5.3 Algorithms for Segment Sensors Supporting Environmental Monitoring

In addition to monitoring specific points of interest using virtual point sensors, this section puts focus on the monitoring of street segments using virtual sensors (cf. Section 5.1). In essence, a virtual segment sensor is a data-centric abstraction to access sensor data acquired along a specific road segment. More precisely, this section focuses on monitoring of static environmental phenomena, i.e., generating maps of phenomena such as noise. Main parts of this section were previously published in [WDR10a].

Since the mobility of nodes is uncontrollable, no guarantees about coverage and data quality are possible. In order to determine the quality, i.e., coverage, of the sensed data that is returned by a virtual segment sensor, the spatial coverage metrics is defined in Section 5.3.1. In essence, the spatial coverage metrics is tailored to the scenario of public sensing. Using this metrics, applications can specify quality requirements for virtual sensors. Based on these application requirements, this section introduces centralized and distributed best-effort algorithms that minimize the energy consumption of mobile nodes, while fulfilling the quality requirements. In particular, these algorithms aim for avoiding unnecessary sensing operations and communication operations to save energy. More detailed, sensing should be avoided at times and places that does not increase quality or that would exceed the quality requirements.

An approach to access measurements of a specific location would be to reactively send a query as a geocast message to the nodes close to that location. These nodes then could sense while they are at that location and send their measurements as reply. However, this approach has a couple of disadvantages. First, it introduces a potentially high delay since nodes might take a while until they enter that location. Therefore, the approach presented in this section is to proactively sense while nodes are on the segment of a virtual sensor, and collect these measurements in the infrastructure where applications can access it.

In detail, several node selection schemes are proposed. Two schemes adaptively select nodes, based on V_δ and V_k, for acquiring a virtual reading: *Centralized Spatial Shaper* (CSS) discussed in Section 5.3.2, and *Distributed Spatial Shaper* (DSS) discussed in Section 5.3.3. Shaping refers to the process of forming the output stream of a virtual sensor. Nodes are aware of the point in time when a virtual reading needs to be acquired (cf. Section 5.2.2). Finally, a mechanism that derives a physical sampling interval according to the maximum sensing range r_{range} of a mobile node is presented in Section 5.3.4.

5.3.1 Spatial Coverage

Depending on the distribution and movement patterns of mobile nodes, it cannot be guaranteed that a virtual sensor meets the quality requirements specified in its profile.

For instance, if no node passes through the segment of a virtual sensor, no virtual reading can be acquired. In order to quantify the actual achieved quality of virtual readings, the spatial coverage metric of virtual readings is introduced in this section.

The goal of the conceived spatial coverage metric is to allow for fine-grained comparisons between the achieved coverage of a virtual reading and the requested coverage as specified in the profile of a virtual sensor. Since in particular coverage "holes" are critical for applications, partial loss of coverage on a segment may not be outweighed by partially exceeded coverage requirements. Figure 5.5 shows three examples, each consisting of a set of measurements, the requested coverage V_k, and the achieved coverage. In the left example, measurements are close and therefore only a small part of the segment is covered compared to the example in the middle. In the right example, no part is 2-covered and fulfills the requirement $V_k = 2$. However, to allow for fine-grained comparisons, 1-covered parts are considered as partially covered.

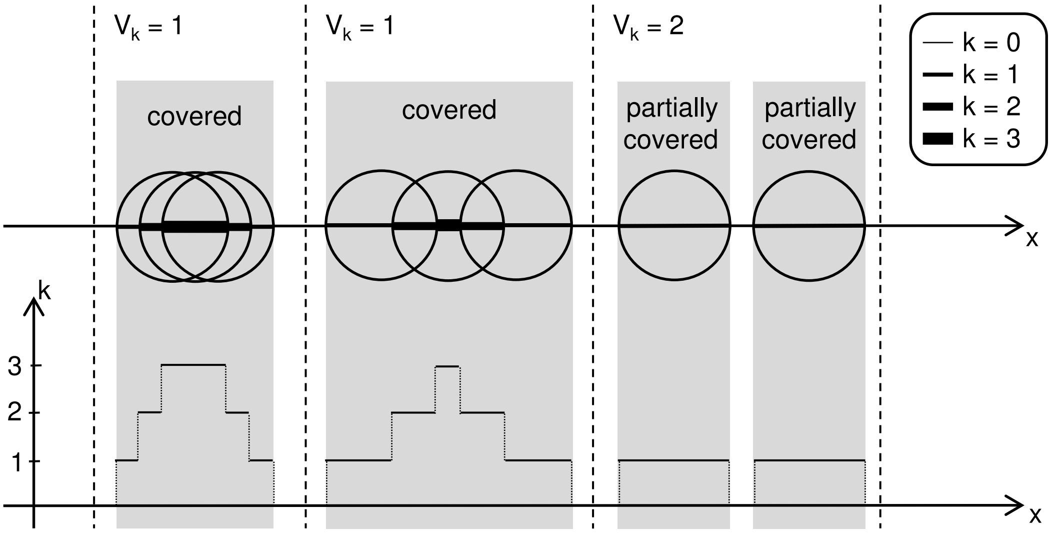

Figure 5.5: Spatial coverage of a virtual reading

The achieved spatial coverage $c_s(V,R)$ of a virtual reading R of a virtual sensor V is defined as follows:

Definition 3. $c_s(V,R) = \frac{1}{|area(V)|} \int_{x \in area(V)} min(k(x), V_k) \cdot dx$

In this definition, $k(x)$ defines the coverage of a point x on the segment of V, i.e., the number of measurements that cover it. Note that in the best case, when the requested coverage is fulfilled, $c_s(V,R) = V_k$. However, the achieved spatial coverage cannot exceed V_k.

5.3.2 Centralized Spatial Shaper

The goal of the *Centralized Spatial Shaper* (CSS) algorithm is to determine and schedule those nodes for sensing that are needed to fulfill the coverage requirements, while minimizing the energy consumption of nodes. This is done by a central instance, i.e., the *Public Sensing Server* (PSS).

A simple approach to select nodes for sensing during a period V_δ would be to select all nodes on a segment. Such a selection mechanism would yield the maximum possible coverage. However, this may lead to redundant measurements if more than V_k nodes move along the same part of a segment during the period V_δ. An improvement to avoid the redundancy that exceeds V_k, would be to deactivate sensing of a node as soon as it enters an area where V_k other nodes already moved and sensed. However, coordination is needed to detect such areas. Moreover, due to position inaccuracy, a small overlap of sensing at such meeting points between nodes is required. Therefore, and to reduce coordination overhead of nodes, CSS minimizes the number of nodes that sense on a segment.

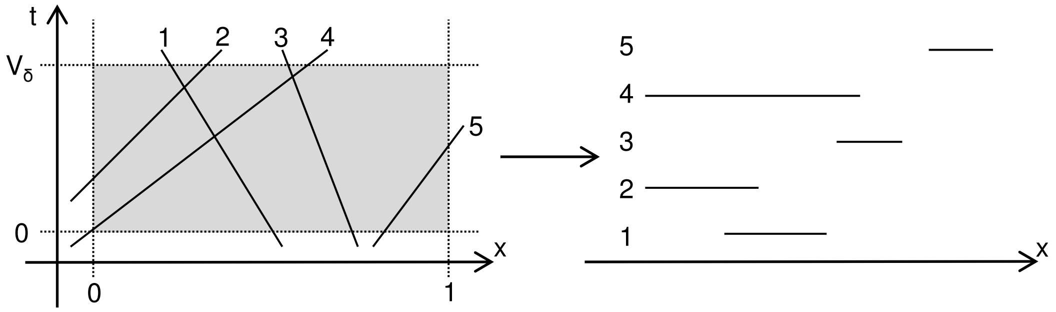

Figure 5.6: Overview of CSS

The basic idea of spatial shaping is, during a period of V_δ, to use movement predictions to select the minimum subset $\mathbb{S}$ of nodes in *area*(V) for sensing that can achieve the same coverage as the complete set of nodes $\mathbb{N}$. Figure 5.6 shows the trajectories of five nodes and the fragments of the segment they can cover during a period of V_δ. In this example, the nodes 3, 4, and 5 form such a minimal subset assuming $V_k = 1$.

Although this selection yields the minimal number of nodes for sensing, certain parts of the segment might still be covered by multiple nodes. For instance, in Figure 5.6, the fragments of nodes 3 and 4 overlap. In order to avoid such redundancy that exceeds V_k, nodes are coordinated, i.e., overlapping parts are split and assigned to the nodes for sensing. Moreover, progress of nodes is monitored and, if nodes deviate from their predicted movement such that they cannot cover their assigned part of the segment, other nodes are selected to take over sensing. The individual elements of CSS are discussed in detail in the following sections.

Minimum Subset Selection

Based on its current speed and the assumption of a uniform movement on the current segment a node can predict the part of a segment it can cover within a period of V_δ. Such a part is referred to as fragment f. Each fragment is defined by a start point f_{start} and an end point f_{end}. Both are one-dimensional coordinates relative to the start point of $area(V)$. Note that one-dimensional coordinates are only used to simplify the presentation of the algorithm.

Formally, the goal of the minimal subset selection is to determine a subset $\mathbb{S}$ of nodes whose predicted coverage $c_s(V,\mathbb{S})$ is the same as the predicted coverage $c_s(V,\mathbb{N})$ of the complete set of nodes $\mathbb{N}$. Moreover, this subset is to be minimal. In contrast to selecting the minimum subset, also fair strategies can be implemented that consider, e.g., the remaining energy level of nodes as a selection criteria. However, here the focus is on efficiently fulfilling the coverage requirements.

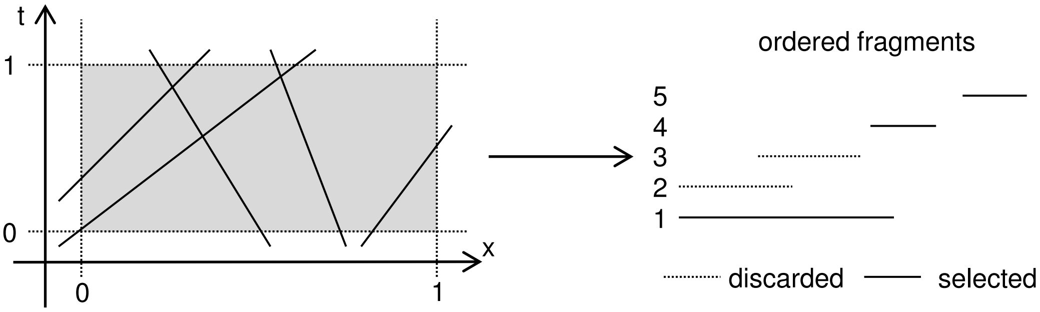

Figure 5.7: Generation and ordering of fragments from predicted trajectories

The algorithm for determining the minimal subset of nodes for sensing can be described as follows. At the beginning of a virtual reading, all nodes in $area(V)$ predict which fragment of the segment they can cover. If the fragment's length is larger than zero, they report their fragment and the respective prediction to the PSS. Based on all relevant fragments, the minimum subset is computed according to the *Minimum Subset Selection* (MSS) algorithm. The output of the MSS algorithm determines the nodes which participate in sensing and, thus, need to coordinate sensing. Figure 5.7 shows the trajectories of several nodes according to their prediction and the resulting fragments based on a given sensing window. In this example, the spatial and the temporal range of this window is $[0;1]$. The figure symbolizes totally covered fragments, which can be discarded if V_k is one, using dashed lines.

The basic idea of the algorithm is to sort the fragments according to f_{start} in ascending order. Fragments that are equal in this value, are sorted based on f_{end} in descending order. Based on this ordering, the algorithm iterates over the fragments and checks for each fragment f whether the currently selected fragments $\mathbb{S}$ k-cover the point f_{start}. If this is the case, the fragment is added to a temporary set $\mathbb{T}$ from which it can be selected afterwards if needed. Otherwise, the fragment f' with largest $f^{'}_{\text{end}}$ is selected from $\mathbb{T}$ and added to $\mathbb{S}$. Fragments are discarded from $\mathbb{T}$ if they are completely covered by the fragments in $\mathbb{S}$. This is repeated until f_{start} is k-covered or until $\mathbb{T}$ is empty. If $\mathbb{T}$ is empty but the coverage is not fulfilled, f is added to $\mathbb{S}$. The algorithm is listed in Algorithm 5.4.

Algorithm 5.4 The *Minimum Subset Selection* (MSS) Algorithm

```
Require: $\mathbb{F}, k$ sorted list of fragments, redundancy requirement
  $\mathbb{S} \leftarrow \varnothing$    // selected fragments
  $\mathbb{T} \leftarrow \varnothing$    // temporary container
  for all $f$ in $\mathbb{F}$ do
    while $|\mathbb{T}| > 0$ and not cover($\mathbb{S}, k, f_{\text{start}}$) do
      removeCovered($\mathbb{T}, \mathbb{S}$)
      if $|\mathbb{T}| > 0$ then
        $\mathbb{S} \leftarrow \mathbb{S} \cup$ popMaxEnd($\mathbb{T}$)
      end if
    end while
    if cover($\mathbb{S}, k, f_{\text{start}}$) then
      $\mathbb{T} \leftarrow \mathbb{T} \cup f$
    else
      $\mathbb{S} \leftarrow \mathbb{S} \cup f$
    end if
  end for
```

The function cover($\mathbb{S}, k, f_{\text{start}}$) checks whether the fragments in $\mathbb{S}$ k-cover the point f_{start}. The method removeCovered($\mathbb{T}, \mathbb{S}$) removes all fragments from $\mathbb{T}$ that are completely covered by the fragments in $\mathbb{S}$. The function popMaxEnd($\mathbb{T}$) removes the fragment f from $\mathbb{T}$ with largest f_{end} and returns it.

Node Coordination

As discussed at the beginning of this section, selected nodes need to coordinate to avoid redundant measurements. Basically, coordination is needed to determine for each notified node where to start and stop sensing. This means that each node has at most two coordination partners to determine start and stop of sensing.

Each node, whose fragment starts at a part of the segment where the coverage requirement is exceeded, coordinates with a node whose fragment ends at the respective part. Figure 5.8 shows eight selected fragments and the coverage they would achieve if they would sense uncoordinated. Assuming $V_k = 2$, the coverage requirement would be exceeded in three parts of the segment (marked as gray). In the left part, (2,3) can be easily identified as coordination partners that need to avoid duplicate sensing where their fragments overlap. Similarly, (7,8) are partners in the right part of the segment. More interesting is the middle part, where (1,5) and (4,6) are selected as partners. Alternatively, (1,6) and (4,5) can be selected as partners.

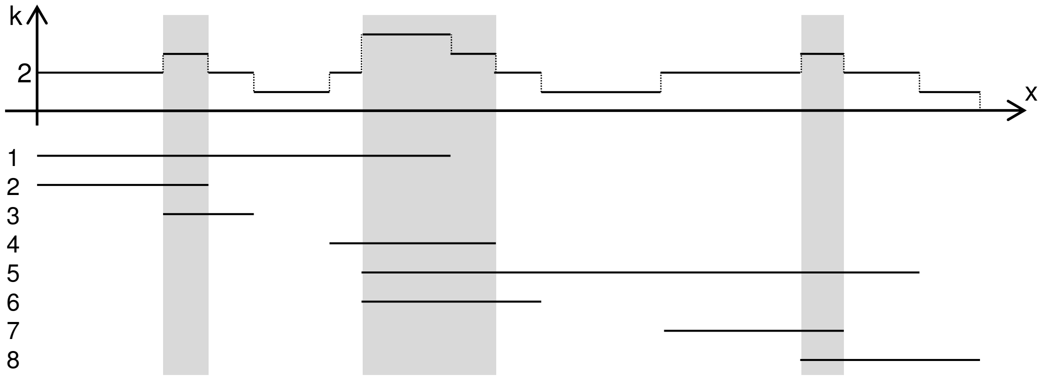

Figure 5.8: Determination of coordination partners with $V_k = 2$

If a node has no coordination partner, start and stop positions are the predicted ones. Otherwise, coordination partners need to coordinate sensing where their trajectories overlap. Figure 5.9 shows a classification of the coordination problem, depending on movement patterns of nodes.

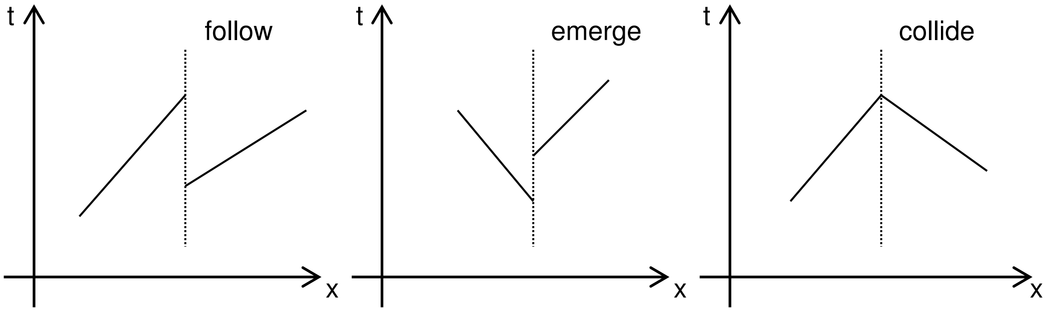

Figure 5.9: Classification of coordination problems

The first class is *follow*, where one node follows another. In this case nodes simply start sensing when notified by the PSS. However, the following node stops sensing at the start point of the leading node. The second class is *emerge*. In this case two nodes pass by each other. Therefore, start coordination is necessary. One node immediately starts to sense and the other starts as soon as it passes the point where the first node started. The second node emerges from a covered area and starts to sense. As a heuristic solution the node with the highest probability to cover the gap between them is selected. The third class is *collide*. In this case, both nodes start sensing immediately. However, coordination is needed to determine the point where to stop sensing of both nodes.

Although coordination for classes *follow* and *emerge* can be determined based on the result of the MSS algorithm, coordination in the case of *collide* is much more challenging, and requires constant monitoring of node movement. The idea of this coordination is to prevent nodes from sensing at the same location, and to prevent nodes from stopping sensing while some gap between them is still uncovered. In the following, this mechanism is described in detail.

Initially, the PSS detects the need for collide coordination and informs the affected nodes along with the notification to start sensing. Based on their predictions, the nodes compute the collision point and consider it as the point where to stop sensing. However, as a node's movement deviates from its prediction, the actual collision point deviates from the computed. To update the prediction at the other node with every change of a node's speed leads to high communication overhead.

Therefore, updates are only sent in two cases. First, if a node is slower than predicted, it notifies the other node before this would stop sensing. Hereby, a new collision point can be computed, and the remaining gap between the nodes is divided and re-assigned to them. Second, if a node is faster than predicted, it updates before it passes into the section of the other node. Hereby, a new collision point is computed and the gap between the nodes is re-assigned accordingly. Both cases are depicted in Figure 5.10. On the left hand side, the predicted collision point is shown. On the ride hand side, both cases to send an update are visualized.

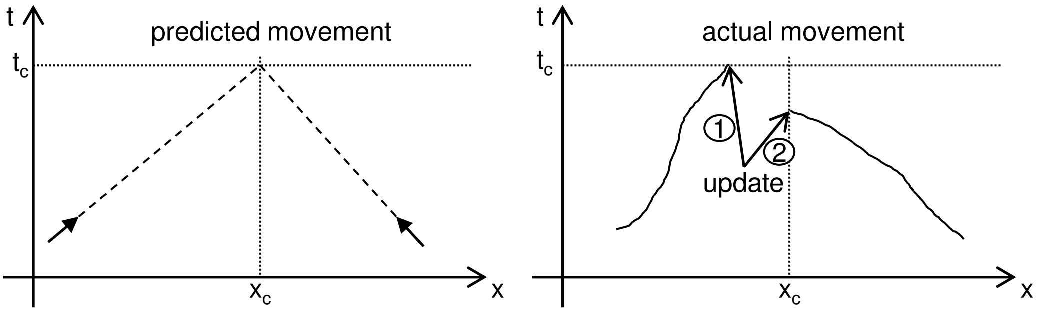

Figure 5.10: Update in collide coordination

Progress Control

Without a limitation of the period V_δ of a virtual reading, the above described mechanisms would be sufficient for coordination. However, if a node cannot cover its fragment within the available read period, a re-assignment of the uncovered fragment needs to be performed. For this purpose, a mechanism similar to collide coordination is applied.

According to the different coordination classes, each node knows a fixed or variable point where to stop sensing. If its current prediction does not allow a node to completely cover its fragment in time, it updates its prediction with the PSS. Note that updates may be deferred to allow for speed fluctuations and to reduce cost for updating. However, deferring updates increases the risk for parts of the segment to remain uncovered. This is the case when a node would have been able to take over sensing if it was informed earlier. Similarly, if a node enters a segment during V_δ or if a node's movement deviates from its prediction in a way that allows it to cover a part of the segment that was not assigned in the MSS, it updates its prediction with the PSS.

Receiving a new prediction or an updated prediction, the server then initiates the MSS algorithm on the respective part of the segment, and notifies suitable nodes to start sensing. Note that the previous results of the MSS are not discarded. In contrast, nodes that were selected for sensing, are now pre-selected in the MSS.

When a node fails that is assigned for sensing, it may not update its measurements and its progress to the PSS. To prevent this from happening, an optional extension to the basic progress control mechanism is proposed, which can be applied in scenarios with a high rate of node failures. Here, nodes send periodic progress reports including their measurements to the PSS. If the server fails to receive a progress report it assumes the respective node to have failed. In this case it initiates the MSS algorithm on the scope of the remaining fragment of the failed node.

5.3.3 Distributed Spatial Shaper

In the previous section, the *Centralized Spatial Shaper* algorithm was introduced. Although it is effective, it depends on the availability of a permanent connection between nodes and the PSS. Although GPRS or UMTS networks are widely available today, their energy consumption for communicating a message exceeds that of WiFi, and usage of these networks produces costs. The *Distributed Spatial Shaper* (DSS) algorithm coordinates the nodes in the area of a virtual sensor using an ad-hoc network to exchange movement predictions of nodes as basis for the distributed coordination.

The basic design principle of DSS is that nodes deactivate sensing only if other nodes explicitly commit to take over sensing for them. Thus, no loss of coverage compared to CSS is possible. In essence, this algorithm consists of the same parts as the *Centralized Spatial Shaper* (CSS). Especially, the *Minimum Subset Selection* (MSS) algorithm, as presented in the previous section (cf. Algorithm 5.4), is an essential part of DSS. In an initial phase, nodes on a segment exchange their predictions in the ad-hoc network. In a second phase, they cooperatively select the nodes for sensing. Then, the selected nodes coordinate sensing and perform a cooperative progress control.

Prediction Exchange

The goal of the initial phase of DSS is to distribute local movement predictions of nodes on a segment among each other as a base for node selection. When a virtual reading is initiated (cf. Section 5.2.2), nodes broadcast their prediction in the ad-hoc network. To prevent simultaneous broadcasts, nodes chose a random delay. On receiving such a broadcast, nodes add the fragments contained in the broadcast to a local buffer. For each fragment, nodes store the ID of the node from which the fragment was first received. Although, due to mobility, these routes may break over time, communication shortly after this phase is likely to succeed.

On receiving such a broadcast, nodes predict the fragment they can cover during V_δ and broadcast this together with their local fragment buffer, if they have not already broadcasted. At the end of this phase, every node stores in its fragment buffer a local view on the fragments of nodes in the segment. The size of the fragment buffer is limited by the number of nodes on a segment. Typically, this number is at most a few dozen, allowing transmitting the buffer in a single frame.

To reduce message overhead, a node broadcasts only if it is selected by the MSS executed on its local fragment buffer. Since each node broadcasts at most one message, this algorithm has, in the worst case, a linear message complexity. However, since nodes suppress broadcasting if their fragment is covered, the number of messages is on average much smaller. Due to its limited message complexity, this phase is rapidly completed. It can be assumed that node movement during that time frame is neglectable.

Cooperative Node Selection

The goal of the second phase is to consistently select nodes for sensing by resolving possible inconsistencies between the individual fragment buffers of the nodes. Using the next hop information associated with the fragments in the fragment buffer, nodes can communicate with their neighbors to resolve these inconsistencies.

Each node performs the MSS on the local fragment buffer. If its fragment f is locally selected for sensing, it also locally determines the fragments with which to coordinate. Then, it contacts the node of the fragment that overlaps f_{start}. Note that also f_{end} could be chosen, without affecting the performance of the mechanism. A node contacts its coordination partner under the assumption that both nodes are selected for sensing. However, due to partial local knowledge, this might be incorrect. If the receiver can determine locally that the sender or itself is not selected for sensing, it notifies the sender accordingly by replying with its local fragment buffer. This process is repeated until each coordination partner is verified. When this is achieved, consistent node selection is achieved.

Figure 5.11 shows an example with three nodes on a segment, where, if $V_k = 1$, only nodes 1 and 3 need to sense. In the example, node 2 initiates the prediction exchange. Nodes 1 and 3 receive the fragment of node 2. They also broadcast the fragments of their local view, and node 2 learns from fragments 1 and 3. In the node selection phase, node 3 is the only one that has a left neighbor based on its local view. It contacts this neigh-

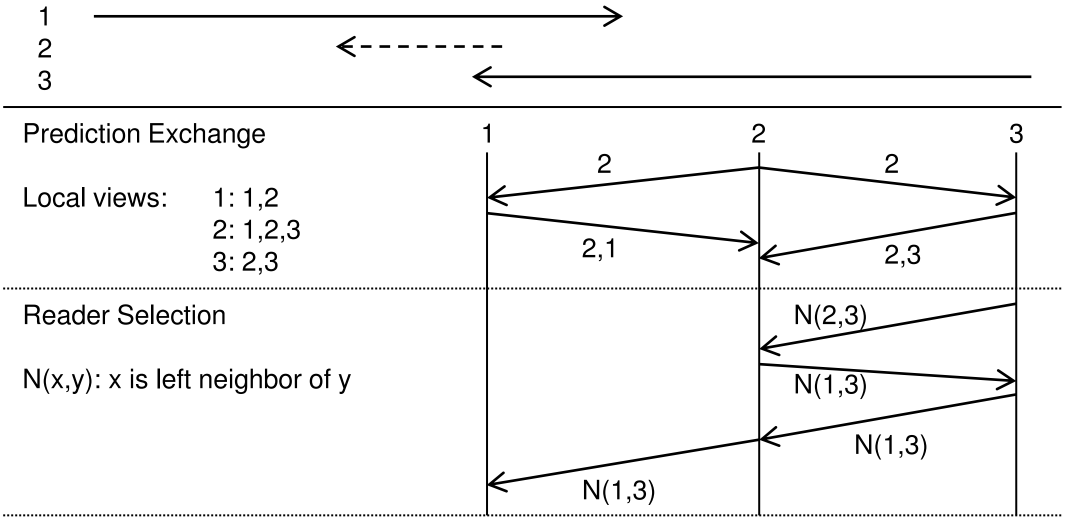

Figure 5.11: Distributed cooperation: prediction exchange and node selection

bor (node 2), which however computes that it does not need to coordinate with node 3. It replies with its local view. Based on this reply, node 3 then again determines its left neighbor and contacts it to initiate coordination.

Cooperative Node Coordination

The idea of the node coordination is to avoid redundant sensing due to overlapping fragments, which is needed between nodes of neighboring fragments. It is performed according to the classification in Section 5.3.2. In case of *follow* and *emerge*, coordination partners determine start and end points for sensing based on the consistent view on their fragments after node selection.

In contrast, *collide* coordination requires constant monitoring of movement. Since routes may break due to movement, a node addresses its prediction update directly to the coordination partner. The idea of this is that coordination is only needed if neighboring nodes are close. If the coordination partner is in transmission range it receives the message, computes a new meeting point and replies. Otherwise, the delivery of the update is not crucial, since there is no risk of overlapping trajectories. However, in this case, the sender needs to re-send an update regularly. The sender derives the interval based on an estimation of the receiver's relative speed and on the communication range. Note that this approach does not require a routing mechanism.

Cooperative Progress Control

The idea of the progress control is to monitor the progress of sensing and, if this deviates from the predicted, to assign sensing tasks to other nodes. In essence, when a node's prediction indicates that it cannot cover its fragment in time, other nodes need to take over sensing. Therefore, it broadcasts its current prediction, its fragment, and its neighboring fragments to its direct communication neighbors. With this information a cooperative MSS mechanism is initiated in the scope of the fragment, i.e, nodes whose current predictions overlap the unassigned part of the segment exchange their predictions. Similarly, when a node's prediction indicates that it can cover an unassigned part of the segment or if a new node enters the segment, it broadcasts its current prediction, its fragment, and its neighboring fragments to its direct neighbors to initiate the MSS mechanism.

When a node fails, it typically cannot initiate an MSS beforehand to let other nodes take over sensing. Therefore, an optional mechanism for monitoring nodes based on periodic progress reports is proposed. Such a progress report includes the position and the fragment associated with the node. It is broadcasted to the direct neighbors. A node monitors those nodes whose fragments overlap with its own. Based on the predicted current position of the monitored node, it determines its probability to be in communication range. If it has a high probability to be in range, and if it fails to receive a progress report it assumes the other node to have failed. In that case, it initiates a cooperative node selection.

5.3.4 Resolution Shaper

The purpose of the resolution shaper is to adjust the physical sampling interval such that r_{range} is respected. Basically, the resolution shaper needs to omit sensing if its position is within an area where it already acquired measurements. Moreover, it needs to adjust the physical sampling interval according to its speed v in a way that it covers a distance of $2 \cdot r_{\text{range}}$ between two measurements. For this purpose, a node computes its sampling interval as $2 \cdot r_{\text{range}} / v$. Nodes adjust this interval with every update of their positioning system. To avoid coverage holes, which might result when nodes accelerate, the maximum node speed can be used instead of the current node speed.

Since positioning has a high energy consumption, it is essential to deactivate it if not needed. In the sensing area, between start and end point of sensing, a node needs continuous positioning. However, outside that area, it may deactivate positioning. The positioning interval is set according to the distance a node may move before it enters its sensing area or before it enters a different segment. Based on this distance and the maximum speed, a node can compute the time to deactivate positioning (cf. Section 5.3.3).

5.4 Algorithms for Segment Sensors Supporting Object Detection

In addition to providing maps of environmental conditions such as noise through virtual sensors, the concept of virtual sensors (cf. Section 5.1) can also be applied to the detection of mobile objects using Bluetooth or RFID technology. In essence, a virtual segment sensor serves as a data-centric abstraction to access object detection events originating along a specific road segment. Main parts of this section were previously published in [WWDR11].

In particular, a virtual sensor V shall report mobile objects detected within $area(V)$ during the detection phase V_δ. A virtual sensor might be configured to report only the detection of specific objects. In addition to the ID of a detected object, it reports the position, and the time of the detection. Due to uncontrolled node mobility, no guarantees about coverage and, therefore, completeness of the search for mobile objects are possible. If an object is not detected, it might be because it is not in the area of V. However, it might also be that the coverage of the virtual sensor was incomplete, and the object was simply missed. In such a case, the search results in *false negatives*.

The spatio-temporal coverage metric (cf. Section 5.4.1) is developed in order to determine the quality of the search, and as an indicator for the probability of false negatives. It integrates spatial and temporal coverage, suiting scenarios where spatial and temporal coverage cannot be considered independent within the spatial and temporal bounds of a virtual reading.

Given the number of detected objects during the search period and the detection ratio of sensing, an application could easily determine the overall number of objects by dividing the number of detected objects by the detection ratio. However, since the detection ratio is unknown, it is substituted by the achieved coverage. Therefore, the coverage metric is designed such that it closely models the detection ratio by incorporating information about the object speed distribution. Mechanisms to monitor and determine the object speed distribution are presented in Section 5.4.6.

Coordination algorithms achieve the coverage-aware automated detection of mobile objects in urban areas. A straightforward approach would be to let nodes sense continuously while they are on a segment that needs to be searched for objects. On leaving the segment or when the search ends, they transmit their measurements to the PSS. This server collects the measurements from all nodes, and computes their coverage. Basically, such an *isolated* approach where nodes sense independently achieves the highest coverage and, thus, maximizes the probability to detect mobile objects. However, it is likely to produce redundant measurements with high node density.

The main goal is to achieve the coverage of the isolated approach, while minimizing energy consumption of the battery powered mobile nodes. To achieve this goal, a centralized and a distributed algorithm are proposed that select and coordinate nodes such that only essential nodes participate in sensing. The central coordination algorithm is based on a central instance that coordinates sensing of mobile nodes (cf. Section 5.4.3). The distributed coordination algorithm is based on the distributed coordination of nodes on a segment (cf. Section 5.4.4). To further reduce sensing, the sensing strategies in Section 5.4.5 allow, in contrast to continuous sensing, to reduce the sampling rate of sensing.

5.4.1 Spatio-Temporal Coverage

The coverage metric defined in this section serves two purposes. First, it allows to define sensing goals for the coordination of sensing. Second, it allows to estimate the detection ratio of sensing and, thus, to estimate the overall number of objects on a segment. Due to the mobility of objects, the coverage metric must consider the temporal aspect in addition to the spatial aspect of sensing. This is in contrast to spatial coverage metrics used in Section 5.3.1.

In essence, a spatial area is said to be spatio-temporal covered during a time period if every phenomena, i.e., object, that is within this area at any point in time during this period is detected. More general, a spatio-temporal area is said to be covered if every object that is within this area is detected at some point in time. This is in analogy to the spatial coverage, where an area is said to be spatial covered if every phenomena within this area is detected. The spatial coverage of measurements depends on sensor characteristics. In contrast, the spatio-temporal coverage depends on sensor characteristics and characteristics of the observed phenomena. In the following, a unit-disc sensor model is used to model sensor characteristics, and phenomena are assumed to be mobile objects.

The remainder of this section is structured as follows. First, the spatio-temporal coverage of a single measurement is explained and it is shown how it depends on the assumed object speed. Afterwards, the coverage of multiple measurements from possibly different nodes is analyzed. Then, detailed information about the object speed distribution are incorporated to derive a realistic coverage that closely relates to the detection ratio of sensing. Finally, the spatio-temporal coverage metrics is defined.

Coverage of a Single Measurement

Due to the continuous movement of mobile objects, a single measurement at a certain point in time also covers a certain area after and also before the moment of sensing. Figure 5.12a shows a diagram, where the x-axis shows the one-dimensional spatial position on a segment, while the t-axis represents time. An object moving towards the positive x-axes, i.e., moving at a positive speed value, is said to move from left to right, and an object moving in the opposite direction, i.e., moving at a negative speed value, is said to move from right to left.

According to the unit-disc sensor model, a measurement acquired at position x_{read} at time t_{read} spatially covers the surrounding up to a distance of r_{range}, i.e., it covers the interval $I = [x_{\text{read}} - r_{\text{range}}, x_{\text{read}} + r_{\text{range}}]$. If an object is detected with this measurement it is within I. Otherwise, is is outside of I.

To derive the spatial intervals that are covered by this measurement at other points in time, a model of the object movement is required. Assuming the object speed to be limited by v_{maxo} (worst case), i.e., objects may move in any direction with at most v_{maxo}, the covered area is a shape as depicted in Figure 5.12a. Any object within this area is known to be within I at the time of the measurement and, thus, detected with the measurement.

To show that this is the case for the points in time after the measurement, consider a mobile object moving at a speed of v_{maxo} that is just outside I at the time of the measurement. At time $t_2 = t_{\text{read}} + r_{\text{range}} / v_{\text{maxo}}$ the object reaches x_{read}. Since the object cannot exceed a speed of v_{maxo}, it cannot enter the covered area if it was not already within I at the time of the measurement. Put differently, any node within the future covered area must have been within I at the time of the measurement. Otherwise, it would have exceeded the maximum speed.

Similarly to the covered triangle after t_{read}, a measurement also covers a triangle before t_{read}. To clarify this, assume a mobile object at x_{read} at time $t_1 = t_{\text{read}} - r_{\text{range}} / v_{\text{maxo}}$. This object needs to move with maximum speed v_{maxo} to leave I before the measurement occurs and it would be detected. In essence, every object within the past covered area is detected by the measurement since it cannot exceed the maximum speed and, thus, cannot escape the measurement.

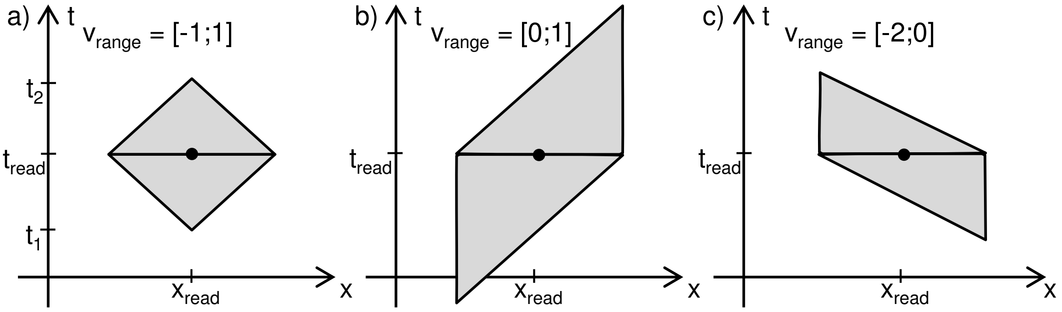

Figure 5.12: Coverage of a single measurement

In contrast to Figure 5.12a, which shows the coverage of a measurement assuming that objects may move at maximum speed in any direction, Figure 5.12b and c show the coverage of a measurement for the case that objects may only move in one direction, either to the left or to the right. In Figure 5.12b the lower and upper bounds for the object speed are [0;1], and in Figure 5.12c the speed range v_{range} is [-2;0]. As this example shows, the coverage of a measurement heavily depends on the model of the object movement.

In the following, an analytic model that describes the covered area of a measurement is derived based on bounds for the mobile object speed. Without loss of generality, the coordinates where the measurement is taken are $(0,0)$ and the object speed is bound to the range $v_{\text{range}} = [v_{\text{lower}}, v_{\text{upper}}]$ with $v_{\text{lower}} < v_{\text{upper}}$ and $|v_{\text{lower}}|, |v_{\text{upper}}| \in [0,\ v_{\text{maxo}}]$. No spatial bounds are considered. A trajectory of an object is said to be valid if it is continuous and if the object speed $v(t)$ is in v_{range} for each point in time.

The future coverage of the measurement can be computed using the intersection point of two straights. First, s_{lower} which intersects $(r_{\text{range}}, 0)$ and has a slope of $1/v_{\text{lower}}$, and s_{upper} which intersects $(-r_{\text{range}}, 0)$ and has a slope of $1/v_{\text{upper}}$.

$$s_{\text{lower}} : \begin{cases} y = \frac{1}{v_{\text{lower}}} \cdot x - \frac{r_{\text{range}}}{v_{\text{lower}}} & \text{if } v_{\text{lower}} \neq 0, \\ x = r_{\text{range}} & \text{else.} \end{cases} \tag{5.9}$$

$$s_{\text{upper}} : \begin{cases} y = \frac{1}{v_{\text{upper}}} \cdot x + \frac{r_{\text{range}}}{v_{\text{upper}}} & \text{if } v_{\text{upper}} \neq 0, \\ x = -r_{\text{range}} & \text{else.} \end{cases} \tag{5.10}$$

If $v_{\text{lower}} \neq v_{\text{upper}}$, these straights intersect at point P_{future}. Otherwise, there is no intersection. The point P_{past} is computed analogously.

$$P_{\text{future}} : \left(\frac{r_{\text{range}} \cdot \left(v_{\text{upper}} + v_{\text{lower}}\right)}{v_{\text{upper}} - v_{\text{lower}}}, \frac{2 \cdot r_{\text{range}}}{v_{\text{upper}} - v_{\text{lower}}} \right) \tag{5.11}$$

$$P_{\text{past}} : \left(\frac{-r_{\text{range}} \cdot \left(v_{\text{upper}} + v_{\text{lower}}\right)}{v_{\text{upper}} - v_{\text{lower}}}, \frac{-2 \cdot r_{\text{range}}}{v_{\text{upper}} - v_{\text{lower}}} \right) \tag{5.12}$$

The covered area A_{covered} of a measurement is a polygon, i.e., the union of the future and past covered triangles.

$$A_{\text{covered}} : \text{polygon}\left(P_{\text{future}}, (r_{\text{range}}, 0), P_{\text{past}}, (-r_{\text{range}}, 0)\right) \tag{5.13}$$

From this follows the size of the covered area A_{covered}:

$$|A_{\text{covered}}| = \frac{4 \cdot r_{\text{range}}^2}{v_{\text{upper}} - v_{\text{lower}}} \tag{5.14}$$

Two main characteristics of this equation are worth mentioning. First, the size of the covered area depends on the size of the speed range $v_{\text{upper}} - v_{\text{lower}}$. The smaller this range, the larger the covered area. If $v_{\text{lower}} = v_{\text{upper}}$, the covered area has an infinite size. Intuitively, if objects that move with constant speed are detected, their position can be determined for any point in time by extrapolation. Second, the size of the covered area is independent from the actual values of the object speed range. In essence, the spatio-temporal area covered by a measurement differs greatly between a speed range of [0,1] and a speed range of [-6, -5]. However, in both cases, the size of the covered area is $4 \cdot r_{\text{range}}^2$. The effect of bounded spatial areas is discussed in the following section.

Coverage of Multiple Measurements

The considerations of the previous section can be easily generalized to determine the covered area of multiple readings. As shown in the previous section, a point of a spatio-temporal area is covered if all valid object trajectories that pass through this point also pass the point of a measurement x_{read} at the time of the measurement t_{read}. Transitively, a point is covered if all valid object trajectories that pass through this point also pass a covered area.

Therefore, measurements with overlapping coverage, may lead to covered areas in addition to the union of the individual covered areas of the measurements. Figure 5.13a and b illustrate this phenomenon. Transitively covered areas are marked with gray.

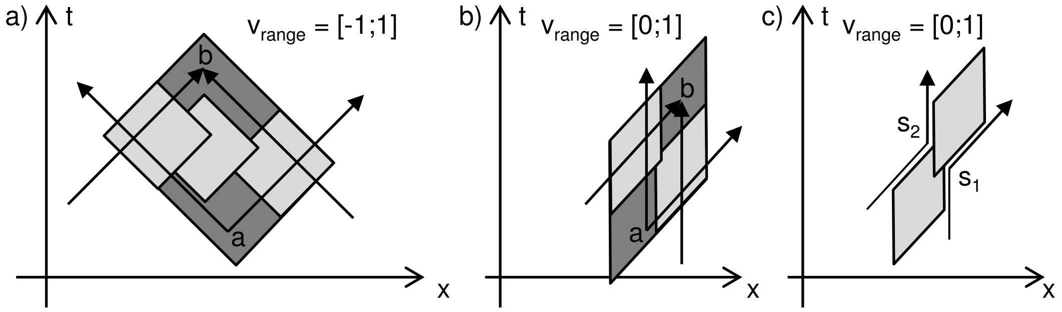

Figure 5.13: Coverage of multiple measurements

To understand why the lower gray area is covered, consider an object at position a. Every possible trajectory from this location intersects a covered area, i.e., the mobile object is detected by the measurement of that area. Analogously, the upper gray area can be explained. Any mobile object that reaches b must have passed through at least one covered area. This is illustrated by the two trajectories that lead to b.

In contrast, Figure 5.13c depicts two overlapping covered areas that do not lead to transitively covered areas. In this example, the trajectories s_1 and s_2 of mobile objects follow the boundary of the covered areas. Since these do not violate the object speed range, it is clear that no transitive coverage is achieved in this example.

This principle can be generalized to the case of sensing of multiple nodes. For instance, Figure 5.14a shows the trajectory and the covered area of two nodes within the bound-

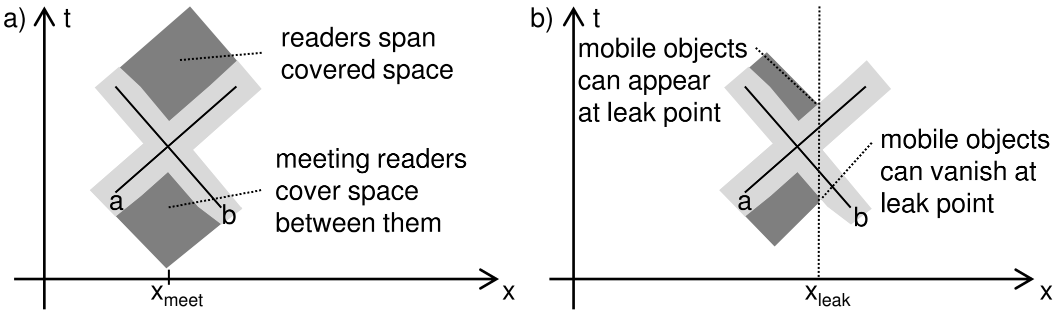
a)
t
readers span covered space
meeting readers cover space between them
a
b
x
x_{meet}
b)
t
mobile objects can appear at leak point
mobile objects can vanish at leak point
a
b
x_{leak}
x

Figure 5.14: Coverage of multiple nodes

aries of a segment. For simplicity, the figure shows the coverage achieved by continuous sensing. In this example, the object speed range is $[-v_{\text{maxo}}, v_{\text{maxo}}]$.

Intuitively, two nodes a, b that move towards each other cover the space between them and, as soon as they meet at time t_{meet}, it can be deduced that the space between them is covered since no object can pass a node undetected. Furthermore, if these nodes move in opposite directions after the meeting, it can be deduced that the space between them is covered, as well. Figure 5.14b shows the effect of a leak point, e.g., a crossing where mobile objects enter or leave a segment, on the coverage.

In essence, the coverage principles discussed in this section can be unified to define the v_{range}-coverage of a spatio-temporal point.

Definition 4. *A point P of the tx-plane is v_{range}-covered if all valid object trajectories passing P pass at some point in time at least one node within sensor range r_{range} at the moment it acquires a measurement.*

Coverage Depending on Object Speed Distribution

So far, the v_{range}-coverage of measurements was investigated. Knowing for instance the maximum speed of objects ($v_{\text{range}} = [-v_{\text{maxo}}, v_{\text{maxo}}]$) and having a set of measurements, the v_{range}-coverage can be used to determine the spatio-temporal area where no mobile object can be without being detected by at least one of the measurements. However, if not all objects move at maximum speed, the v_{range}-coverage is a pessimistic metric, i.e., the fraction of v_{range}-covered area might be lower than the actual observed area.

Applications such as the traffic density estimation require a coverage metric where the difference between the reported covered area and the actual observed area is as small as possible to allow for extrapolation of partial observations. Therefore, this section describes how detailed information about the object speed distribution can be used to provide such a coverage metric.

The basic idea is to divide the set of observed phenomena into classes with similar movement characteristics, to determine the individual coverage for each class of phenomena and, finally, to aggregate the partial results.

The speed distribution of mobile objects is seldom known in detail in a real world scenario. Often only partial information about the distribution is available. For instance, road traffic in a city might by modeled as half the cars moving in one direction (v_{range} = [0,50]), and the other half moving in the opposite direction (v_{range} = [-50,0]). In general, a partial discrete speed distribution function $f_v : (\mathbb{R}, \mathbb{R}) \rightarrow [0;1]$ can be used to model object movement.

$$f_v(v_{\text{lower}}, v_{\text{upper}}) = p \tag{5.15}$$

In this function, p is the ratio of objects bound to a speed range of $[v_{\text{lower}}, v_{\text{upper}}]$. Using this function, the v_{range}-coverage is computed for every speed range defined by this function. Each covered area is assigned the respective p value as a weight.

Figure 5.15 illustrates the weighted covered areas of a measurement and the corresponding partial discrete speed distribution function. In this example, four ranges are mapped to a weight. For instance, half of the objects are bound to a speed range of [-1,0], and a fifth of the nodes are bound to a range of [1,2]. For each of these ranges, the figure shows on the left hand side the four respective covered areas.

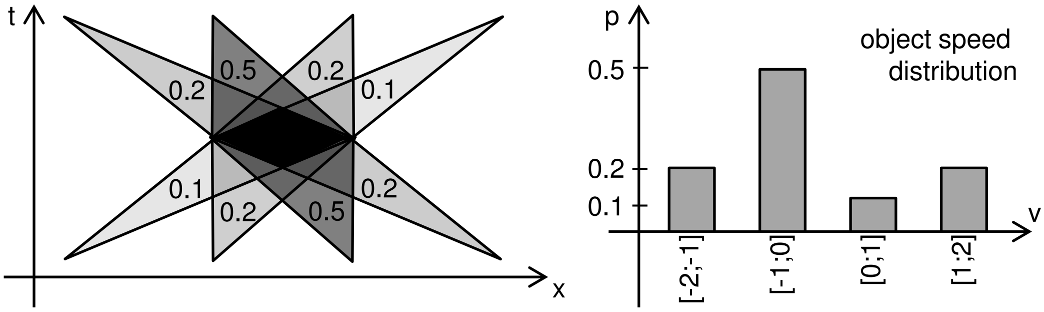

Figure 5.15: Weighted coverage of a single measurement

The shades of gray indicate the weight of the covered areas. Where these areas overlap, the shades are darker. This indicates that overlapping areas are weighted with the sum of the individual weights of the areas that overlap. For instance, the black inner parallelogram, i.e. where all four areas overlap, is the area within which all objects are detected, if they move within the speed range [-2;2].

Coverage Metrics

Based on these considerations, the spatio-temporal coverage c_{st} of a segment can be defined as the average coverage during a time period. As a base for this definition, the coverage $c : (\mathbb{R}, \mathbb{R}) \rightarrow [0;1]$ of a spatio-temporal point, and the current coverage $c : \mathbb{R} \rightarrow [0;1]$ for a given point in time are defined.

Definition 5. *The spatio-temporal coverage $c(x,t)$ of a spatio-temporal point is the sum of the weights for all defined speed ranges for which holds (x,t) is v_{range}-covered.*

In essence, this definition generalizes the v_{range}-coverage such that the coverage of measurements can be determined for arbitrary object speed distributions.

Definition 6. *The current coverage $c(t)$ of a segment at a specific point in time t is the integral over $c(x,t)$ normalized by the segment length, i.e., $area(V)$.*

$$c(t) = \frac{1}{|area(V)|} \cdot \int_{area(V)} c(x,t) \cdot dx \tag{5.16}$$

This definition generalizes the spatial coverage as introduced in Section 5.3.1. In contrast to the spatial coverage, which basically only defines coverage for the point in time of a measurement, $c(t)$ defines coverage for any point in time.

Definition 7. *The spatio-temporal coverage c_{st} of a virtual reading R during the time period V_δ is the integral over $c(t)$ normalized by the length of this time span.*

$$c_{st}(V,R) = \frac{1}{V_\delta} \cdot \int_{t=t_0}^{t_0+V_\delta} c(t) \cdot dt \tag{5.17}$$

In essence, c_{st} is evaluated for a specific segment and time period by cutting the polygons on the respective segment and time period and computing their area. The sum of these weighted areas is divided by the the segment length and the time period V_δ.

The following sections introduce algorithms for the coordination of sensing based on the coverage definition introduced in this section.

5.4.2 Coordination Algorithms

Coordination of sensing is essential for efficiency. Using information about the movement of nodes and, thus, their coverage, sensing plans can be devised that allow to fulfill sensing goals while preventing unnecessary redundancy of sensing. By reducing redundancy, energy consuming sensing is limited.

As shown in the previous section, the spatio-temporal coverage of a set of measurements is, in general, larger than the sum of the individual coverage values of these measurements. Thus, the decision whether a node needs to sense at its current position requires to consider the position and sensing state of other nodes as well. This is in contrast to the coordination based on spatial coverage, where a node's sensing decision at a specific position is only influenced by the sensing of other nodes at that position.

Consequently, optimal coordination algorithms need to consider the complete trajectories of nodes during the sensing period. However, the trajectories of nodes are not known a priori. In the following, a pessimistic algorithm is presented that is designed to deactivate sensing of nodes only if sensing does not add to the already achieved coverage. Afterwards, an optimistic algorithm is presented that relies on movement predictions to deactivate sensing of nodes, i.e., sensing is deactivated if it does not add to the coverage that will be achieved according to the predicted trajectories of nodes.

The pessimistic algorithm, Centralized Coordination Algorithm (cf. Section 5.4.3), is designed as an algorithm with a central coordinator that relies with a global view. In contrast, the optimistic algorithm, Distributed Coordination Algorithm (Section 5.4.4), is a distributed algorithm where nodes only have partial views.

5.4.3 Centralized Coordination Algorithm

The basic idea of this algorithm is to track the position and the sensing state of mobile nodes at a central instance, i.e, the *Public Sensing Server* (PSS). The PSS selects and activates nodes based on their current state and the sensing state of their neighbors on the segment. If the trajectories of nodes would be known in advance, an algorithm could decide along which parts of these trajectories to sense to achieve maximum coverage at lowest cost. However, due to uncontrollable node movement, it is impossible to prevent all redundancy in an online algorithm without risking to lose coverage. This section presents an algorithm that focuses on maximizing the coverage. It is a pessimistic approach where nodes only deactivate sensing if it is within an area that is $[-v_{\text{maxo}}, v_{\text{maxo}}]$-covered. First, node states and the corresponding state transitions that occur at meeting points of nodes that allow for identifying nodes in covered areas are introduced. Then, an efficient update protocol allows to collect the necessary position information to detect meetings of nodes.

In the following, a position of a node on a specific segment is represented by a 1D coordinate relative to the start point of the segment. A position P_L is left of P_R and P_R is right of P_L, if $P_L < P_R$. The following node states are distinguished.

RL-active: The node is sensing independently, i.e., it does not span a covered area with another node. This is the initial state of each node as it enters a segment.

L-active: An L-active node extends the left side of a coverage polygon by sensing. A node is in this state if it spans a covered area with another node.

R-active: An R-active node extends the right side of a coverage polygon by sensing. Similarly to L-active nodes, a node in this state spans a covered area with another node.

inactive: The node is inactive. Sensing is deactivated.

State transitions occur at meetings of nodes based on node states. Table 5.1 shows all node combinations at meeting points, and their state transitions. The first column shows the state of the node from the left and the top row shows the state of the node from the right. The fields of the table show the new states of the nodes. The first state is that of the node from the left and the second state is that of the node from the right. Combinations of states that cannot occur in a real scenario are marked accordingly (-).

As an example consider the scenarios in Figure 5.16. In Figure 5.16a, nodes a and b are RL-active. When they meet at (a,b), they change their states: a to R-active and b to L-active. From the meeting point, these nodes span the covered area between them. Figure 5.16b shows the case where an RL-active node c from the left meets the L-active node b. After meeting at (c,b), node c enters a covered area and transits to inactive, i.e., it deactivates sensing. Node b remains L-active. When c leaves the covered area at (c,a), it gets R-active. At the same time, node a is in a covered area and transits to inactive.

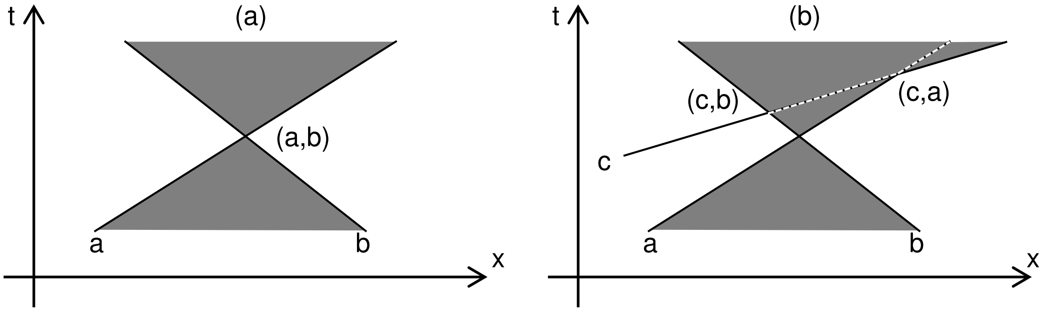

Figure 5.16: Node state transitions

Coordination reduces sensing of nodes. Additionally, reducing positioning and communication of position fixes to the PSS saves energy. To track node states, the server needs to be informed about their positions. Since it only needs to know where and when nodes

Table 5.1: Node state transitions

	RL-active	**L-active**	**R-active**	**inactive**
RL-active	R-active L-active	inactive L-active	-	-
L-active	-	-	R-active L-active	inactive L-active
R-active	R-active inactive	inactive inactive	-	-
inactive	-	-	R-active inactive	inactive inactive

meet, their position update frequency can be optimized accordingly.

The idea of this optimization is that nodes may deviate from their predicted movement without updating as long as they move within predefined border lines in the tx-plane to direct neighbors. Assuming constant node speed, the server sets the borders as bisecting lines of the predicted trajectories of neighbors. In addition, segment boundaries are borders. A node sends an update to the PSS as soon as it reaches a border line. It may only defer the update if the corresponding neighbor node could not have reached its location at maximum speed v_{maxn}.

Figure 5.17 shows on the left hand side two nodes and their extrapolated trajectories based on their current speed. The border line of these nodes is dashed. For each of the nodes its maximum speed trajectory is depicted as pointed line. Figure 5.17 depicts on the right hand side the two cases when an update needs to be performed. In the first case, *b* updates as soon as it crosses the border. In the second case, *a* can defer its update until it reaches the maximum speed line of *b*.

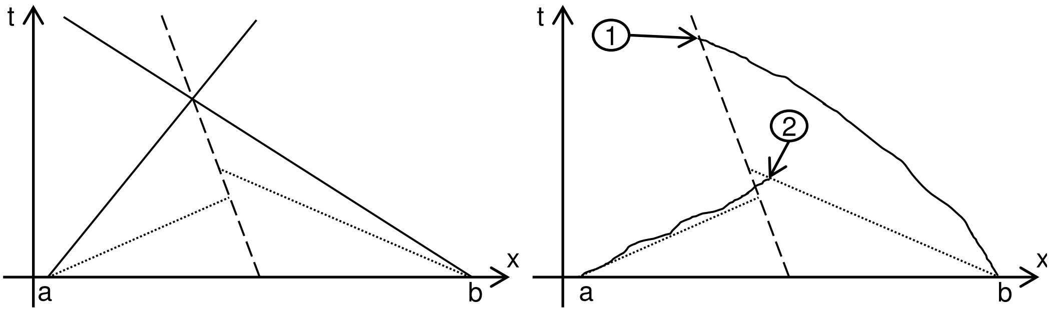

Figure 5.17: Deactivation of positioning

A node locally determines its positioning rate according to a worst case prediction, i.e., it computes the time until it reaches the closest border at maximum speed v_{maxn}. Note that sensing may have stricter requirements on positioning (cf. Section 5.4.5).

On receiving an update, the server computes a new border line and notifies the neighbors accordingly. If it detects a meeting of neighbors, it updates their neighbor relations and their state transitions. Note that inactive nodes do not need to synchronize with other inactive nodes.

5.4.4 Distributed Coordination Algorithm

Although the pessimistic centralized algorithm is effective, it does not exploit energy efficient local communication over WiFi, which is feasible since coordination only requires communication between neighbor nodes. Therefore, the distributed algorithm is based on node cooperation in the area of a segment in an ad-hoc network. In contrast to the centralized algorithm, the distributed algorithm follows an *optimistic* approach where sensing is deactivated based on coverage predictions. In case of deviations from the predicted movement, coverage may be lost. In this approach, nodes locally decide where they need to sense based on local knowledge about the predicted movement and coverage of neighbors. A major design principle of the distributed algorithm is that nodes deactivate sensing if other nodes explicitly commit to take over sensing for them. Thus, loss of coverage, compared to the centralized algorithm, may occur when nodes deviate from their predicted movement.

The basic idea of the algorithm is that nodes manage a local view of the node movement on their segment. To achieve this, they broadcast their position and speed in the ad-hoc network if their actual movement deviates from the predicted by more than a threshold. Moreover, nodes notify their neighbors when they enter or leave a segment. Direct neighbors receive these updates and add them to their local view. To further distribute these updates, nodes include the received updates into their own update messages. Receiving nodes update their local view.

Based on the local view of node movement on a segment, each node locally predicts the coverage of these nodes. If a node is essential to achieve the predicted coverage, it starts sensing. Otherwise, it deactivates sensing to save energy. If an area can be covered by multiple nodes, rules are required to select the sensing node in order to avoid redundant measurements or coverage holes. Thus, an absolute ordering of nodes according to the node ID is used. Algorithm 5.5 lists the algorithm in detail.

Algorithm 5.5 Distributed Coordination Algorithm

```
Require: $\mathbb{L}$ ordered list of nodes
  $\mathbb{C} \leftarrow \emptyset$    // init set of candidate nodes
  $\mathbb{S} \leftarrow \emptyset$    // init set of selected nodes
  for all $N \in \mathbb{L}$ do
    if $|\text{coverage}(\mathbb{C} \cup N)| > |\text{coverage}(\mathbb{C})|$ then
      $\mathbb{C} \leftarrow \mathbb{C} \cup N$
    end if
  end for
  for all $N \in \mathbb{C}$ do
    remain $\leftarrow$ coverage$(\mathbb{C} \setminus N)$
    if pos$(N) \notin$ remain then
      $\mathbb{S} \leftarrow \mathbb{S} \cup N$
    end if
  end for
```

In the first step, the algorithm iterates over all nodes on the segment and checks incrementally if they extend the coverage. If so, they are added to the list of candidates. Otherwise, they do not need to sense. Then, the algorithm checks, for each candidate

N, whether its current position is outside the coverage polygon resulting from the set of candidates without the node itself (coverage($\mathbb{C} \setminus N$)). If N is inside this polygon, it can deactivate sensing. Otherwise, N is added to the set of selected nodes, i.e., it needs to sense. This algorithm is executed on receiving a position update message and on meetings of nodes. As a result of this algorithm, each node knows whether to sense or not.

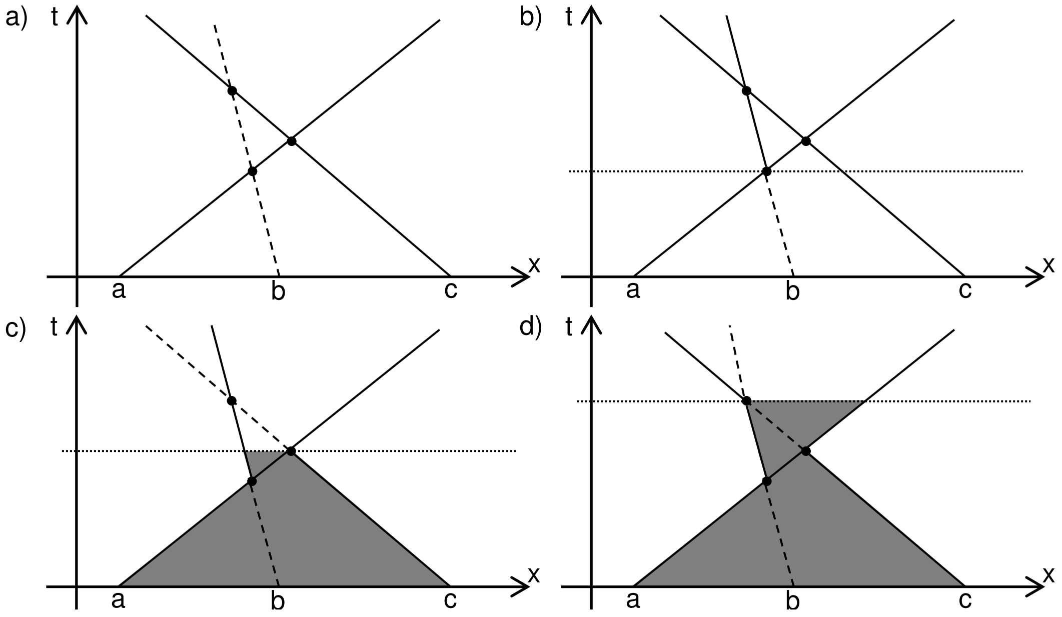

Figure 5.18: Distributed coordination based on predictions

Figure 5.18 shows an example of three nodes at four points in time. Initially, node b can deactivate sensing (marked as dashed line) since it is in the covered area of nodes a and c. When nodes a and b meet, node c leaves the covered area and, thus, starts sensing. Then, when a and c meet, node c enters the covered area and, thus, stops sensing. Finally, in Figure 5.18d, node b can stop sensing, and node c needs to start sensing again.

Deactivating sensing allows for relaxing a node's positioning interval. However, if it enters an uncovered area while positioning is deactivated, coverage is lost. Pessimistically, the time period for deactivating positioning is computed as the time for a mobile node to reach the coverage boundaries with maximum speed v_{maxn}. A more optimistic strategy is to assume a constant movement, where a node continues to move at its current speed. However, if nodes move faster than predicted, coverage might be lost.

5.4.5 Efficient Sensing

The algorithms in the previous sections determine for each node whether to sense or not until the next meeting with another node. To prevent energy consuming continuous sensing of nodes, this section discusses a mechanism to efficiently sense, while achieving coverage comparable to continuous sensing.

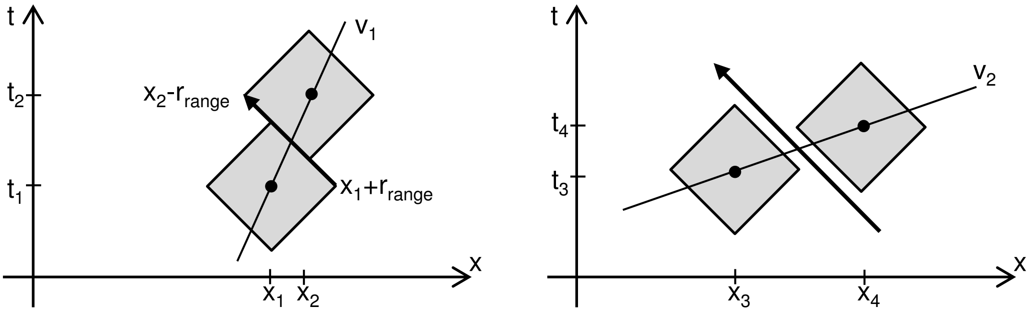

Figure 5.19: Discrete sensing interval based on speed of object and node

Figure 5.19 shows on the left hand side, that a node can determine a discrete sensing interval according to the time for node and object to cover a distance of twice r_{range}. As the arrow indicates, an object moving at maximum speed can only escape two overlapping measurements at t_1 and t_2 if it moves at maximum speed. In order to detect every object within sensing range the worst case coverage is considered. Nevertheless, inaccurate speed predictions may lead to coverage holes, i.e., the coverage of succeeding measurements do not overlap and objects might pass undetected, as depicted on the right hand side of Figure 5.19. Moreover, since sensing is only successful with probability p_{detect}, multiple measurements may be needed to actually detect a passing object.

The goal is to detect mobile objects passing by a sensing node with a required probability. In essence, this allows for trading off sensing quality, i.e., probability to detect a passing object, and energy cost. This leads to the following equation for computing the time interval between sensing.

$$t_{max} = \frac{2r_{\text{range}}}{c_{redundant} \cdot (v_{maxo} + |v_{limit}|)} \tag{5.18}$$

First, a user specifies the required redundancy $c_{\text{redundant}}$ of sensing. This parameter can be chosen according to the success probability of sensing. With p_{detect} close to 1, no redundancy is needed and, therefore, this value can be set to 1. If $p_{\text{detect}} = 0.9$, and measurements are independent, a value $c_{\text{redundant}} = 2$ is sufficient to detect any passing object with a probability of $(p_{\text{detect}})^{c_{\text{redundant}}} = 0.9^2 = 0.99$. Second, a user specifies the speed v_{limit} a node assumed not to exceed. Optimistically, this is the current speed; pessimistically, it is set to v_{maxn}.

5.4.6 Monitoring of Object Speed Distribution

Using only the maximum speed of objects to model their movement, the computed coverage of measurements is only a worst-case estimation. For the coverage to closely model the detection ratio of sensing, detailed information about the object speed distribution is needed. Data about the speed distributions of objects might be available from simulations or real world monitoring. However, if such data is missing, it can also be gathered during sensing. This section presents a mechanism to sample the speed distribution by collecting speed information from the measurements of nodes.

The basic idea of this mechanism is to determine the speed of objects that are detected more than once. Collecting a statistically relevant number of such speed estimations, yield a rough approximation of the object speed distribution. In essence, the speed of an object detected at $P_1 = (x_1, t_1)$ and at $P_2 = (x_2, t_2)$ can be computed as $v = \frac{x_1 - x_2}{t_1 - t_2}$ with $t_1 > t_2$. However, two challenges have to be tackled.

First, the spatial coordinate of the detected object is not know accurately, since it is only known that the object is within sensor range r_{range} of the node that detected it. In addition to this inaccuracy, the position of the node is also only known within some accuracy bound r_{accuracy}. The following equation incorporates these position uncertainties and allows to determine the error of the calculated speed.

$$v_{\text{error}} = \pm \frac{2(r_{\text{range}} + r_{\text{accuracy}})}{t_1 - t_2} \tag{5.19}$$

The second problem is that objects might have multiple possible routes to move from P_1 to P_2. This is especially a problem if the time interval between two detections is large. In essence, the assumption that the object moved on the shortest path between these points might be wrong.

To deal with these issues, the speed of an object is only considered if subsequent detections are within a temporal range $[t_{\text{min}}, t_{\text{max}}]$. Equation 5.19 can be used to determine t_{min}. For instance, to limit the speed error to $\pm 0.5 m/s$ with an overall position uncertainty of $5m$, t_{min} can be set to $20s$. t_{max} can be set to a value of several minutes.

Choosing these values allows to trade the accuracy of the resulting speed distribution for the length of the period needed to collect enough speed samples to determine the speed distribution. Section 5.5.4 presents a more detailed discussion, and shows the applicability of this mechanism to determine the object speed distribution with an accuracy sufficient for modeling the coverage closely to the detection ratio.

5.5 Experimental Setup and Evaluation

This section presents the simulation model followed by the results of extensive simulations of the proposed coordination algorithms. Main parts of this section have been previously published in [WDR09, WDR10a, WWDR11]. The algorithms were implemented for the network simulator ns-2 using the 802.11b extension of ns-2 with the transmission range $r_{\text{tx}} = 100\,m$. The position uncertainty r_{accuracy} is set to $5\,m$ and the sensor range r_{range} is set to $5\,m$ (cf. RFID range [BW08]), as well. According to [Sky11], the duration of a sensing operation, i.e., the time to transmit a measurement, is set to $\delta_{\text{read}} = 20\,\text{ms}$.

Mobile nodes and mobile objects move at pedestrian speed on a street graph according to a graph based mobility model. In the experiments two different mobility generators are used.

CanuMobiSim [SMR05]: A graph-based mobility model, where nodes move on a street graph and their movement is determined by the spatial environment, user travel decisions, and user movement dynamics. For the simulations, a 1000 m x 1000 m section of the city of Stuttgart is used as road graph. By default, the maximum node speed is 3 m/s.

UDEL Models [KSB09]: A graph-based mobility model that is based on surveys from a number of research areas and produces realistic urban traces of pedestrians moving non-uniformly distributed on a street graph. Movement patterns, as in reality, heavily depend on the simulated hour of the day. In the experiments, nodes and mobile objects move at pedestrian speed (between 0.7 m/s and 1.8 m/s) on a nine block section of Chicago consisting of 93 street segments.

Movement predictions are based on the assumption of uniform movement at current speed along the current road segment. Although the simulated sections are relatively small compared to the size of a city, they are sufficiently large for this evaluation since the proposed algorithms apply to local coordination problems of virtual sensors. More important than the size of the service area is the effect of node density which is evaluated in a wide range. Each simulation is performed at least 5 times and lasts 1800 seconds.

The energy model given in Table 5.2 is used to measure the energy consumption of the battery powered mobile nodes. The table also provides references for the different values. The energy to determine a single position fix with GPS [Nav11] is 75 mJ. The energy consumption of the RFID reader is 80 mJ per measurement [Sky11]. The energy for sending and receiving a packet of 1000 Bit at the default broadcast rate of 802.11b at 1Mbps is 2 mJ and 1 mJ, respectively [Sum11]. The energy needed for the communication through a cellular network based on GPRS is 80 mJ for sending and 40 mJ for receiving [GL06].

Based on this model, the energy consumption (EC) metric is the average energy a node consumes per hour. It is computed as the average energy a node spends for positioning,

Table 5.2: Energy Model

Component	Energy [mJ]
GPS [Nav11]	
Position Fix	75
RFID [Sky11]	
Read	80
802.11b at 1 Mbps (broadcast rate) [Sum11]	
Send (1000 Bit)	2
Receive (1000 Bit)	1
GPRS [GL06]	
Send (1000 Bit)	80
Receive (1000 Bit)	40

communication, and sensing. By default, a high energy sensor (RFID reader) is considered. Since positioning is required while sensing, even usage of low power sensors would result in a similar energy consumption.

In the following, the evaluation results regarding the algorithms for coordinating sensing at specific points are presented in Section 5.5.1. Then, the performance of the coordination algorithms for virtual sensors along road segments is shown in Section 5.5.2. Afterwards, the performance of the algorithms for the coordinated detection of objects is shown and discussed in Section 5.5.3. Finally, the results for the object number estimation are shown in Section 5.5.4.

5.5.1 Results for Point Sensor Algorithms

The following implementations are compared in this section:

EROA/P: This implementation manages update times proactively with EROA/P (cf. Section 5.2.2), detects proximity to virtual sensors with DROA (cf. Section 5.2.3) and updates virtual sensors efficiently with CROA (cf. Section 5.2.4).

EROA/R: In contrast to the EROA/P implementation, the EROA/R implementation manages update times reactively (cf. Section 5.2.2) instead of proactively.

MEROA/R: In essence, MEROA/R performs as EROA/R. However, instead of cellular networks, a WMN connects the nodes to the PSS.

AEROA/P: In addition to EROA/P, AEROA/P performs adaptation as described in Section 5.2.5. The $delay_{\mathrm{th}}$ is set to 20% of V_s. The average time a node stays in the service area is set to 15 min. The numbers of nodes leaving and entering the service area are equal.

Isolated: A simple isolated approach where nodes assume an update interval of ten seconds. They sense independent from other nodes. Moreover, nodes implement the DROA algorithm (cf. Section 5.2.3). This implementation presents the worst case for duplicates.

Global: The Global approach is implemented to show the best case where nodes access global knowledge to perform only the necessary updates.

For the simulations in this section, mobility traces generated by *CanuMobiSim* are used, since it allows to generate traces with high node densities. This is essential for investigating the performance of the algorithms in sparse and dense urban scenarios. By default, 25 virtual sensors are randomly distributed on the roads of the service area.

Effectiveness

In order to show that the algorithms achieve the desired update frequency, the update validity (UV) metric is evaluated. In essence, it is defined as the average temporal coverage c_t (cf. Section 5.2.1) of all virtual sensors in the simulation.

Using this metric, the effectiveness of the approaches can be compared with the Global approach, which exploits global knowledge, and, therefore, does not miss any updates. Since Isolated is as effective as Global, it is not considered in this section. Moreover, since MEROA/R is as effective as EROA/R, it is also omitted in this section.

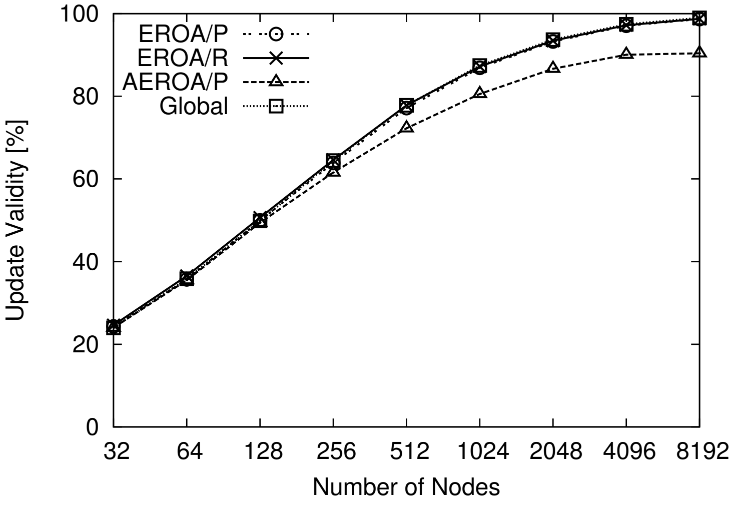

Figure 5.20: Update validity for different numbers of nodes

Figure 5.20 depicts the update validity for experiments with varying number of nodes. V_s is reduced to 1 min to evaluate the effectiveness in case of stressing conditions. The figure shows that no updates are missed, since EROA/P and EROA/R perform as well as the Global approach. The AEROA/P approach performs almost equally well up to the scenario with 256 nodes. Here, the adaptation mechanism starts reducing the number of nodes assigned to individual virtual sensors. Therefore, updates may be delayed, as specified by $delay_{\text{th}}$, and the update validity slightly degrades.

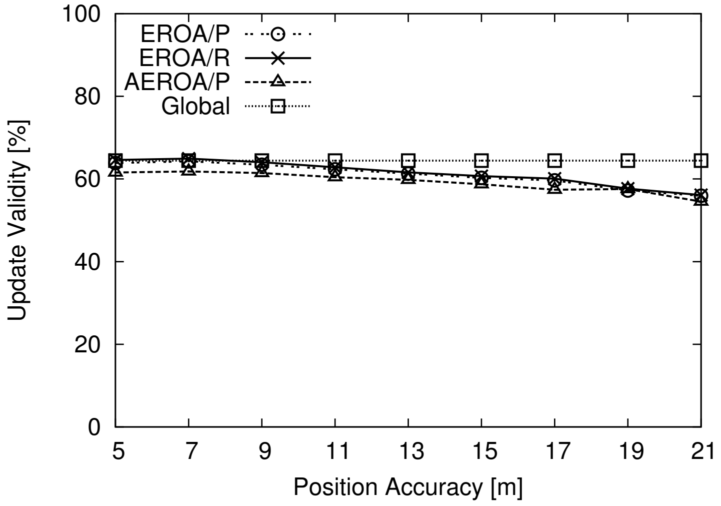

Figure 5.21: Update validity for different values of position accuracy r_{accuracy}

The relation between the update validity and the position accuracy is studied with a series of simulations with r_{accuracy} ranging from 5m to 21m. Figure 5.21 shows that AEROA/P performs slightly worse than the other approaches due to the $delay_{\text{th}}$ as explained above. Moreover, Figure 5.21 shows a slight decrease of the update validity, which indicates that some updates are missed. Due to the large inaccuracy, the probability p_{success} of a node to successfully read a virtual sensor is limited. However, nodes may skip sensing if p_{success} is below a threshold, a trade-off between energy savings and actual read success probability.

With growing node speed, the frequency of nodes passing by a virtual sensor is increased. This leads to an increase of the update validity as depicted in Figure 5.22. EROA/P and EROA/R do not miss updates and behave like the Global approach. AEROA/P performs slightly worse as specified by $delay_{\text{th}}$.

Efficiency

The proposed coordination algorithms reduce the energy spent by mobile nodes by reducing the number of read operations, communication and for position fixes. This sec-

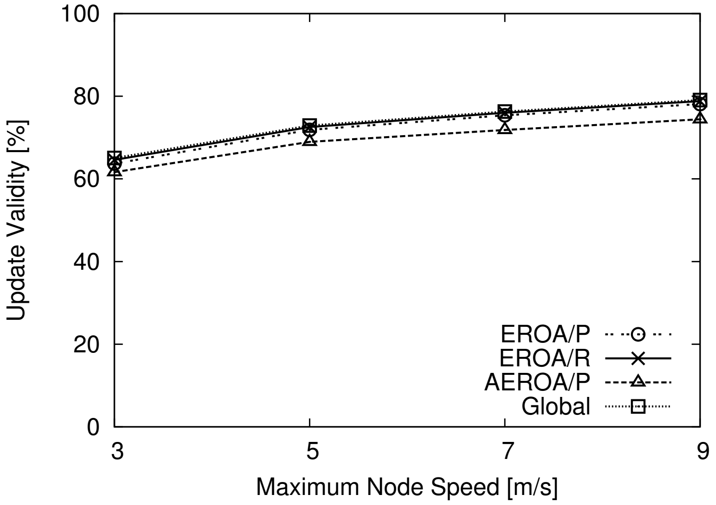

Figure 5.22: Update validity for different values of maximum speed v_{maxn}

tion shows the results of the quantitative evaluation of the achieved energy savings. Since Global is only meant to analyze the best case in terms of effectiveness, it does not produce results in terms of efficiency. Thus, it is omitted in this section.

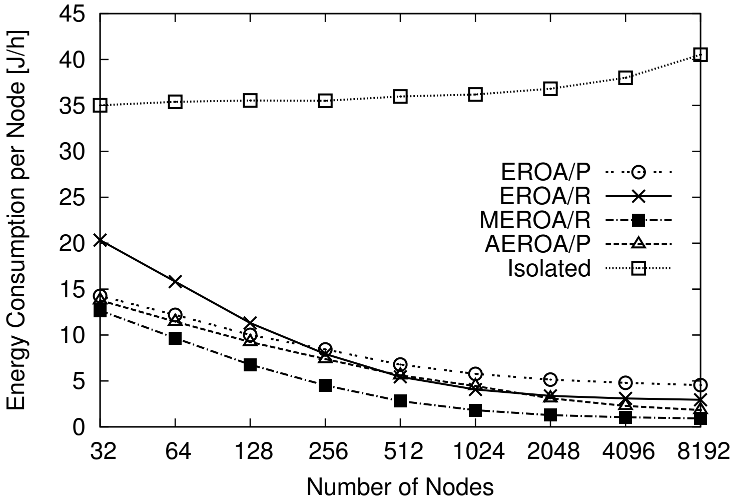

Figure 5.23: Energy consumption for different numbers of nodes

In Figure 5.23 the energy consumption is depicted for scenarios with different numbers of nodes. The Isolated approach shows the expected constant energy consumption for small numbers of nodes. However, the higher the number of nodes, the higher is the energy consumption, due to the increasing number of collisions. The EROA/P approach profits from growing numbers of nodes, because the load is distributed over a larger set of nodes. The value for 8192 nodes is only about 33% the value for 32 nodes. The adaptive AEROA/P approach benefits even more from high node densities. It only consumes about 15% of the energy in case of 8192 nodes compared to 32 nodes. This is because of the adaptive reduction of nodes that participate in updating a virtual sensor. The EROA/R approach benefits from higher node density, because the time between notification and update is reduced. After an update, the node can ignore the virtual sensor again. A similar behavior shows MEROA/R. However, since communication cost are much lower using WiFi compared to the cellular radio, it achieves the best performance.

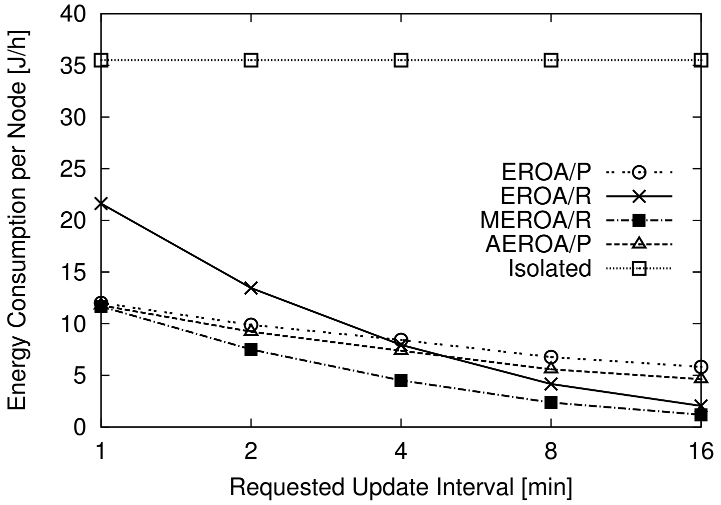

Figure 5.24: Energy consumption for different values of requested update interval V_s

Figure 5.24 depicts the energy consumption with growing requested update interval V_s. The values for the reactive and the proactive EROA approaches are roughly equal. For lower values of V_s, the EROA/P and AEROA/P approaches have a lower energy consumption than EROA/R because of the high energy consumption of GPRS, which is used to notify nodes. The energy consumption of EROA/R scales with the number of updates, while that of EROA/P depends on the length of V_s (cf. Section 5.2.2). Due to the low communication cost, MEROA/R achieves the lowest energy consumption.

The energy consumption of a node also depends on the position accuracy r_{accuracy}. Figure 5.25 shows that the average energy consumption increases when r_{accuracy} is incremented from 5m to 21m. This is due to the increased number of measurements outside sensor range. The increase is higher for the Isolated approach because it performs more updates and, thus, the impact of sensing on the energy consumption is higher compared to the other approaches.

The overall number of updates grows with the number of virtual sensors in the service area. Since the energy consumption of EROA/R and MEROA/R scales with the number

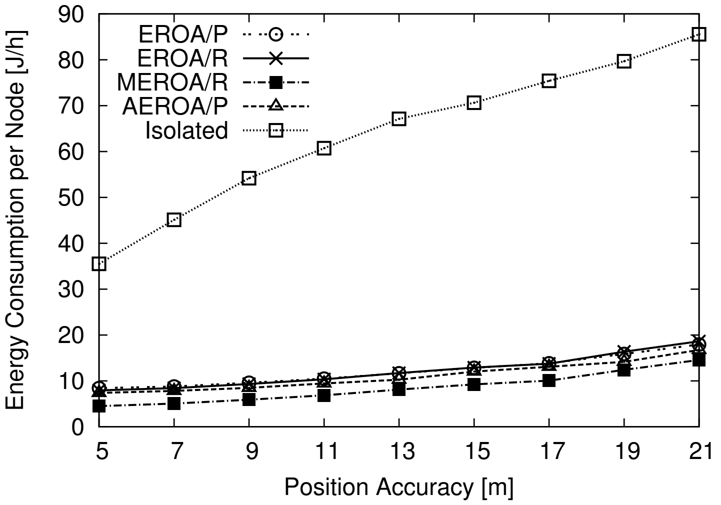

Figure 5.25: Energy consumption in relation to the position accuracy r_{accuracy}

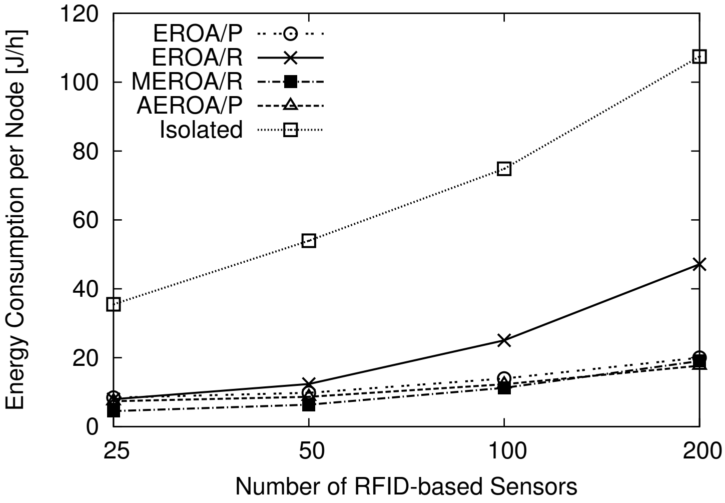

Figure 5.26: Energy consumption for different numbers of virtual sensors

of updates, it shows an increase with growing number of virtual sensors as depicted in Figure 5.26. For the EROA/P and AEROA/P approaches the energy consumption increases much slower.

The energy consumption of the EROA/R algorithm is independent of V_s when considering the energy consumption per update. However, it depends on the node density. With high node density, an update can be performed shortly after the notification, which allows nodes to ignore the respective virtual sensor again. The EROA/P shows a similar decrease of the energy consumption. However, the efficiency of EROA/P degrades with increasing V_s. The different characteristics between EROA/P and EROA/R explain the point of balance between the energy consumption of the two approaches. In the evaluated scenario, with 256 nodes in the networks, it is at $V_s = 4$ min. AEROA/P, which allows for delayed updates, proved to reduce the energy consumption by up to 50% in the evaluated scenarios.

The most significant energy consumer is positioning. Mechanisms such as dead reckoning or map matching could reduce the overall energy consumption. However, these optimizations work independently of the proposed algorithms and, therefore, are outside the scope of this dissertation.

Next, the efficiency of the algorithms in terms of duplicate updates is evaluated. In addition to Global, MEROA/R is omitted in the following, since it achieves the same number of duplicates as EROA/R.

An update is valid for the time V_s. The time δ_{valid} for which the PSS has valid updates during a simulation run is measured. The number of updates U_{measure}, which the mobile nodes performed in order to provide the PSS with valid updates for this time span is measured as well. The minimum number of updates to provide valid updates for the same timespan is computed as $\delta_{\text{valid}}/V_s$. Based on this value, the percentage of duplicates (POD) is defined as follows:

$$\text{POD} = 1 - \frac{\delta_{\text{valid}}/V_s}{U_{\text{measure}}} \tag{5.20}$$

At first, the percentage of duplicates is plotted in Figure 5.27 for different numbers of nodes in the network. In this scenario, V_s is set to 4 minutes. EROA/R produces practically no duplicates, independent of the node density. This is caused by the reactive mechanism, which prevents nodes from sensing as long as they receive a notification. It also shows the effectiveness of the invalidation of notifications when an update is performed. EROA/P and AEROA/P behave similarly. However, effectiveness in suppressing duplicates depends on the node density. With lower node densities, the cooperative management of update times is less effective. Thus, nodes sometimes do not know about previous updates and have to perform redundant read operations. However, the percentage of duplicates does not exceed 7% for EROA/P and AEROA/P even with, on average, only one node within the transmission range r_{tx} of each virtual sensor (32 nodes: $32\pi(r_{\text{tx}})^2/A = 1$). The Isolated

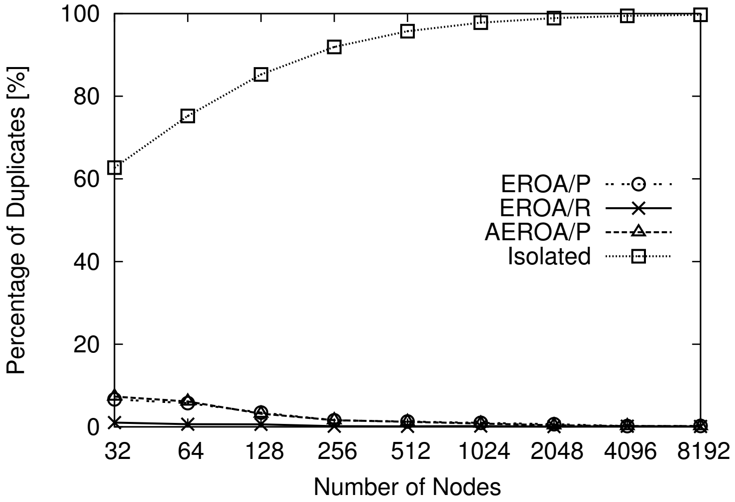

Figure 5.27: Percentage of duplicates for different numbers of nodes

approach fails to prevent duplicate updates, since nodes are unaware of previous updates. Even for a relatively small number of 32 nodes the number of duplicates reaches 60% since multiple nodes pass by the same virtual sensors shortly after each other.

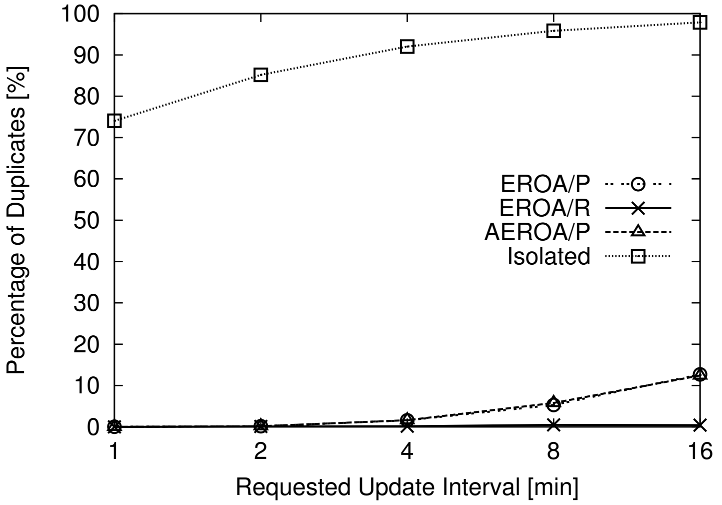

Figure 5.28: Percentage of duplicates for different update intervals V_s

Figure 5.28 depicts the percentage of duplicates for varying values of the requested update interval V_s. The number of nodes in the network is 256. EROA/R produces, independent of the requested update interval V_s, practically no duplicates. The percentage of duplicates for EROA/R and AEROA/P increases slowly with growing values of V_s, since the cooperative management of update times is less reliable for longer update intervals. Still, Figure 5.28 indicates that the effectiveness of the lightweight proactive approach degrades gracefully. The absolute number of updates for the Isolated approach is independent of V_s at a high level and, thus, the percentage of duplicates grows.

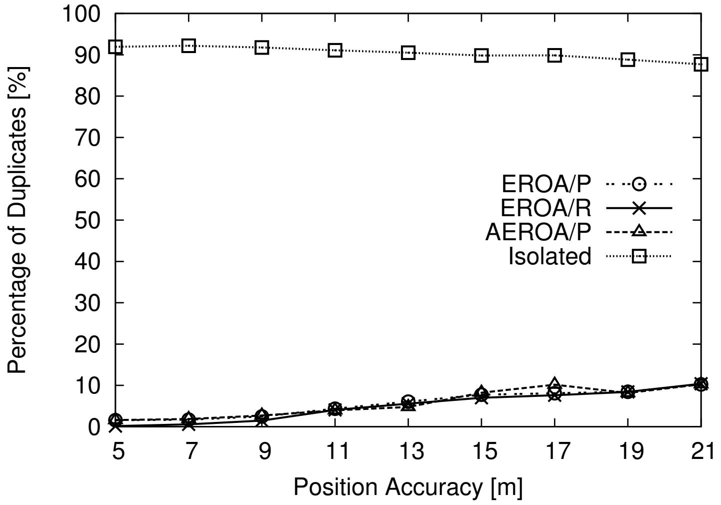

Figure 5.29: Percentage of duplicates for different values of position accuracy r_{accuracy}

In Figure 5.29, the percentage of duplicates for different position accuracies is plotted. In this case, EROA/R, EROA/P, and AEROA/P behave similarly. The increasing percentage of duplicates with high inaccuracy is caused by the fact that the size of target regions grows and, therefore, these regions can overlap. While trying to read a certain virtual sensor, a node may unintentionally read another sensor, possibly resulting in a duplicate update. The drop of the percentage of duplicates for the Isolated approach is due to the increased number of collisions, which prevents nodes from successfully sensing and updating a virtual sensor. When evaluating the percentage of duplicates for different sensor densities, the same effect can be observed. However, in this case, the overlapping of target regions is not due to growth in size but simply because there are more virtual sensors. In both cases, these duplicates do not put additional load on the nodes, because they are a byproduct of the requested sensing.

Figure 5.30 plots the percentage of duplicates for different values of the maximum node speed v_{maxn}. The percentage of duplicates for the EROA/P and AEROA/P approach increases with growing speed of the nodes. This is caused by the shorter average time that nodes stay in transmission range of a virtual sensor and, therefore, the probability of losing the most current update time.

The EROA/R algorithm causes almost no duplicates at all. The only duplicate updates are performed when a node tries to read a specific virtual sensor and unintentionally succeeds in sensing a different sensor. In addition, EROA/P produces duplicates, as the eval-

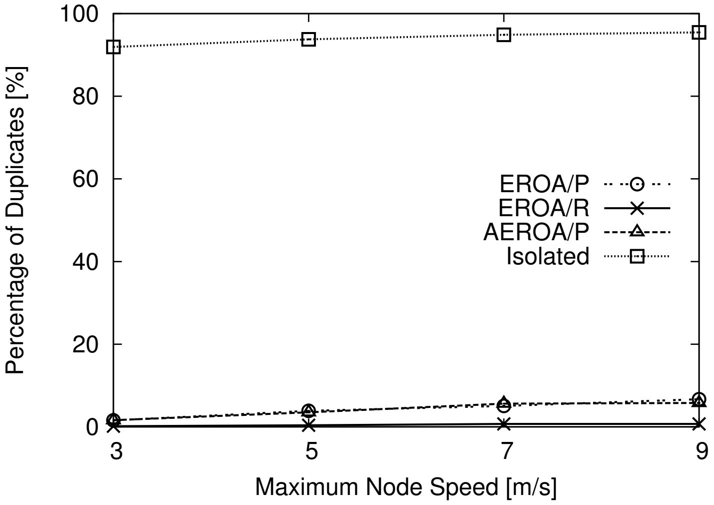

Figure 5.30: Percentage of duplicates for different values of maximum speed v_{maxn}

uation shows, due to the loss of the most recent update time. This effect was expected for large update intervals V_s, as well as for increased node speed. However, only a small number of duplicates are performed for a wide range of node densities. This is due to the self-tuning of EROA/P. In the case of low node density, the percentage of duplicates is small because nodes only seldom pass by a virtual sensor. With high node density, it is even lower, because EROA/P profits from an increased redundancy of nodes that manage t_{update}.

5.5.2 Results for Segment Sensors Supporting Environmental Monitoring

The following implementations are compared in this section:

CSS: This implementation is based on the CSS algorithm (cf. Section 5.3.2). A central instance coordinates sensing of mobile nodes.

MCSS: In essence, MCSS implements the CSS algorithms. In contrast to CSS, it relies on a WMN for the communication between mobile nodes and the PSS.

DSS: In contrast to the CSS implementation, DSS is based on the distributed stream shaping algorithm (cf. Section 5.3.3), where mobile nodes coordinate sensing in an ad-hoc network. The central instance is only responsible for managing the data read by mobile nodes.

Isolated: A simple isolated approach where all nodes sense independently. A node starts sensing when it enters a segment for which a virtual reading needs to be acquired;

it stops sensing as soon as it leaves the segment or as soon as the period of the virtual reading is finished. This implementation presents the worst case for redundant sensing, but also the best case for coverage.

Each edge of the street graph is assigned a virtual sensor. During the experiments, a virtual reading is acquired every 100 seconds. By default, V_δ is 60 seconds, and V_k is 2.

Effectiveness

This section presents the results regarding the effectiveness of the algorithms CSS, DSS and Isolated in terms of achieved spatial coverage as defined in Section 5.3.1. The results of MCSS are skipped, since they are the same as those of CSS. The average spatial coverage $c_s(V, R)$ of the virtual readings acquired during a simulation run is used as performance metric.

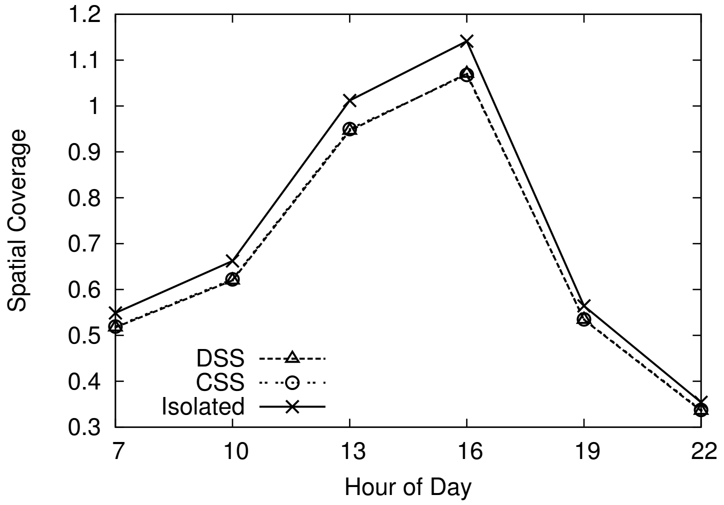

Figure 5.31: c_s in the course of day

At first, the performance of the algorithms is evaluated at different hours of the day, and the respective node mobility. Figure 5.31 plots the spatial coverage during the course of a day. In this scenario, 10000 mobile nodes are in the service area. Note that only a small fraction of these nodes moves on roads during simulation time. And this fraction depends on the hour of the day. At rush hour, e.g., at 16:00 o'clock, Isolated achieves its peak coverage value. However, as the coverage value of about 1.1 indicates, the node density is still too low to achieve the requested coverage of 2. As expected, the coverage highly varies during the course of the day. However, CSS and DSS show a similar behavior as Isolated resulting in a slightly lower coverage value. This gap of about 6% is due to the fact that Isolated starts sensing as soon as a node enters a segment. In contrast, CSS and DSS start sensing based on movement predictions. However, e.g., at a crossing, such predictions can only be determined after the node follows a road segment for some time. Therefore, during that time, sensing is deactivated, and the coverage is reduced. The coverage-loss is independent from the hour of the day. It depends on the position uncertainty, which is 5 m in the simulations. A node misses some measurements when the uncertainty is high compared to r_{range}. If the uncertainty is low, or if r_{range} is high, a node does not move far, compared to r_{range}, until it can start sensing. Therefore, it does not miss measurements.

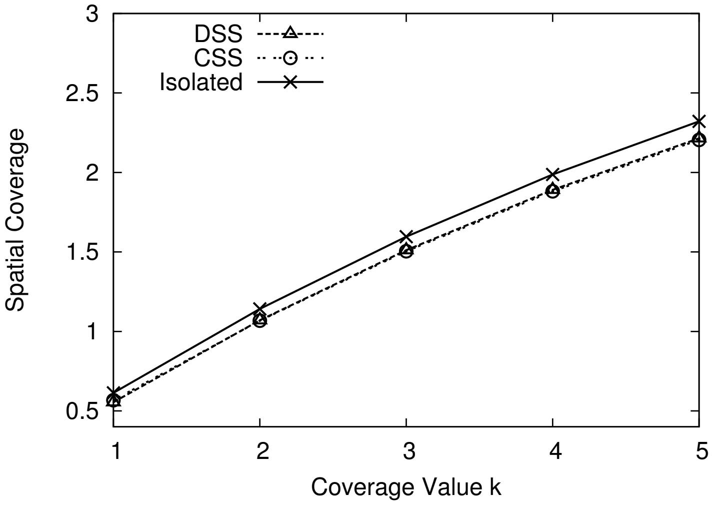

Figure 5.32: c_s depending on requested coverage

Figure 5.32 plots the achieved spatial coverage over V_k at 16:00 o'clock with a node density of 120 nodes per km. The Isolated approach achieves the maximum possible spatial coverage. CSS and DSS achieve a coverage value that is about 5% lower. This gap is, as in the previous paragraph, due to the delay introduced by the movement prediction. However, as the figure shows, this gap is independent from V_k, i.e., CSS and DSS are effective for a wide range of coverage values. The achieved coverage value is always below the requested coverage V_k. This shows that some segments are not populated enough to be covered by nodes, while others are highly populated and a much higher coverage value as the requested could be achieved.

Only a small loss of coverage can be seen in Figure 5.33. Here, the spatial coverage is depicted for varying node densities. As the figure shows, this loss is independent of the node density. This shows that CSS and DSS are effective independent from the node density. Moreover, the effectiveness of the algorithms is studied for varying values of V_δ. The results confirm that DSS and CSS are effective independent of V_δ.

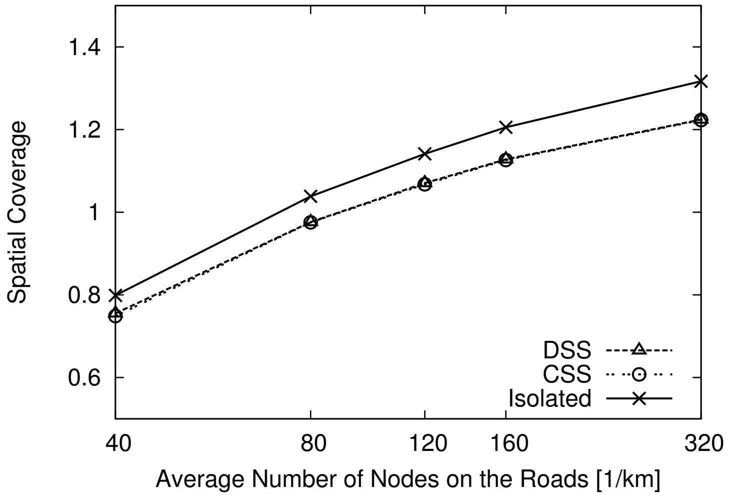

Figure 5.33: c_s depending on number of nodes

Efficiency

This section presents the results with respect to efficiency of the algorithms in terms of energy consumption (EC).

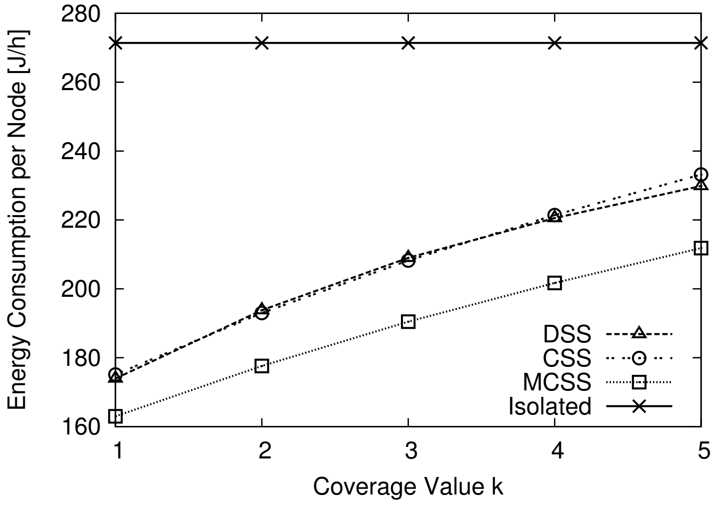

Figure 5.34: Energy consumption depending on V_k

Figure 5.34 plots the average energy consumption per node and hour depending on different values of V_k. In this scenario, the number of nodes per km is 120. Isolated, as expected, is independent from V_k resulting in a constant energy consumption. In contrast, CSS, MCSS and DSS show an increasing energy consumption for increasing values of V_k. This is due to the fact that less redundant sensing can be avoided through coordination.

Due to the lower communication cost, MCSS results in the lowest energy consumption. Interestingly, the energy consumption of DSS and CSS are nearly identical. While the energy consumption for communication are much higher in case of CSS due to GPRS, DSS suffers from redundant sensing. This redundancy is based on the fact that nodes can only coordinate using an ad-hoc network if they are in transmission range.

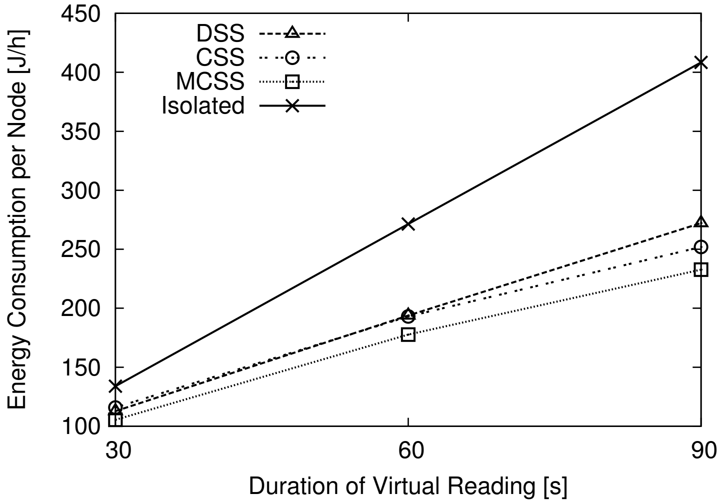

Figure 5.35: Energy consumption depending on V_δ

As Figure 5.35 shows, savings of CSS, MCSS and DSS increase compared to Isolated with larger V_δ. This is because the overhead of the coordination is almost independent from V_δ, while its benefit, i.e., the reduction of redundant sensing, grows with the increasing time-span for sensing.

Finally, Figure 5.36 plots the energy consumption for several numbers of nodes in the network. As expected, Isolated shows an energy consumption that is independent of the node density, since each node independently senses without considering other nodes. In contrast, DSS, CSS and MCSS benefit from increasing node density by preventing unnecessary sensing and, therefore, allow to reduce the energy consumption per node on average. Interestingly, the difference between DSS and CSS is decreasing with increasing node density. This is due to the improved ability of nodes for ad-hoc coordination. Moreover, with increasing node density, savings of DSS and CSS compared to Isolated increase. High node densities promise even higher energy savings for DSS and CSS compared to Isolated. Again, MCSS achieves the lowest energy consumption, since it has the lowest communication cost.

5.5.3 Results for Segment Sensors Supporting Object Detection

In this section, the following implementations are evaluated:

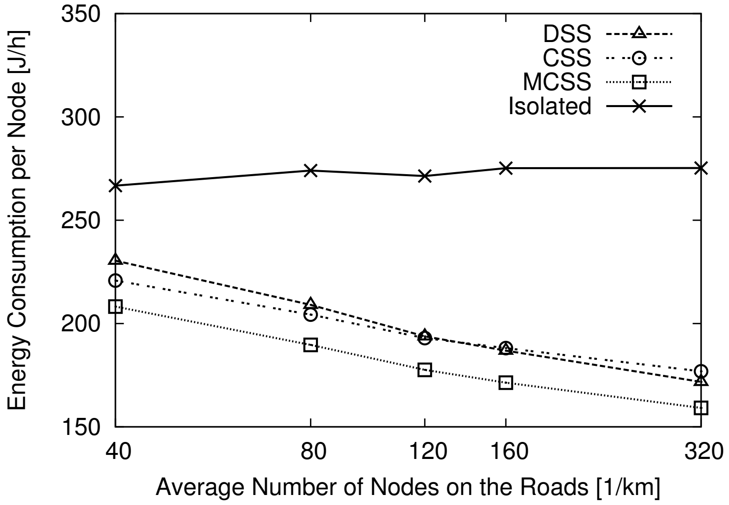

Figure 5.36: Energy consumption depending on number of nodes

Centralized: Pessimistic implementation based on the central coordination algorithm (cf. Section 5.4.3). A central instance coordinates sensing of mobile nodes.

MCentralized: Pessimistic implementation similar to Centralized. However, it relies on a WMN for the communication between mobile nodes and the PSS.

Distributed: Optimistic implementation based on the distributed coordination of nodes (cf. Section 5.4.4), where mobile nodes coordinate sensing in an ad-hoc network. The central instance is only responsible for managing the measurements.

Isolated: Reference implementation where nodes independently sense while on the road network. The Isolated approach presents the worst case for redundant sensing, but also the best case for coverage. This is basically the state of the art [LZG$^+$06].

A virtual sensor is established for each edge of the street graph. Nodes have an RFID reader as proximity sensor with $r_{\text{range}} = 5\,\text{m}$, and they use GPRS for communication with the server. The average object density is 33 objects per km of the road network.

Effectiveness

This section shows the effectiveness of the algorithms in terms of the achieved spatio-temporal coverage (cf. Section 5.4.1). Note that these results show the coverage derived from a single object speed range: $[-v_{\text{maxo}}, v_{\text{maxo}}]$. The average coverage C_{avg} achieved by the mobile nodes is used as performance metric. In essence, it is the average of the spatio-temporal coverage values c_{st} for every virtual reading acquired during a simulation run.

Since MCentralized achieves the same results as Centralized, it is omitted in this section.

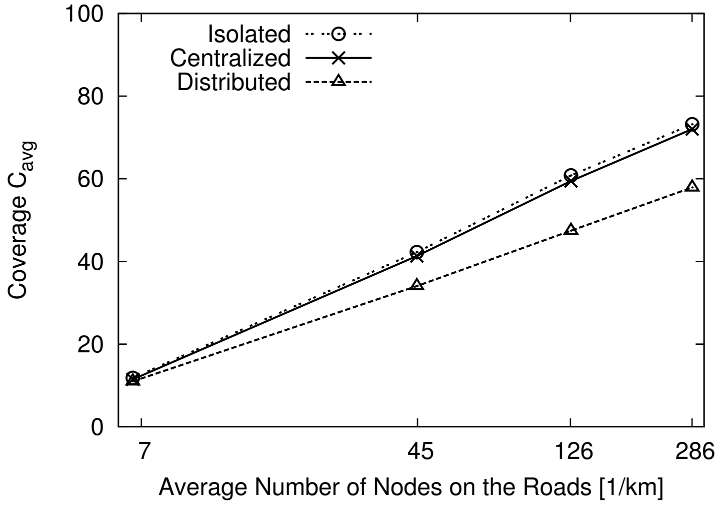

Figure 5.37: C_{avg} against number of nodes

Figure 5.37 shows C_{avg} depending on the average number of nodes on the roads. In essence, all three approaches show similarly increasing coverage. Due to redundancy in node movement, C_{avg} scales less than linear with the number of nodes. Isolated achieves the maximum coverage. While Centralized achieves a coverage only slightly lower (about 1%), Distributed further reduces the coverage (at most 13%). This is due to the optimistic design of Distributed that may loose coverage when nodes accelerate. However, when the density is low, nodes are more likely to read independently from other nodes and, therefore, the probability of optimistic sensing deactivation is reduced. On the other side, with high node density, redundancy leads to more optimistic sensing deactivations and, thus, to a higher loss.

Figure 5.38 shows C_{avg} depending on v_{maxo} with a node density of 45 nodes per km. As expected, all three approaches behave similarly. The maximum coverage is achieved for stationary objects. With increasing v_{maxo}, C_{avg} decreases rapidly. However, the decrease is limited since covered areas spanned by multiple nodes are independent from v_{maxo}. The explanation for the gap between the three approaches is analogous to Figure 5.37.

Efficiency

To compare the efficiency of the different algorithms in terms of energy consumption (EC), Figure 5.39 plots the energy consumption depending on the average number of nodes. As expected, Isolated's energy consumption is independent from the number of nodes. In contrast, Centralized, MCentralized and Distributed benefit from increasing

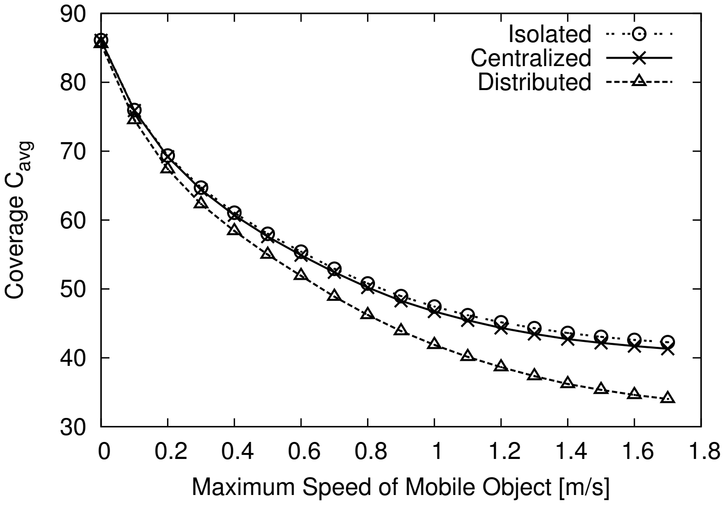

Figure 5.38: C_{avg} depending on v_{maxo}

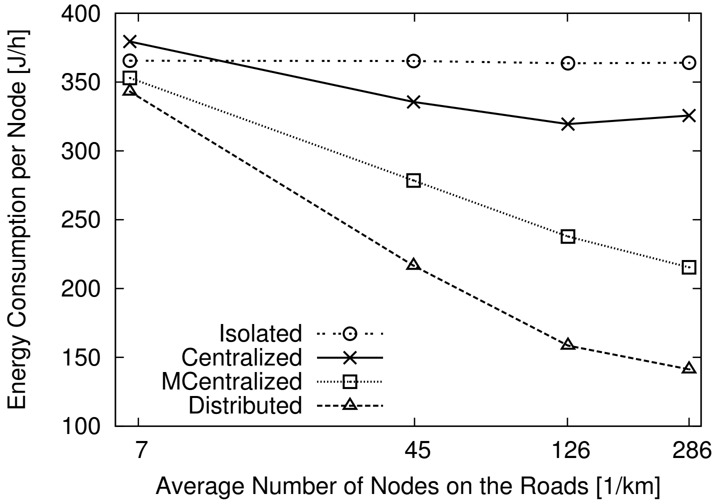

Figure 5.39: Energy consumption against number of nodes

node density by preventing unnecessary sensing. Savings are higher with increasing node density, when redundancy of movement increases. Moreover, a large gap between Centralized and Distributed can be observed. With 7 nodes per km of road, Centralized's energy consumption even exceeds that of Isolated. One reason is the high power consumption for communication with the server over GPRS. When the node density is low, few redundant measurements are prevented, and even outweighed by the additional communication cost. The second reason for the gap are the design principles of Centralized and Distributed. Centralized is pessimistic and developed to achieve the maximum possible coverage. Distributed is more optimistic, i.e., it may loose coverage. It achieves energy savings of up to 63% compared to Isolated at the cost of only 13% less coverage. Although MCentralized does not need expensive cellular communication, it is less efficient than Distributed. In essence, this is because the optimistic approach reduces sensing much more than the pessimistic approach.

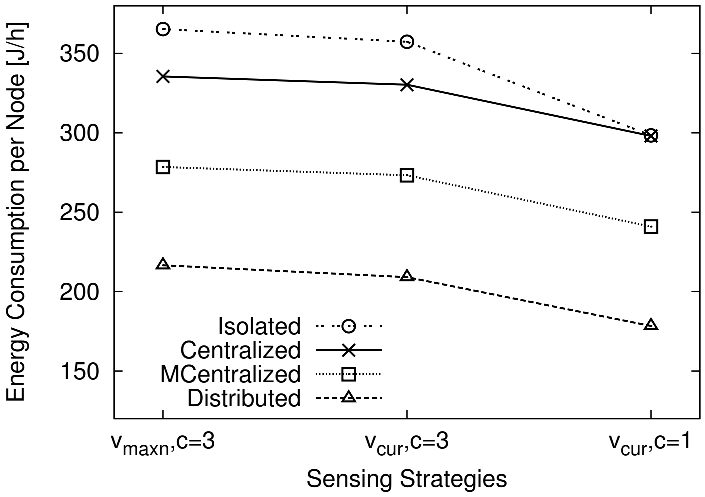

Figure 5.40: Effect of sensing strategies

In Figure 5.40, different sensing strategies (cf. Section 5.4.5) are compared. The average node density is 45 nodes per km, and $r_{\text{range}} = 15\,\text{m}$. On the x-axis, from left to right, the optimism of the strategies increases. Most pessimistic is $v_{\text{maxn}}, c = 3$. It assumes that the node speed is limited to v_{maxn}, and that the sensing redundancy needs to be set to 3 to achieve the required coverage along a trajectory. More optimistic is $v_{\text{cur}}, c = 3$ since it assumes that a node keeps its current speed v_{cur}. Most optimistic is $v_{\text{cur}}, c = 1$ without redundancy. As expected, all approaches reduce the energy consumption for higher degrees of optimism. However, the differences between the strategies w.r.t. absolute savings are small, due to the high energy consumption of positioning. This suggests that increasing the success probability only slightly increases the energy consumption.

5.5.4 Results for Object Number Estimation

This section shows how well the proposed coverage metric (cf. Section 5.4.1) suits for estimating the overall number of objects, i.e., how well the coverage models the detection ratio of sensing. For readability reasons, and because Distributed and Centralized show a similar behavior, only the results of the Isolated approach are plotted in this section. Unless stated differently, the coverage C_{avg} is computed according to the object speed distribution that has $[-v_{\text{maxo}}, v_{\text{maxo}}]$ equally divided into 18 object speed ranges. Furthermore, the sensing period V_δ during which objects are detected and coverage is achieved is 1200 seconds.

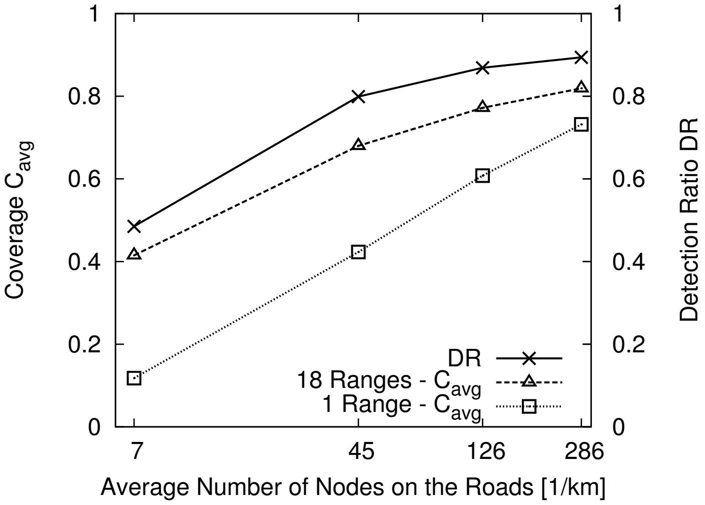

Figure 5.41: Comparison of C_{avg} and DR

In Figure 5.41, C_{avg} is compared with the detection ratio (DR), i.e., the number of detected objects divided by the overall number of detectable objects. Since C_{avg} depends on the number and size of the assumed object speed ranges, C_{avg} is plotted for a single range (pessimistic) and for 18 ranges. While the former is a lower bound for the DR, the latter is much more consistent with DR. Both, C_{avg} and DR similarly depend on the average number of nodes. The higher C_{avg}, the higher is DR. In the following, it is first shown how well the estimation captures the real number of objects, before the reasons for deviations between estimated and real number of objects are studied.

Figure 5.42 depicts the relative estimation, i.e., the estimated number of objects divided by the real number of objects in the simulation, as a metric for the quality of the estimation. The x-axis shows the size of the sensing window. As the figure shows, the estimation is closer to one (the optimum) for increasing sensing window and for larger sensor densities. The estimation suffers from small window sizes, due to the random variations of node movement, which are more likely to cancel out each other for larger window sizes.

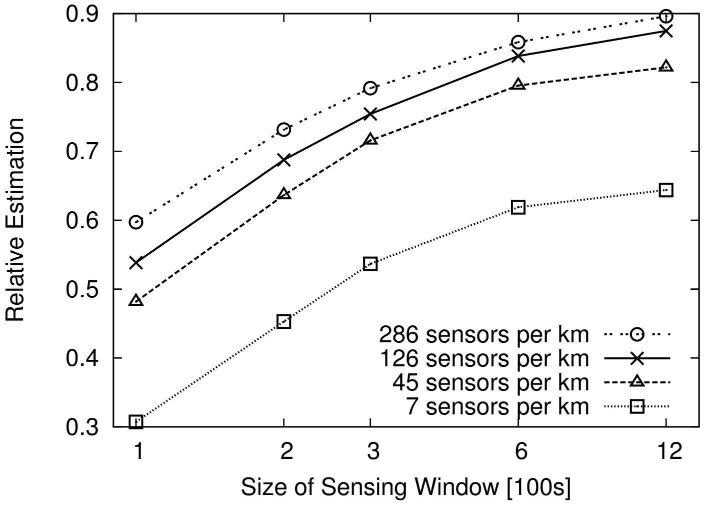

Figure 5.42: Relative estimation for different sensing window sizes

Moreover, the figure suggests that the quality of the estimation is limited by the sensor density, i.e., further increasing of the sensing window does not result in a better estimation.

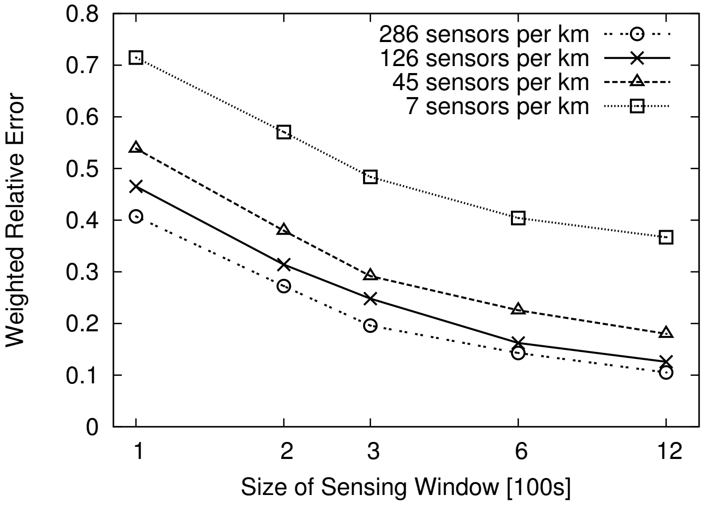

Figure 5.43: Relative weighted error for different sensing window sizes

Figure 5.43 shows the weighted relative error of the estimation. It is computed as the sum of the absolute values of the differences of real number and estimated number of objects for all segments divided by the overall number of objects. In contrast to the relative estimation, it assures that positive and negative errors of the estimation do not cancel out each other. In essence, Figure 5.43 shows the same results as Figure 5.42. The fact that the sum of the weighted error and the relative estimation is roughly one, indicates that there are almost no negative errors, i.e., estimations are too low rather than too high.

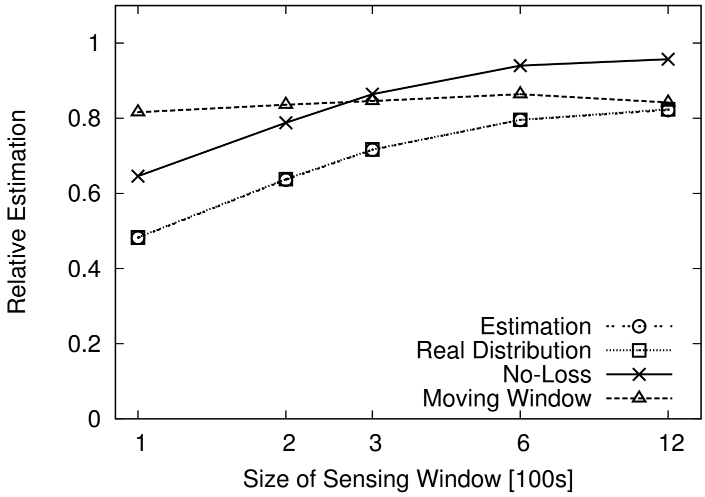

Figure 5.44: Effect of error causes on the relative estimation

To study this systematic error of the estimation, the following causes of error are evaluated in a scenario with a node density of 45 nodes per km.

Real Distribution: The estimation is based on the knowledge of the real object speed distribution, rather than the observed object speed distribution.

No-loss: No-Loss shows the estimation that would be achieved if all objects would be detected that are within a covered area. Note that the coverage is computed based on the assumption of continuous object movement. However, this assumption might be violated by objects that enter or leave a segment not at a crossing, but for instance to a building. In such a case, an object might be in a covered area, however disappearing before actually being detected by a mobile node.

Moving Window: Moving Window shows the estimation based on a moving sensing window. In this case, sensing of nodes is done continuously and the estimation is based on the measurements and the coverage during the past sensing time frame. In such a scenario the computed coverage is not biased by start and end period of sensing that lead to wrong coverage values.

Figure 5.44 shows the results of the estimation if the causes of error were compensated. For comparison, Estimation shows the default estimation without compensation of any error. Real Distribution achieves almost the same values as Estimation, which is based on the monitored speed distribution. This shows that the mechanism for monitoring the object speed distribution achieves to approximate the real speed distribution with an accuracy that is good enough for the estimation. No-loss shows a constant offset to Estimation. Note, that this error is only relevant if every object that enters a segment needs to be detected. If only the object density is of interest, then the number of appearing and disappearing objects would cancel out each other, and no error would be caused. As depicted in Figure 5.44, the estimation of Moving Window is independent from the size of the sensing window. Moreover, the figure shows that errors caused by the start and end of sensing are negligible with a fixed sensing window of 1200 seconds.

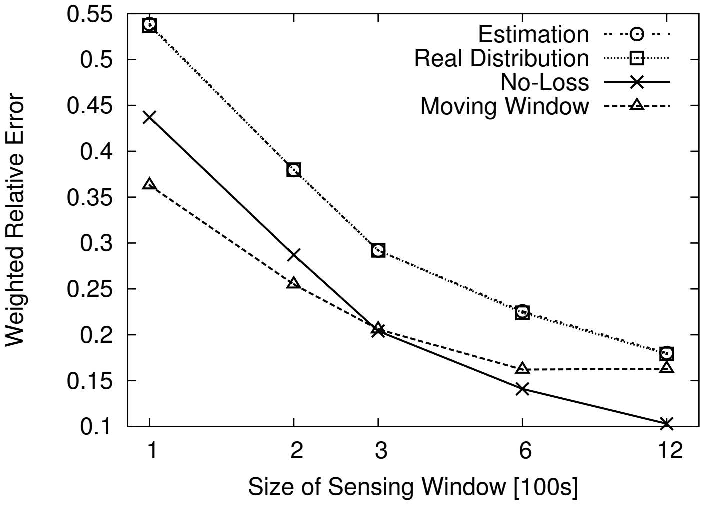

Figure 5.45: Effect of error causes on the relative weighted error

Interesting to mention is that the weighted relative error of the Moving Window still depends on the size of the sensing window (cf. Figure 5.45). This is due to the fact that the coverage and the actual number of detected objects during the sensing window is subject to coincidence, and with small sensing windows the sampling size is too small to compute accurate estimations. The other estimations show no unexpected behavior with respect to the weighted relative error.

To further show the significance of the actual number of detections that are the base for an estimation, Figure 5.46 plots the weighted relative error of estimations for four scenarios with different conditions. In each of these scenarios an estimation for a segment is only considered if some minimum number of objects where detected on this segment. Especially for low density networks when it is likely that only few objects are detected on a segment, a large gap between the different scenarios can be observed. In essence, this means that without a significant number of detections no accurate estimation is possible.

Finally, Figure 5.47 shows what effect the granularity of the speed distribution has. It plots the relative estimation and the average coverage C_{avg} for a different granularities of

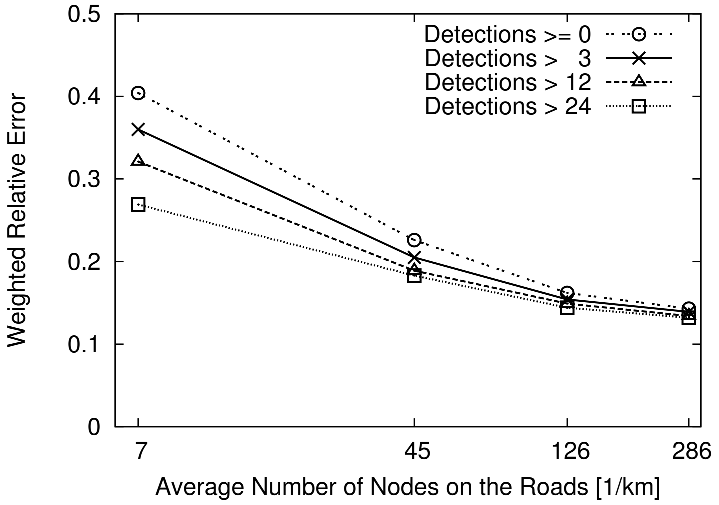

Figure 5.46: Significance of the number of detections on the relative weighted error

the speed distribution, where the range $[-v_{\text{maxo}}, v_{\text{maxo}}]$ is partitioned into a number of ranges of equal size.

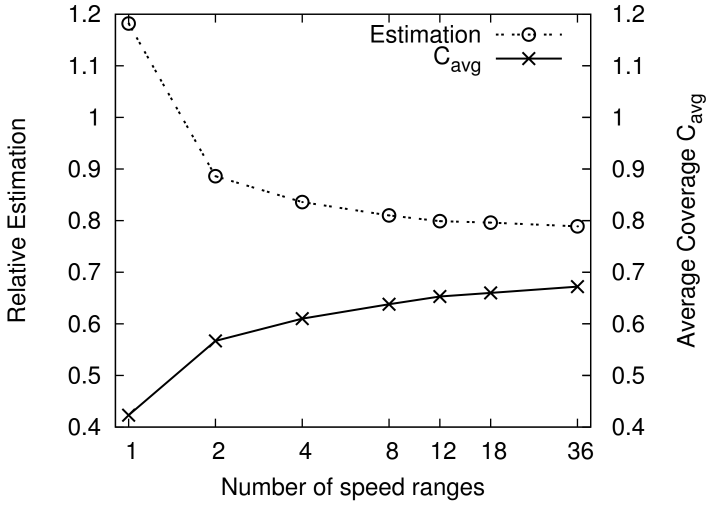

Figure 5.47: Effect of speed distribution granularity

With a low number of speed ranges, the coverage is too pessimistic and, therefore, the estimation is too high. With increasing number of ranges, coverage and the estimation flat out. This suggests, that a wrong number of ranges leads to only small errors of the estimation when the number of ranges is high. Although it is hard to model the movement characteristics of mobile objects accurately, this figure shows that the estimation is quite robust against deviations from the assumed object movement. Although the relative estimation is 0.9 with two speed ranges in this figure, it does not suggest to always select two speed ranges to achieve the best relative estimation. In contrast, it only shows that the error introduced by using two speed ranges almost completely compensates for the other, previously described, errors in this scenario.

5.5.5 Summary

The coordination algorithms presented in this section aim for the acquisition of sensor data according to given coverage requirements. Depending on the type of virtual sensor that is provided as an abstraction for applications from physical entities, different algorithms were proposed and evaluated.

The evaluation covered two main aspects of sensing. First, the effectiveness in terms of coverage compared to continuous sensing. Second, the efficiency of sensing in terms of energy consumption of the battery powered mobile nodes. The evaluation was performed using extensive simulations. To model the mobility of mobile nodes and mobile objects, realistic mobility models were incorporated.

The evaluation results concerning the algorithms to coordinate sensing for virtual point sensors suggest that hundreds of hours of public sensing consumes only as much energy as is the capacity of a typical mobile phone battery. Moreover, compared to continuous sensing, the results showed that hardly any updates are missed.

Regarding virtual segment sensors, mechanisms for the automated mapping of urban areas were introduced. The evaluation results showed that the proposed algorithms achieve about 95% of the maximum possible coverage. Moreover, the results showed that energy savings of up to 50% compared to uncoordinated sensing can be achieved.

In addition, mechanisms for the automated detection of mobile objects in urban areas that provide a virtual sensor abstraction were conceived. Using simulations, it was shown that these algorithms achieve energy savings of up to 63% while reducing coverage by only 13%. It could also be shown that the proposed coverage metric closely models the detection ratio of objects.

5.6 Related Work

Due to advances in miniaturization and computing, sensing is currently an active research field. In Section 2.2, the potential of public sensing, and its challenges were reviewed. Moreover, some systems that deal with public sensing were exemplarily presented. In contrast, this section presents and compares related work in detail. Deficiencies of related work that are overcome by this dissertation are pointed out. Moreover, it is shown how related work complements this dissertation.

In this dissertation, privacy in public sensing is not explicitly addressed. However, the concepts presented in this dissertation are easily extended by approaches that preserve privacy. For instance, AnonySense [CKK$^+$08] proposes a framework for preserving privacy of participating users in urban sensing scenarios. Similarly, [APG$^+$10] proposes a technique for privacy preservation. Pham et al. [PGU$^+$10] investigate mechanisms to preserve the privacy of sensors through perturbation, i.e., sensors are allowed to lie about their location.

The remainder of this section is structured as follows. First, mechanisms for coordinating sensing are discussed. Then, approaches that address efficient sensing are discussed. Afterwards, related work for the sensing of environmental phenomena are listed. This section is closed with a discussion of related work covering mechanisms for object detection.

5.6.1 Coordination of Sensing

The main contribution of this dissertation are the mechanisms for the coordination of the cooperative sensing to fulfill defined quality goals. The general need for coordination of sensing in large-scale sensor networks was already highlighted by [EGHK99].

The research field of wireless sensor networks mainly focuses on the autonomous monitoring of environmental parameters in inaccessible areas. For instance, [SRJ10] presents decentralized algorithms for selecting a subset of sensor nodes from a stationary sensor network in the plane such that the sensor coverage is maximized and frequency conflicts of nearby sensor nodes are prevented. The system model of [SRJ10] is completely different from this dissertation and, thus, its results are not applicable for public sensing scenarios. However, it shows the relevance of coordination algorithms for the efficiency of cooperative sensing.

Considering mobile sensors, [BCF$^+$10] presents a decentralized mechanism for the autonomous deployment of mobile sensors to achieve complete coverage of the area that is to be monitored. Nodes cooperatively select positions on a hexagonal grid, which is proven to be the optimal arrangement to achieve complete coverage of a planar area. This approach requires control over the sensor nodes. However, such a control is not available in opportunistic sensing systems.

Approaches that deal with sensor nodes especially tailored to urban environments are [CM06, BKM+04]. They aim for environmental sensing. However, due to the high price of sensor nodes and the high density requirements of sensing, these approaches suffer from the same issues as traditional wireless sensor networks. They imply high cost and maintenance effort for large-scale deployments. As discussed in Section 2.2, such approaches are complementary to public sensing with mobile phones, and especially suited to scenarios that require continuous monitoring of phenomena, which cannot be guaranteed in a completely opportunistic sensing system.

The coordination algorithms rely on mechanisms to inform the nodes about their sensing tasks. The field of data-centric storage deals with similar goals. Ratnasamy et al. [RKS+03] propose a geographic hash table (GHT) for dissemination of data to specific locations. However, GHT is proposed for static environments and, therefore, not applicable to the dynamic environment of opportunistic sensing. Zahn et al. [ZS05] propose a distributed hash table (DHT) for MANETs. However, the overhead for maintaining a DHT in a dynamic environment is too high for the infrequent data access rates considered in this dissertation. In general, DHTs do not consider geographic proximity. In contrast, [DMR07] presents location based storage and migration mechanisms for maintaining data at specific locations in MANETs. Although [DMR07] allows for storing data close to specific locations, it requires frequent position fixes for geographic routing.

Bubble-sensing [LLEC10] binds sensing tasks to physical areas of interest. Data acquired at these locations is then sent to the node that initiated the corresponding bubble. This is similar to the mechanism proposed in this dissertation for periodically updating sensor data at specific points. However, the approaches differ in two basic characteristics. First, bubble-sensing is a communication-heavy mechanism that puts a lot of effort into anchoring a sensing task at a specific location. In contrast, this dissertation proposed a lightweight mechanism that limits this effort if it would exceed the cost induced by a loss of the anchor. Second, this dissertation optimizes the maintenance of the state of the sensing task through quality-awareness.

Similar to this dissertation, [KZ07] proposes virtual sensors as a data-centric abstraction for applications to decouple applications from the mobile sensors. When querying measurements, an application addresses virtual sensors rather than the physical mobile sensors. However, [KZ07] lacks suitable metrics for specifying the quality of the data to be acquired and for measuring the resulting quality. A more specific sensor network abstraction for a public sensing system is presented by [PDR11]. It allows to specify so called points of interest for sensing. Several algorithms are proposed to select those nodes for sensing that allow for efficient acquisition of sensor data at those points. Although we agree on the idea for a virtual sensor layer to decouple applications from mobile sensors, this approach has some limitations. First, creating detailed environmental maps is inefficient since it handles every point of interest independently. Second, the system only supports one-shot queries. Periodic queries need to be transformed to multiple one-shot queries.

Besides cooperative sensing, [FBO09] shows that cooperation of nodes is also beneficial for achieving other goals such as downloading data in vehicular ad-hoc networks (VANETs). Based on traffic flow modeling, [FBO09] selects carriers for data chunks that are likely to meet the target and deliver the data. Similar to the coordination problem in this dissertation, the coordination of downloading needs to deal with the dynamics of the system.

5.6.2 Efficiency of Sensing

The general problem of sensing efficiency is considered relevant by the literature. First, CenceMe [MLF+08] shows that continuous sensing reduces standby time of mobile phones drastically. Second, Mobiscopes [AAB+07] argues that continuous sampling needs to be prevented even in case of sensors integrated in cars. Otherwise, the large amount of measurements might result in network congestions and high storage and processing load. Finally, for reagent-based sensors that rely on chemical reactions to detect specific compounds such as carbon monoxide or nitrogen oxide, the lifetime of a sensor depends on the consumption of its reagent [OD08, Dia04]. The more a sensor is used to detect a compound the faster its remaining lifetime shrinks, which suggest that unnecessary sensing should be avoided.

Although commonly believed that efficiency is crucial, only a few projects propose mechanisms that increase efficiency of sensing. Most projects focus on system aspects. Gellersen et al. [GSB02] propose to integrate sensors into mobile devices to achieve direct context awareness of mobile devices. In essence, locally collected sensor data is not shared between devices. In contrast, [RNC05, KBP+08, KBRL09] use mobile devices for context data collection. Rudman et al. [RNC05] attach sensors for monitoring air pollution to a tablet PC. MobGeoSen [KBP+08] is based on the integration of sensors to mobile phones, which are carried by a large number of people. All of these approaches simply rely on continuous sensing at a fixed sampling rate and do not consider quality aspects of sensing.

LittleRock [PLL11] argues that continuous sensing requires a mobile phone's main processor and other components with high power consumption, to operate at a high power level, although the power consumption of the sensor might be very low. They show that even low sampling rates prevent a mobile phone from entering the sleep state and, thus, from saving energy. LittleRock proposes a special system architecture based on a dedicated low-energy microcontroller that controls sensing. Although this solution cannot exploit the potential for savings through reduced redundancy, it complements this dissertation's mechanisms that achieve savings through coordination of sensing.

Several other projects propose strategies to increase efficiency of sensing. A simple mechanism to reduce cost for sensing is to reduce the sampling frequency. However, this approach may result in loss of data quality. CarTel [HBZ$^+$06] employs a reactive mechanism to reduce load of cars that collect measurements. Before they transmit the collected data to a server for further processing and management, they transmit a short summary of their data. Based on this summary, the server then requests the required data. With this mechanism, data that was already received from another car, does not need to be transmitted redundantly. CarTel relies on the assumption that cost for data acquisition are irrelevant for car-based sensor. However this might not be true all kinds of sensors integrated in cars. Moreover, this assumption does not hold for mobile phone sensing.

Sensor substitution and sensor sharing are proposed by [ELC08] to fulfill sensing requirements in an opportunistic sensing scenario. The idea of sensor substitution is that applications require data of a specific type rather than data from a certain sensor. For instance, an application that needs position information may easily tolerate one of the alternatives GPS or WiFi-based positioning systems. The proposed sensor sharing allows nodes to access the sensors of other nodes. This basic idea is also the main motivation for the public sensing system in this dissertation. However, in this dissertation this principle is extended by selecting nodes that can contribute to a global sensing goal rather than having multiple, possibly overlapping, sensing goals.

Jigsaw [LYL$^+$10] proposes two mechanisms for efficient sensing of accelerometer data and sound data. First, adaptive sensing rate adjustment based on the quality of the data. For instance, if the quality of data is bad, further processing is skipped. Second, which is also employed by [WLA$^+$09], user state dependent sensor selection and sampling rate specification. Both strategies to save energy work only locally. The data quality or the user state is locally evaluated to determine the set of sensors that need to be sampled. However, in an opportunistic sensing scenario, the type of sensors that need to be sampled is determined globally and not subject to local optimizations of sensing.

Wang [Wan11] presents a sensor management scheme for efficient and automatic user state recognition. Based on a model of state transitions, the system activates only those sensors that are needed to detect state transitions. Power hungry sensors are activated on demand to verify user states. The proposed management scheme suits well personal sensing activities such as user state recognition. However, it does not consider the problem of sensor node selection.

A similar idea is investigated in the scenario of traffic monitoring with mobile phones that are carried by vehicles [MPR08]. Mohan et al. propose the concept of triggered sensing to activate expensive sensors, such as audio, based on the measurements of low-energy sensors such as the accelerometer. Moreover, the authors propose to process the measurements locally to save energy otherwise needed to send the measurements to the infrastructure.

In contrast, [CBkC$^+$10] investigates scenarios that benefit from offloading data processing from the mobile phone to a processing infrastructure. This enables complex processing operations that, otherwise, would not be feasible on the mobile device itself. Moreover, this mechanism may also reduce the energy consumption of mobile devices. However, savings from reduced processing load are easily outweighed by the energy cost for transmitting the data that needs to be processed. In essence, this strategy is only a reasonable means for energy efficient sensing in few specialized scenarios. But even in these cases, it has several drawbacks. First, it requires powerful infrastructure support and does not exploit powerful local resources. Second, it implies large communication costs. This may congest the wireless communication infrastructure, and it raises privacy concerns.

The problem of efficiently uploading measurements in case of continuous sensing is addressed by [MPF$^+$10]. The authors argue that communication cost directly relates to energy costs and, thus, needs to be limited to prevent rapid depletion of batteries. In essence, the authors propose server-side prediction of measurements. As long as the actual measurements only deviate from this prediction by a defined amount, the mobile sensors do not transmit their measurements. This mechanism is orthogonal to the coordination mechanisms proposed in this dissertation, and it complements them well.

Reddy et al. [RES10] focused on the development of a reputation system to support manual selection of nodes for long-running sensing tasks. This system allows to limit the number of involved sensors and, thus, increase efficiency. However, this system does not allow for autonomous node selection and adaptation of the selection. It only proposes candidate nodes based on the past characteristics such as availability or movement of the nodes. In contrast, to achieve efficiency of sensing, PRISM [DMP$^+$10] relies on a registration mechanism. Mobile phones inform a central instance about their availability for sensing. The central instance then distributes sensing tasks to the set of nodes that match the requested characteristics. However, central coordination does not scale for fine-grained task assignments. Nevertheless, such a mechanism can complement the coordination mechanisms developed in this dissertation. It can be used to deploy sensing tasks to nodes that are available for sensing.

5.6.3 Sensing of Environmental Phenomena

Especially related to the issue of coordinating the access to external sensors is work from the field of RFID-based multi-reader coordination [VD08, HES06]. [VD08] adaptively adjusts the power level of a RFID reader to increase the overall read throughput. In contrast, [HES06] coordinates frequency and time slot assignments of readers. Both approaches aim for increasing read throughput instead of optimizing energy efficiency of read operations. Moreover, these approaches assume static readers instead of mobile sensors.

Model-driven sensing approaches deal with quality-aware sensing and energy considerations. In contrast to this dissertation, model-driven sensing approaches select those nodes for sensing that can provide the highest information value. They are also referred to as utility-driven approaches. For instance, [DK08] assumes spatial correlation of measurements, i.e., sensor values differ at most by a value that is limited by the distance between the locations where the values are acquired. The authors propose algorithms for selecting a subset of nodes that allow to predict aggregate functions such as maximum or mean with a minimum error. Although the selection of sensors based on the information value they can provide is an interesting approach, the proposed algorithms are not applicable to a highly dynamic scenario of opportunistic sensing.

More tailored towards community sensing is the model-driven sensing approach by Krause et al. [KHKZ08]. The authors propose sensing policies based on models of sensor availability, models of demand and models of the phenomena. In essence, this is a centralized approach that deals with mobility of sensors by modeling the inaccuracy of their position. However, without detailed position information, fine-grained coordination and assignment of sensing tasks is not feasible. Moreover, this approach lacks distributed coordination algorithms that are essential for scalability and efficiency, and it does not optimize continuous sensing.

5.6.4 Detection of Mobile Objects

Beyond the monitoring of stationary phenomena, the detection and tracking of mobile objects or mobile events imposes additional challenges. MobEyes [LZG+06] uses instrumented vehicles to detect mobile objects, for instance, using license plate scanners. MobEyes puts focus on the efficient dissemination of summaries describing the detected objects rather than on the efficient detection of the objects. Similarly, [FBRK07, FBMK08] use instrumented mobile devices to find objects. The authors propose scoped query mechanisms to efficiently distribute queries to the sensors that are likely to find the queried objects. These mechanisms complement this dissertation in that they allow to specify interesting regions for sensing. However, these approaches perform simple constant rate sensing, and do not consider efficiency of sensing.

In the area of robotic sensors, i.e., sensors whose mobility can be controlled, [KSJ+04] introduces mechanisms to build a so called fluid infrastructure for sensing that adapts to the detection of objects and moves the sensors accordingly to achieve a good coverage for the tracking. However, such control of sensor mobility is not available in an opportunistic sensing system.

MetroTrack [AML+10] extends beyond the scope of object detection. It aims for tracking mobile events. Nodes locally evaluate sensor measurements and, if a measurement meets the criteria of the event that is to be tracked, they inform their neighbors. To recover lost targets, MetroTrack proposes a geocast-based mechanism that forwards the query to the predicted area of the target. However, it does not monitor the achieved coverage of sensing to reduce redundant sensing in order to increase efficiency.

In the field of sensor networks, coverage metrics [Bra07] and mechanisms [MLG+10] for tracking or detecting objects have been actively researched. Shrivastava et al. [SMUS06] present fundamental limits and algorithms for target tracking in wireless sensor networks. If sensor locations and the sensing range are known, a deterministic coverage analysis based on computational geometry can be performed. Otherwise, if only statistical properties of locations are known, the coverage can be determined by stochastical geometry. The statistical coverage of a moving target is analyzed by [MM09]. Cao et al. [CJDC10] consider the case of collaboratively tracking a group of mobile targets. They achieve efficiency of sensing by monitoring the perimeter of the moving group instead of monitoring each individual target. Since these wireless sensor network scenarios fundamentally differ from the opportunistic sensing in urban areas, available coverage metrics that assume stationary or controlled sensors moving in the plane cannot be applied.

In addition to detecting or tracking targets, mechanisms for traffic delay estimation have been actively researched. Early work [MSHA03] determines traffic density based on loop detector data. Real world tests [MiKAM10, HWH+10] have shown that vehicles can serve as probes to analyze traffic flow and derive traffic delay estimations. VTrack [TRL+09] adjusts the sampling rate of different position sensors to estimate traffic delay efficiently. Work et al. [WBT+10] developed a traffic flow model that allows to incorporate velocity information from mobile devices. Although these mechanisms support analyzing traffic in urban areas, they either require a dedicated sensor network infrastructure or active probes that are part of the traffic that is observed. In contrast, we propose quality-aware and, thus, efficient mechanisms to estimate the density of passive objects without any sensing or communication capabilities. Optionally, we incorporate known speed distributions [AAR07, ZWHS09] of the objects that are monitored.

6 Conclusion

This chapter summarizes this dissertation and concludes it with a brief discussion of future research directions.

6.1 Summary and Conclusions

The evolution and proliferation of mobile sensing platforms such as mobile phones, enables services that not only adjust to the whereabouts and the profile of a user, but also to the state of the real world. With sensors seamlessly integrated into our everyday life through billions of mobile phones all around the globe, the vision of public sensing, i.e., monitoring and detecting a variety of physical phenomena by continuously collecting an abundance of sensor data, is feasible.

The challenges public sensing faces are manifold. First, in order to benefit from the large opportunistic coverage, sensing may not be obtrusive, i.e., sensing must not require user interaction and it may not require control of user mobility. In this thesis, this challenge is tackled by performing opportunistic sensing. Furthermore, since mobile phones are battery powered, the algorithms are designed for energy efficient sensing. Data acquisition exceeding the application requirements is avoided. To specify and assess the quality of the data, spatial and temporal coverage metrics were devised. Virtual sensors are used as a data-centric abstraction to decouple applications from physical entities. Thus, allowing to cope with the dynamics of sensor availability.

More detailed, this dissertation presented a layered architecture and a data-centric interface to build and interact with such an opportunistic public sensing system. Based on the characteristics of the phenomena that are to be monitored, virtual sensors were introduced that allow applications to request sensor data intuitively based on spatial, temporal, and quality requirements. In essence, virtual point and virtual segment sensors were devised that enable the monitoring of environmental phenomena as well as the monitoring of mobile objects.

In order to cope with the varying quality of data in an opportunistic sensing system, spatial, temporal and spatio-temporal coverage metrics were presented. These metrics allow specifying the quality requirements of applications, and they allow assessing the quality of the sensed data. These metrics differ depending on the class of virtual sensor that is considered.

Depending on the type of virtual sensor, centralized and distributed algorithms were developed for the selection and coordination of nodes according to the sensing requirements, while minimizing the energy consumption. The algorithms to coordinate sensing for virtual point sensors consist of two concurrent operations. The first operation provides the update time of virtual sensors. The second operation efficiently detects whether a mobile device is close to a virtual sensor. If this is the case it efficiently coordinates sensing. Besides a decentralized algorithm for coordinating read operations based on an ad-hoc network formed by the mobile nodes, this dissertation showed a complementary algorithm that exploits infrastructure-based coordination. By extensive simulations, the effectiveness as well as the efficiency of these algorithms was shown. The evaluation results suggest that hundreds of hours of public sensing consumes only as much energy as is the capacity of a typical mobile phone battery. Moreover, compared to a simple continuous sensing approach, hardly any updates are missed.

Regarding virtual segment sensors, mechanisms for the automated mapping of urban areas were introduced. More precisely, two algorithms for coordinated sensing in order to achieve the quality requirements, while minimizing the average energy consumption of nodes were proposed. By extensive simulations, it could be shown that the algorithms achieve about 95% of the maximum possible coverage. Moreover, the algorithms achieve a very high energy efficiency allowing for drastic savings up to 50% compared to uncoordinated sensing.

In addition, mechanisms for the automated detection of mobile objects in urban areas that provide a virtual sensor abstraction were conceived. The dissertation proposed a metric that captures the coverage. A centralized and a distributed algorithm were proposed to coordinate nodes such that redundant sensing is reduced, i.e., energy consumption is reduced. Using simulations, it was shown that these algorithms achieve energy savings of up to 63% while reducing coverage by only 13%.

In order to cope with the dynamics of sensor availability, the dissertation showed how to monitor the progress of sensing and how to adapt sensing to changes in sensor density and mobility of nodes. Moreover, this dissertation showed how to adapt the coordination mechanisms to the density of participating mobile nodes.

Basic group communication mechanism were conceived as a basis for the coordination mechanisms. These mechanisms allow for addressing specific nodes based on their symbolic location. With a symbolic location model that matches the structure of the virtual sensors, this communication abstraction allows to easily identify and address nodes relevant for the coordinated data acquisition of the virtual sensors. The proposed routing algorithms are tailored for WMNs. In essence, a hierarchic routing structure and a connectivity graph between dedicated nodes, structured according to a simple symbolic location model, is created and proactively maintained. Message forwarding is done greedily along paths of the connectivity graph and, if this fails, through the hierarchic routing structure. It was shown that routing achieves high message delivery rates at low routing overhead in terms of routing messages and path length stretch.

6.2 Future Research Directions

Beyond the potential for optimizations of the proposed concepts and algorithms of this dissertation, several promising research directions to extend this work can be named.

The assumption that nodes are homogeneous, i.e., their capabilities are comparable, is reasonable from a technological point of view. However, it might not hold for real world scenarios. For instance, only clients of a certain operator might connect to access points or mesh networks, whereas other clients, in the same area, would require to connect to cellular networks to access the Internet. Which access technology is used might even be subject to human control and, thus, resulting in scenarios where nodes at one location use different access technologies. However, in such scenarios, coordination algorithms need to take into account these differences of capabilities.

Mobile phones might also differ in their remaining battery capacity. Moreover, if mobile phones run out of energy, they opt-out of the public sensing. In this dissertation this effect was neglected, and infinite energy resources of the mobile phones were assumed. A worthwhile optimization would be to model battery capacity and adjust the coordination algorithms such that batteries are not depleted. In essence, this is related to the issue of fairness. A system is widely considered fair if all its components are equally loaded. However, this goal might be reconsidered in the field of public sensing. In community projects such as OpenStreetMap, contribution is often ranked and adds to the reputation of the contributors. Therefore, fair coordination algorithms must not prevent people from contributing more than others.

The question about the modalities of contribution is related to the general question of contribution. For instance, since sensing might affect privacy considerations of contributors, mechanisms need to be developed that preserve privacy of the contributors at the level specified by the contributor. Moreover, this means that users need mechanisms to specify the modalities for their contribution.

In order to further increase efficiency of sensing, self-tuning mechanisms need to be developed that automatically relax the coverage parameters of virtual sensors such that coordination overhead is reduced. Therefore, the trade-off between coordination overhead and benefit of coordination needs to be studied. Then, coordination algorithms can be adjusted to allow for deviations from the defined sensing goal, if this leads to an overall reduction of energy consumption. One approach to adjust the algorithms accordingly would be to incorporate randomized sensing decisions. In addition, considering multiple virtual sensors in the coordination process allows to exploit synergies and, thus, increase efficiency.

A different dimension in which this dissertation can be extended is the model-driven data acquisition. Based on a model of the phenomena that is to be observed, measurements at a specific location can be approximated using measurements acquired at other locations. Using such a model driven approach, the number of measurements that need to be acquired can be reduced. In essence, this allows to reduce the quality requirements for the virtual sensors in the system and, thus, save energy. The integration of the coordination-based approach with model driven selection of sensing locations is a promising future research direction.

The performance of the basic symbolic group communication mechanism proposed can be further improved by dynamically integrating network connectivity information between nodes into the connectivity graph that currently is defined solely on static location information. Another promising optimization is to exploit shortcuts for routing in an integrated system of WMN-based symbolic routing and infrastructure-based symbolic routing.

The focus of this dissertation was the automated acquisition of simple sensor data. However, more complex applications such as indoor modeling are feasible with mobile phones. Although such scenarios might require user interaction, e.g., taking pictures of the room that is to be modeled, they also benefit from coordination of sensing. For instance, such a system would greatly benefit from a powerful resource discovery service that allows for complex resource queries and performs discovery in a reactive or proactive manner.

A virtual sensor that returns an indoor model needs to combine multiple measurements, i.e., images, position information, orientation information, and others. Thus, a support system for indoor modeling needs to determine what data needs to be acquired to generate the model accurately and efficiently based on the available resources. The progress of model generation needs to be monitored, and sensing goals need to be adapted to changes in resource availability. Thus, such a system can evaluate complex operators close to the data sources more efficiently than collecting and processing all the sensor data at a central instance.

Another dimension to extend the work of this dissertation is to coordinate the control of actuators in the real world in addition to coordinating sensing. Such actuators might be, for instance, satellite-based navigation systems. If these are fed with detailed information about the traffic flow, traffic can be controlled in a way that goals, such as overall traffic throughput, is increased. A holistic view on monitoring and controlling real world phenomena promises a large potential as a future research direction.

List of Notations

Notation	Definition
N	mobile node
O	mobile object
E	external sensor
v_{maxn}	maximum speed of a mobile node
v_{maxo}	maximum speed of a mobile object
r_{tx}	maximum transmission range of a mobile node's WLAN radio
r_{accuracy}	maximum error of positioning device
r_{range}	sensor range: maximum range of a sensor
p_{detect}	probability of proximity sensor to detect mobile object in sensor range
δ_{read}	time to take a measurement
M	a physical measurement
$pos(X)$	the position associated with X
$area(X)$	the area associated with X
$time(X)$	the time associated with X
$id(X)$	unique identifier of X
$type(X)$	the type of X
$level(X)$	level of location X in the LHT
$d_{\text{geo}}(X, Y)$	distance between symbolic locations X and Y based on LNG
$d_{\text{hier}}(X, Y)$	distance between symbolic locations X and Y based on LHT
$d_{\text{topo}}(X, Y)$	distance between nodes X and Y measured in hops
G	graph representing street network
S	edge of G representing a street segment

List of Figures

List of Tables

List of Algorithms

Publications

The following publications of the author of this dissertation contribute main parts to this dissertation:

[WDR09] Harald Weinschrott, Frank Dürr, and Kurt Rothermel. Efficient Capturing of Environmental Data with Mobile RFID Readers. In *Proceedings of the 10th International Conference on Mobile Data Management (MDM'09)*, pages 41–51, Taipei, Taiwan, May 2009. IEEE Computer Society Conference Publishing Services.

[WDR10a] Harald Weinschrott, Frank Dürr, and Kurt Rothermel. StreamShaper: Coordination Algorithms for Participatory Mobile Urban Sensing. In *Proceedings of the 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'10)*, pages 195–204, San Francisco, CA, USA, November 2010. IEEE.

[WDR10b] Harald Weinschrott, Frank Dürr, and Kurt Rothermel. Symbolic Routing for Location-based Services in Wireless Mesh Networks. In *Proceedings of the IEEE 24nd International Conference on Advanced Information Networking and Applications*, pages 851–858, Perth, Australia, April 2010. IEEE Computer Society.

[WWDR11] Harald Weinschrott, Julian Weisser, Frank Dürr, and Kurt Rothermel. Participatory Sensing Algorithms for Mobile Object Discovery in Urban Areas. In *Proceedings of the 9th Annual IEEE International Conference on Pervasive Computing and Communications*, pages 128–135, Seattle, WA, USA, March 2011. IEEE Computer Society.

The following publications of the author of this dissertation have influenced parts of this dissertation:

[BWDR11] Patrick Baier, Harald Weinschrott, Frank Dürr, and Kurt Rothermel. MapCorrect: Automatic Correction and Validation of Road Maps Using Public Sensing. In *Proceedings of the 36th Annual IEEE Conference on Local Computer Networks (LCN 2011)*, pages 58–66, Bonn, Germany, October 2011. IEEE Computer Society.

[DWM08] Dominique Dudkowski, Harald Weinschrott, and Pedro José Marrón. Design and Implementation of a Reference Model for Context Management in Mobile Ad-Hoc Networks. In *Proceedings of the International Workshop on Data Management for Wireless and Pervasive Communications (DMWPC2008)*, pages 832–837, Gino-wan, Okinawa, Japan, March 2008. IEEE.

[DWR10] Frank Dürr, Harald Weinschrott, and Kurt Rothermel. Geocast Routing of Symbolically Addressed Messages in Wireless Mesh Networks. In *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '10)*, pages 552–557, Mannheim, Germany, March 2010. IEEE Computer Society.

[GHLW09] Matthias Grossmann, Nicola Hönle, Carlos Lübbe, and Harald Weinschrott. An Abstract Processing Model for the Quality of Context Data. In *Proceedings of the 1st International Workshop on Quality of Context*, pages 132–143, Stuttgart, Germany, June 2009. Springer.

[LCG$^+$09] Ralph Lange, Nazario Cipriani, Lars Geiger, Matthias Großmann, Harald Weinschrott, Andreas Brodt, Matthias Wieland, Stamatia Rizou, and Kurt Rothermel. Making the World Wide Space Happen: New Challenges for the Nexus Context Platform. In *Proceedings of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, pages 300–303, Galveston, TX, USA, March 2009. IEEE Computer Society.

[LWG$^+$09] Ralph Lange, Harald Weinschrott, Lars Geiger, Andre Blessing, Frank Dürr, Kurt Rothermel, and Hinrich Schütze. On a Generic Uncertainty Model for Position Information. In *Proceedings of the 1st International Workshop on Quality of Context*, pages 76–87, Stuttgart, Germany, June 2009. Springer.

[MGWR07] Steffen Maier, Andreas Grau, Harald Weinschrott, and Kurt Rothermel. Scalable Network Emulation: A Comparison of Virtual Routing and Virtual Machines. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC'07)*, pages 395–402, Aveiro, Portugal, July 2007. IEEE Computer Society.

Bibliography

[AAB$^+$07] Tarek Abdelzaher, Yaw Anokwa, Peter Boda, Jeff Burke, Deborah Estrin, Leonidas Guibas, Aman Kansal, Samuel Madden, and Jim Reich. Mobiscopes for human spaces. *IEEE Pervasive Computing*, 6(2):20–29, 2007.

[AAR07] Marwan Al-Azzawi and Robert Raeside. Modeling pedestrian walking speeds on sidewalks. *Journal of Urban Planning and Development*, 133(3):211–219, September 2007.

[ACGC09] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, and Romit Roy Choudhury. Phonepoint pen: using mobile phones to write in air. In *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds (MobiHeld'09)*, pages 1–6, Barcelona, Spain, 2009.

[ACRC09] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, MobiCom '09, pages 261–272, New York, NY, USA, 2009. ACM.

[ACVS10] M. Afanasyev, Tsuwei Chen, G.M. Voelker, and A.C. Snoeren. Usage patterns in an urban wifi network. *IEEE/ACM Transactions on Networking*, 18(5):1359 –1372, October 2010.

[AD07] Supreetha Rao Aroor and Daniel D. Deavours. Evaluation of the state of passive uhf rfid: An experimental approach. *IEEE Systems Journal*, 1(2):168–176, December 2007.

[ADF$^+$09] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, and S. Parkvall. Lte: the evolution of mobile broadband. *Communications Magazine, IEEE*, 47(4):44 –51, April 2009.

[Ahm09] S. Ahmadi. An overview of next-generation mobile wimax technology. *Communications Magazine, IEEE*, 47(6):84 –98, June 2009.

[ALM11] Jong Hoon Ahnn, Uichin Lee, and Hyun Jin Moon. Geoserv: A distributed urban sensing platform. In *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 164 –173, May 2011.

[AML$^+$10] G.-S. Ahn, M. Musolesi, H. Lu, R. Olfati-Saber, and A. T. Campbell. MetroTrack: Predictive Tracking of Mobile Events Using Mobile Phones. *Lecture Notes in Computer Science*, 6131:230–+, 2010.

[APG$^+$10] Hossein Ahmadi, Nam Pham, Raghu Ganti, Tarek Abdelzaher, Suman Nath, and Jiawei Han. Privacy-aware regression modeling of participatory sensing data. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 99–112, New York, NY, USA, 2010. ACM.

[ASC$^+$10] Karl Aberer, Saket Sathe, Dipanjan Chakraborty, Alcherio Martinoli, Guillermo Barrenetxea, Boi Faltings, and Lothar Thiele. Opensense: open community driven sensing of environment. In *Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming*, IWGS '10, pages 39–42, New York, NY, USA, 2010. ACM.

[ASSC02] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102–114, August 2002.

[Awe11] Awekas. Automatisches wetterkarten system, September 2011. Available online at `http://www.awekas.at`.

[BBHS03] M. Bauer, C. Becker, J. Hahner, and G. Schiele. Contextcube - providing context information ubiquitously. In *Proceedings of 23rd International Conference on Distributed Computing Systems Workshops*, pages 308–313, 2003.

[BCF$^+$10] Novella Bartolini, Tiziana Calamoneri, Emanuele Guido Fusco, Annalisa Massini, and Simone Silvestri. Push & pull: autonomous deployment of mobile sensors for a complete coverage. *Wirel. Netw.*, 16:607–625, April 2010.

[BCQ$^+$07] Cristiana Bolchini, Carlo A. Curino, Elisa Quintarelli, Fabio A. Schreiber, and Letizia Tanca. A data-oriented survey of context models. *SIGMOD Rec.*, 36:19–26, December 2007.

[BD05] Christian Becker and Frank Dürr. On location models for ubiquitous computing. *Personal Ubiquitous Comput.*, 9(1):20–31, 2005.

[BDR07] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *J. Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[BI04] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive 2004*, pages 1–17, April 2004.

[BKM$^+$04] Jan Beutel, Oliver Kasten, Friedemann Mattern, Kay Römer, Frank Siegemund, and Lothar Thiele. Prototyping wireless sensor network applications with btnodes. In *Proc. 1st European Workshop on Sensor Networks (EWSN 2004)*, pages 323–338. Springer, 2004.

[BMV10] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. Augmenting mobile 3g using wifi. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 209–222, New York, NY, USA, 2010. ACM.

[Bra07] Peter Brass. Bounds on coverage and target detection capabilities for models of networks of mobile sensors. *ACM Trans. Sen. Netw.*, 3(2):9, 2007.

[BW08] Michael Beuttner and David Wetherall. An empirical study of uhf rfid performance. In *Proceedings of MobiCom 2008*, 2008.

[BWDR11] Patrick Baier, Harald Weinschrott, Frank Dürr, and Kurt Rothermel. Map-Correct: Automatic Correction and Validation of Road Maps Using Public Sensing. In *Proceedings of the 36th Annual IEEE Conference on Local Computer Networks (LCN 2011)*, pages 58–66, Bonn, Germany, October 2011. IEEE Computer Society.

[CBD02] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing: Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2:483–502, 2002.

[CBkC$^+$10] Eduardo Cuervo, Aruna Balasubramanian, Dae ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Ch, and Paramvir Bahl. Maui: Making smartphones last longer with code offload. In *In In Proceedings of ACM MobiSys*, 2010.

[CCDU05] A. Caruso, S. Chessa, S. De, and A. Urpi. Gps free coordinate assignment and routing in wireless sensor networks. In *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, 2005.

[CCH$^+$10] Andrew Campbell, Tanzeem Choudhury, Shaohan Hu, Hong Lu, Matthew K. Mukerjee, Mashfiqui Rabbi, and Rajeev D.S. Raizada. Neurophone: brain-mobile phone interface using a wireless eeg headset. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, MobiHeld '10, pages 3–8, New York, NY, USA, 2010. ACM.

[CEB+09] Nazario Cipriani, Mike Eissele, Andreas Brodt, Matthias Grossmann, and Bernhard Mitschang. Nexusds: a flexible and extensible middleware for distributed stream processing. In *Proceedings of the 2009 International Database Engineering & Applications Symposium*, IDEAS '09, pages 152–161, New York, NY, USA, 2009. ACM.

[CEL+08] Andrew T. Campbell, Shane B. Eisenman, Nicholas D. Lane, Emiliano Miluzzo, Ronald A. Peterson, Hong Lu, Xiao Zheng, Mirco Musolesi, Kristóf Fodor, and Gahng-Seop Ahn. The rise of people-centric sensing. *IEEE Internet Computing*, 12(4):12–21, 2008.

[CEM+08] M.E.M. Campista, P.M. Esposito, I.M. Moraes, L.H.M. Costa, O.C.M. Duarte, D.G. Passos, C.V.N. de Albuquerque, D.C.M. Saade, and M.G. Rubinstein. Routing metrics and protocols for wireless mesh networks. *IEEE Network*, 22(1):6–12, January 2008.

[CEQM+04] M. Castillo-Effer, D.H. Quintela, W. Moreno, R. Jordan, and W. Westhoff. Wireless sensor networks for flash-flood alerting. In *Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits and Systems*, volume 1, pages 142 – 146, November 2004.

[CHK08] Dana Cuff, Mark Hansen, and Jerry Kang. Urban sensing: out of the woods. *Commun. ACM*, 51(3):24–33, 2008.

[CJDC10] Donglei Cao, Beihong Jin, Sajal K. Das, and Jiannong Cao. On collaborative tracking of a target group using binary proximity sensors. *J. Parallel Distrib. Comput.*, 70:825–838, August 2010.

[CJNP04] Alminas Civilis, Christian S. Jensen, Jovita Nenortaitė, and Stardas Pakalnis. Efficient tracking of moving objects with precision guarantees. In *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004)*, pages 164–173, Boston, MA, USA, August 2004.

[CKK+08] Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. Anonysense: Privacyaware people-centric sensing. In *Proc. ACM 6th Int'l Conf. on Mobile Systems, Applications and Services (MOBISYS '08)*, 2008.

[CM06] Canfeng Chen and Jian Ma. Mobile enabled large scale wireless sensor networks. *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, 1:333–338, February 2006.

[CMT+08] Sunny Consolvo, David W. McDonald, Tammy Toscos, Mike Y. Chen, Jon Froehlich, Beverly Harrison, Predrag Klasnja, Anthony LaMarca, Louis LeGrand, Ryan Libby, Ian Smith, and James A. Landay. Activity sensing in the wild: a field trial of ubifit garden. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1797–1806, New York, NY, USA, 2008. ACM.

[CSM11] Nazario Cipriani, Oliver Schiller, and Bernhard Mitschang. M-top: multi-target operator placement of query graphs for data streams. In *IDEAS*, pages 52–60, 2011.

[CSR11] CSR. SiRFstarV Architecture and SiRFusion Platform, December 2011. Available online at `http://www.csr.com/products/download/138`.

[Dey01] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5:4–7, January 2001.

[Dia04] Dermot Diamond. Internet-scale sensing. *Analytical Chemistry*, 76(15):278 A–286 A, 2004.

[DK08] Abhimanyu Das and David Kempe. Sensor selection for minimizing worst-case prediction error. In *Proceedings of the 7th international conference on Information processing in sensor networks*, IPSN '08, pages 97–108, Washington, DC, USA, 2008. IEEE Computer Society.

[DMP+10] Tathagata Das, Prashanth Mohan, Venkata N. Padmanabhan, Ramachandran Ramjee, and Asankhaya Sharma. Prism: platform for remote sensing using smartphones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 63–76, New York, NY, USA, 2010. ACM.

[DMR06] Dominique Dudkowski, Pedro Jose Marron, and Kurt Rothermel. An efficient resilience mechanism for data centric storage in mobile ad hoc networks. In *MDM '06: Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, page 7, Washington, DC, USA, 2006. IEEE Computer Society.

[DMR07] Dominique Dudkowski, Pedro José Marrón, and Kurt Rothermel. Migration policies for location-centric data storage in mobile ad-hoc networks. In Hongke Zhang, Stephan Olariu, Jiannong Cao, and David B. Johnson, editors, *Proceedings of the 3rd International Conference on Mobile Ad-hoc and Sensor Networks (MSN'07)*, volume 4864 of *Lecture Notes in Computer Science*, pages 197–208. Springer, 2007.

[DR08] Frank Dürr and Kurt Rothermel. An Adaptive Overlay Network for World-wide Geographic Messaging. In *Proceedings of the 22nd IEEE International Conference on Advanced Information Networking and Applications (AINA 2008)*, pages 875–882, Gino-wan, Okinawa, Japan, March 2008. IEEE.

[Dud09] Dominique Dudkowski. *Fundamental Storage Mechanisms for Location-Based Services in Mobile Ad-hoc Networks*. PhD thesis, University of Stuttgart, 2009.

[Dür10] Frank Dürr. *Geographische Kommunikationsmechanismen auf Basis von feingranularen räumlichen Umgebungsmodellen*. Dissertation, University of Stuttgart, June 2010.

[DVS+06] Hai Deng, M. Varanasi, K. Swigger, O. Garcia, R. Ogan, and E. Kougianos. Design of sensor-embedded radio frequency identification (se-rfid) systems. In *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, pages 792–796, June 2006.

[DWM08] Dominique Dudkowski, Harald Weinschrott, and Pedro José Marrón. Design and Implementation of a Reference Model for Context Management in Mobile Ad-Hoc Networks. In *Proceedings of the International Workshop on Data Management for Wireless and Pervasive Communications (DMWPC2008)*, pages 832–837, Gino-wan, Okinawa, Japan, March 2008. IEEE.

[DWR10] Frank Dürr, Harald Weinschrott, and Kurt Rothermel. Geocast Routing of Symbolically Addressed Messages in Wireless Mesh Networks. In *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '10)*, pages 552–557, Mannheim, Germany, March 2010. IEEE Computer Society.

[EGH+08] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The pothole patrol: using a mobile sensor network for road surface monitoring. In *Proceeding of the 6th international conference on Mobile systems, applications, and services*, MobiSys '08, pages 29–39, New York, NY, USA, 2008. ACM.

[EGHK99] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: scalable coordination in sensor networks. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270, New York, NY, USA, 1999. ACM Press.

[ELC08] Shane B. Eisenman, Nicholas D. Lane, and Andrew T. Campbell. Techniques for improving opportunistic sensor networking performance. In *Proceedings of the 4th IEEE international conference on Distributed Computing in Sensor Systems*, DCOSS '08, pages 157–175, Berlin, Heidelberg, 2008. Springer-Verlag.

[EML$^+$09] Shane B. Eisenman, Emiliano Miluzzo, Nicholas D. Lane, Ronald A. Peterson, Gahng-Seop Ahn, and Andrew T. Campbell. Bikenet: A mobile sensing system for cyclist experience mapping. *ACM Trans. Sen. Netw.*, 6:6:1–6:39, December 2009.

[EPC05] EPCglobal. Epc radio-frequency identity protocols class-1 generation-2 uhf rfid protocol for communications at 860 mhz – 960 mhz version 1.1.0, Dec 2005.

[FBMK08] Christian Frank, Philipp Bolliger, Friedemann Mattern, and Wolfgang Kellerer. The sensor internet at work: Locating everyday items using mobile phones. *Pervasive and Mobile Computing Journal (to be published)*, (to be published), 2008.

[FBO09] M. Fiore and J.M. Barcelo-Ordinas. Cooperative download in urban vehicular networks. In *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, pages 20 –29, October 2009.

[FBRK07] Christian Frank, Philipp Bolliger, Christof Roduner, and Wolfgang Kellerer. Objects calling home: Locating objects using mobile phones. In *Proceedings of the 5th International Conference on Pervasive Computing (Pervasive 2007)*, LNCS, Toronto, Canada, May 2007. Springer.

[FCC$^+$07] Jon Froehlich, Mike Y. Chen, Sunny Consolvo, Beverly Harrison, and James A. Landay. Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys '07, pages 57–70, New York, NY, USA, 2007. ACM.

[FgSPO08] A. Fatah gen. Schieck, A. Penn, and E. O'Neill. Mapping, sensing and visualising the digital co-presence in the public arena. In *Proceedings of the 9th International Conference on Design and Decision Support Systems in Architecture and Urban Planning*, 2008.

[FH11] Daniel Fischer and Klaus Herrmann. Resource management in public sensing - challenges and requirements. *ECEASST*, 2011.

[GBH$^+$05] Matthias Grossmann, Martin Bauer, Nicola Honle, Uwe-Philipp Kappeler, Daniela Nicklas, and Thomas Schwarz. Efficiently managing context information for large-scale scenarios. In *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 331–340, Washington, DC, USA, 2005. IEEE Computer Society.

[GBM08] Dominique Guinard, Oliver Baecker, and Florian Michahelles. Supporting a mobile lost and found community. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 407–410, New York, NY, USA, 2008. ACM.

[GHLW09] Matthias Grossmann, Nicola Hönle, Carlos Lübbe, and Harald Weinschrott. An Abstract Processing Model for the Quality of Context Data. In *Proceedings of the 1st International Workshop on Quality of Context*, pages 132–143, Stuttgart, Germany, June 2009. Springer.

[GL06] B. Gedik and Ling Liu. Mobieyes: A distributed location monitoring service using moving location queries. *IEEE Transactions on Mobile Computing*, 5(10):1384 –1402, October 2006.

[GMH$^+$08] M. Gauger, P.J. Marron, M. Handte, O. Saukh, D. Minder, A. Lachenmann, and K. Rothermel. Integrating sensor networks in pervasive computing environments using symbolic coordinates. In *3rd International Conference on Communication Systems Software and Middleware and Workshops*, pages 564–573, January 2008.

[GPA$^+$10] Raghu K. Ganti, Nam Pham, Hossein Ahmadi, Saurabh Nangia, and Tarek F. Abdelzaher. Greengps: a participatory sensing fuel-efficient maps application. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 151–164, New York, NY, USA, 2010. ACM.

[GSB02] Hans W. Gellersen, Albercht Schmidt, and Michael Beigl. Multi-sensor context-awareness in mobile devices and smart artifacts. *Mob. Netw. Appl.*, 7(5):341–351, 2002.

[Gur91] Klaus-Werner Gurgel. Erfahrungen mit dem satelliten-navigationssystem gps-genauigkeiten an land und auf see. *Ocean Dynamics*, 44:35–49, 1991. 10.1007/BF02226341.

[Gut07] Andreas Gutscher. A Trust Model for an Open, Decentralized Reputation System. In Etalle Sandro and Stephen Marsh, editors, *Trust Management, Proceedings of the Joint iTrust and PST Conferences on Privacy Trust Management* *and Security (IFIPTM 2007)*, volume 238 of *IFIP*, pages 285–300. Springer-Verlag, July 2007.

[Han09] Arun Handa. Mobile data offload for 3g networks, October 2009. Available online at `http://www.intellinet-tech.com/Media/PagePDF/DataOffload.pdf`.

[HB01] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[HBPW08] Richard Honicky, Eric A. Brewer, Eric Paulos, and Richard White. N-smarts: networked suite of mobile atmospheric real-time sensors. In *Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions*, NSDR '08, pages 25–30, New York, NY, USA, 2008. ACM.

[HBZ$^+$06] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. Cartel: a distributed mobile sensor computing system. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 125–138, New York, NY, USA, 2006. ACM.

[HD09] Beverly Harrison and Anind Dey. What have you done with location-based services lately? *IEEE Pervasive Computing*, 8(4):66–70, October 2009.

[HES06] J. Ho, D.W. Engels, and S.E. Sarma. Hiq: a hierarchical q-learning algorithm to solve the reader collision problem. *International Symposium on Applications and the Internet Workshops*, pages 4 pp.–, January 2006.

[HGT11] S.A. Hoseinitabatabaei, A. Gluhak, and R. Tafazolli. udirect: A novel approach for pervasive observation of user direction with mobile phones. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 74 –83, March 2011.

[Hon07] Richard Edward Honicky. Automatic calibration of sensor-phones using gaussian processes. Technical Report Technical Report No. UCB/EECS-2007-34, Electrical Engineering and Computer Sciences University of California at Berkeley, March 2007.

[HW08] Mordechai (Muki) Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, October 2008.

[HWH$^+$10] Juan C. Herrera, Daniel B. Work, Ryan Herring, Xuegang (Jeff) Ban, Quinn Jacobson, and Alexandre M. Bayen. Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568 – 583, 2010.

[ITK+09] Tomoya Ishikawa, Kalaivani Thangamani, Masakatsu Kourogi, Andrew P. Gee, Walterio Mayol-Cuevas, Keechul Jung, and Takeshi Kurata. In-situ 3d indoor modeler with a camera and self-contained sensors. In *VMR '09: Proceedings of the 3rd International Conference on Virtual and Mixed Reality*, pages 454–464, Berlin, Heidelberg, 2009. Springer-Verlag.

[KAB+05] Lakshman Krishnamurthy, Robert Adler, Phil Buonadonna, Jasmeet Chhabra, Mick Flanigan, Nandakishore Kushalnagar, Lama Nachman, and Mark Yarvis. Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 64–75, New York, NY, USA, 2005. ACM.

[Kan10] Eiman Kanjo. Noisespy: A real-time mobile phone platform for urban noise monitoring and mapping. *Mob. Netw. Appl.*, 15:562–574, August 2010.

[KBP+08] Eiman Kanjo, Steve Benford, Mark Paxton, Alan Chamberlain, Danae Stanton Fraser, Dawn Woodgate, David Crellin, and Adrain Woolard. Mobgeosen: facilitating personal geosensor data collection and visualization using mobile phones. *Personal Ubiquitous Comput.*, 12(8):599–607, 2008.

[KBRL09] Eiman Kanjo, Jean Bacon, David Roberts, and Peter Landshoff. Mobsens: Making smart phones smarter. *IEEE Pervasive Computing*, 8(4):50–57, October 2009.

[KHKZ08] Andreas Krause, Eric Horvitz, Aman Kansal, and Feng Zhao. Toward community sensing. In *IPSN '08: Proceedings of the 7th international conference on Information processing in sensor networks*, pages 481–492, Washington, DC, USA, 2008. IEEE Computer Society.

[Kja07] Kristian Ellebaek Kjaer. A survey of context aware middleware. In *Proceedings of IASTED Software Engineering Conference*, August 2007.

[KK00] Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Mobile Computing and Networking*, pages 243–254, 2000.

[KLH04] Yongjin Kim, Jae-Joon Lee, and Ahmed Helmy. Modeling and analyzing the impact of location inconsistencies on geographic routing in wireless networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(1):48–60, 2004.

[KLLK10] A.M. Khan, Y.-K. Lee, S.Y. Lee, and T.-S. Kim. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *5th International Conference on Future Information Technology (FutureTech)*, pages 1 –6, May 2010.

[KLNA09] J. Kukkonen, E. Lagerspetz, P. Nurmi, and M. Andersson. Betelgeuse: A platform for gathering and processing situational data. *IEEE Pervasive Computing*, 8(2):49 –56, April 2009.

[KNLZ07] A. Kansal, S. Nath, Jie Liu, and Feng Zhao. Senseweb: An infrastructure for shared sensing. *IEEE Multimedia*, 14(4):8–13, October 2007.

[KP09] Sunyoung Kim and Eric Paulos. inair: Measuring and visualizing indoor air quality. In *Ubicomp 2009*, 2009.

[KPC$^+$07] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, IPSN '07, pages 254–263, New York, NY, USA, 2007. ACM.

[KSB09] Jonghyun Kim, Vinay Sridhara, and Stephan Bohacek. Realistic mobility simulation of urban mesh networks. *Ad Hoc Netw.*, 7(2):411–430, 2009.

[KSJ$^+$04] Aman Kansal, Arun A. Somasundara, David D. Jea, Mani B. Srivastava, and Deborah Estrin. Intelligent fluid infrastructure for embedded networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, pages 111–124, New York, NY, USA, 2004. ACM.

[KV03] Young-Bae Ko and Nitin H. Vaidya. Anycasting-based protocol for geocast service in mobile ad hoc networks. *Computer Networks*, 41(6):743 – 760, 2003.

[KWM10] Jennifer Kwapisz, Gary Weiss, and Samuel Moore. Activity recognition using cell phone accelerometers. In *Proceedings of the 4th International Workshop on Knowledge Discovery from Sensor Data*, pages 10–18, 2010.

[KZ07] Aman Kansal and Feng Zhao. Location and mobility in a sensor network of mobile phones. In *NOSSDAV 2007: 17th International workshop on Network and Operating Systems Support for Digital Audio & Video*, June 2007.

[LBB05] Seungjoon Lee, Bobby Bhattacharjee, and Suman Banerjeel. Efficient geographic routing in multihop wireless networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 230–241, New York, NY, USA, 2005. ACM Press.

[LBD+05] Benyuan Liu, Peter Brass, Olivier Dousse, Philippe Nain, and Don Towsley. Mobility improves coverage of sensor networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 300–308, New York, NY, USA, 2005. ACM.

[LBP+11] Hong Lu, A. J. Bernheim Brush, Bodhi Priyantha, Amy K. Karlson, and Jie Liu. Speakersense: Energy efficient unobtrusive speaker identification on mobile phones. In *Pervasive'11*, pages 188–205, 2011.

[LCG+09] Ralph Lange, Nazario Cipriani, Lars Geiger, Matthias Großmann, Harald Weinschrott, Andreas Brodt, Matthias Wieland, Stamatia Rizou, and Kurt Rothermel. Making the World Wide Space Happen: New Challenges for the Nexus Context Platform. In *Proceedings of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '09)*, pages 300–303, Galveston, TX, USA, March 2009. IEEE Computer Society.

[LEM+08] Nicholas D. Lane, Shane B. Eisenman, Mirco Musolesi, Emiliano Miluzzo, and Andrew T. Campbell. Urban sensing systems: opportunistic or participatory? In *HotMobile '08: Proceedings of the 9th workshop on Mobile computing systems and applications*, pages 11–16, New York, NY, USA, 2008. ACM.

[LFK07] Lin Liao, Dieter Fox, and Henry Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *Int. J. Rob. Res.*, 26(1):119–134, 2007.

[LLC07] Nicholas D. Lane, Hong Lu, and Andrew T. Campbell. Ambient beacon localization: using sensed characteristics of the physical world to localize mobile sensors. In *Proceedings of the 4th workshop on Embedded networked sensors*, EmNets '07, pages 38–42, New York, NY, USA, 2007. ACM.

[LLEC10] Hong Lu, Nicholas D. Lane, Shane B. Eisenman, and Andrew T. Campbell. Bubble-sensing: Binding sensing tasks to the physical world. *Pervasive and Mobile Computing*, 6(1):58 – 71, 2010.

[LML+10] N.D. Lane, E. Miluzzo, Hong Lu, D. Peebles, T. Choudhury, and A.T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140 –150, September 2010.

[LPL09a] Sangkeun Lee, Sungchan Park, and Sanggoo Lee. A study on issues in context-aware systems based on a survey andservice scenarios. In *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, pages 8 –13, May 2009.

[LPL$^+$09b] Hong Lu, Wei Pan, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. Soundsense: scalable sound sensing for people-centric applications on mobile phones. In *Mobisys '09: Proceedings of the 7th international conference on Mobile systems, applications, and services*, pages 165–178, New York, NY, USA, 2009. ACM.

[LR02] Alexander Leonhardi and Kurt Rothermel. Architecture of a large-scale location service. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, page 465, Washington, DC, USA, 2002. IEEE Computer Society.

[LRL$^+$10] Kyunghan Lee, Injong Rhee, Joohyun Lee, Song Chong, and Yung Yi. Mobile data offloading: how much can wifi deliver? In *Proceedings of the 6th International COnference*, Co-NEXT '10, pages 26:1–26:12, New York, NY, USA, 2010. ACM.

[LWG$^+$09] Ralph Lange, Harald Weinschrott, Lars Geiger, Andre Blessing, Frank Dürr, Kurt Rothermel, and Hinrich Schütze. On a Generic Uncertainty Model for Position Information. In *Proceedings of the 1st International Workshop on Quality of Context*, pages 76–87, Stuttgart, Germany, June 2009. Springer.

[LWKZ08] Azman Osman Lim, Xudong Wang, Youiti Kado, and Bing Zhang. A hybrid centralized routing protocol for 802.11s wmns. *Mob. Netw. Appl.*, 13(1-2):117–131, 2008.

[LYL$^+$10] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 71–84, New York, NY, USA, 2010. ACM.

[LZG$^+$06] Uichin Lee, Biao Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi. Mobeyes: smart mobs for urban monitoring with a vehicular sensor network. *IEEE Wireless Communications*, 13(5):52 –57, October 2006.

[Mai04] Christian Maihöfer. A survey on geocast routing protocols. *IEEE Communications Surveys and Tutorials, 2nd quarter issue*, pages 32–42, 2004.

[MGT$^+$07] Rohan Murty, Abhimanyu Gosain, Matthew Tierney, Andrew Brody, Amal Fahad, Josh Bers, and Matt Welsh. Citysense: A vision for an urban-scale wireless networking testbed. Technical report, Harvard University, 2007.

[MiKAM10] J. Miller, Sun il Kim, M. Ali, and T. Menard. Determining time to traverse road sections based on mapping discrete gps vehicle data to continuous flows. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 615 –620, July 2010.

[MLCOS08] Emiliano Miluzzo, Nicholas D. Lane, Andrew T. Campbell, and Reza Olfati-Saber. Calibree: A self-calibration system for mobile sensor networks. In *Proceedings of the 4th IEEE international conference on Distributed Computing in Sensor Systems*, DCOSS '08, pages 314–331, Berlin, Heidelberg, 2008. Springer-Verlag.

[MLEC07] Emiliano Miluzzo, Nicholas D. Lane, Shane B. Eisenman, and Andrew T. Campbell. Cenceme – injecting sensing presence into social networking applications. *Smart Sensing and Context*, Volume 4793/2007, 2007.

[MLF$^+$08] Emiliano Miluzzo, Nicholas D. Lane, Kristóf Fodor, Ronald Peterson, Hong Lu, Mirco Musolesi, Shane B. Eisenman, Xiao Zheng, and Andrew T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 337–350, New York, NY, USA, 2008. ACM.

[MLG$^+$10] P. Medagliani, J. Leguay, V. Gay, M. Lopez-Ramos, and G. Ferrari. Engineering energy-efficient target detection applications in wireless sensor networks. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 31–39, April 2010.

[MM09] P. Manohar and D. Manjunath. On the coverage process of a moving point target in a non-uniform dynamic sensor field. *IEEE Journal on Selected Areas in Communications*, 27(7):1245 –1255, September 2009.

[MML$^+$08] Mirco Musolesi, Emiliano Miluzzo, Nicholas D. Lane, Shane B. Eisenman, Tanzeem Choudhury, and Andrew T. Campbell. The second life of a sensor: Integrating real-world experience in virtual worlds using mobile phones. In *HotEmNets'08*, 2008.

[MPF$^+$10] Mirco Musolesi, Mattia Piraccini, Kristof Fodor, Antonio Corradi, and Andrew Campbell. Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones. In Patrik Floréen, Antonio

Krüger, and Mirjana Spasojevic, editors, *Pervasive Computing*, volume 6030 of *Lecture Notes in Computer Science*, pages 355–372. Springer Berlin / Heidelberg, 2010.

[MPL$^+$11] Emiliano Miluzzo, Michela Papandrea, Nicholas D. Lane, Andy M. Sarroff, Silvia Giordano, and Andrew T. Campbell. Tapping into the vibe of the city using vibn, a continuous sensing application for smartphones. In *Proceedings of 1st international symposium on From digital footprints to social and community intelligence*, SCI '11, pages 13–18, New York, NY, USA, 2011. ACM.

[MPR08] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys'08*, pages 323–336, 2008.

[MRS$^+$09] Min Mun, Sasank Reddy, Katie Shilton, Nathan Yau, Jeff Burke, Deborah Estrin, Mark Hansen, Eric Howard, Ruth West, and Péter Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 55–68, New York, NY, USA, 2009. ACM.

[MSHA03] L. Munoz, Xiaotian Sun, R. Horowitz, and L. Alvarez. Traffic density estimation with the cell transmission model. In *Proceedings of the 2003 American Control Conference*, volume 5, pages 3750–3755, June 2003.

[MSN$^+$09] Nicolas Maisonneuve, Matthias Stevens, Maria E. Niessen, Peter Hanappe, and Luc Steels. Citizen noise pollution monitoring. In *dg.o '09: Proceedings of the 10th Annual International Conference on Digital Government Research*, pages 96–103. Digital Government Society of North America, 2009.

[MWC10] Emiliano Miluzzo, Tianyu Wang, and Andrew T. Campbell. Eyephone: activating mobile phones with your eyes. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, MobiHeld '10, pages 15–20, New York, NY, USA, 2010. ACM.

[Nav11] Navman. Jupiter 3, September 2011. Available online at `http://www.navmanwirelessoem.com/jupiter3.html`.

[NM04] Daniela Nicklas and Bernhard Mitschang. On building location aware applications using an open platform based on the NEXUS augmented world model. *Software and System Modeling*, 3(4):303–313, 2004.

[OD08] Martina O'Toole and Dermot Diamond. Absorbance based light emitting diode optical sensors and sensing devices. *Sensors*, 8(4):2453–2479, 2008.

[PBRD02] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. IETF Internet draft, Nov. 2002.

[PDR11] Damian Philipp, Frank Dürr, and Kurt Rothermel. A Sensor Network Abstraction for Flexible Public Sensing Systems. In IEEE Computer Society Conference Publishing Services, editor, *Proceedings of the 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, volume E4469 of *IEEE Computer Society Order Number*, pages 460–469, Valencia, Spain, October 2011. IEEE Computer Society Conference Publishing Services.

[Pet09] Michael Peter. Presentation and evaluation of inconsistencies in multiply represented 3d building models. In *Proceedings of the 1st international conference on Quality of context*, QuaCon'09, pages 156–163, Berlin, Heidelberg, 2009. Springer-Verlag.

[PGU$^+$10] Nam Pham, Raghu K. Ganti, Yusuf S. Uddin, Suman Nath, and Tarek F. Abdelzaher. Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing. In Jorge Sá Silva, Bhaskar Krishnamachari, and Fernando Boavida, editors, *EWSN*, volume 5970 of *Lecture Notes in Computer Science*, pages 114–130. Springer, 2010.

[PHH08] E Paulos, R J Honicky, and B Hooker. Citizen science: Enabling participatory urbanism. *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-time City*, 2008.

[PKG$^+$09] Ming-Zher Poh, Kyunghee Kim, A.D. Goessling, N.C. Swenson, and R.W. Picard. Heartphones: Sensor earphones and mobile application for non-obtrusive health monitoring. In *International Symposium on Wearable Computers*, pages 153 –154, September 2009.

[PLL11] Bodhi Priyantha, Dimitrios Lymberopoulos, and Jie Liu. Littlerock: Enabling energy-efficient continuous sensing on mobile phones. *Pervasive Computing, IEEE*, 10(2):12 –15, February 2011.

[PLV$^+$10] Avinash Parnandi, Ken Le, Pradeep Vaghela, Aalaya Kolli, Karthik Dantu, Sameera Poduri, and Gaurav S. Sukhatme. Coarse in-building localization with smartphones. In Ozgur Akan, Paolo Bellavista, Jiannong Cao, Falko Dressler, Domenico Ferrari, Mario Gerla, Hisashi Kobayashi, Sergio Palazzo, Sartaj Sahni, Xuemin (Sherman) Shen, Mircea Stan, Jia Xiaohua, Albert Zomaya, Geoffrey Coulson, Thomas Phan, Rebecca Montanari, and Petros Zerfos, editors, *Mobile Computing, Applications, and Services*, volume 35 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 343–354. Springer Berlin Heidelberg, 2010.

[PRD09] Q. Pan, G. Reitmayr, and T. Drummond. ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition. In *Proc. 20th British Machine Vision Conference (BMVC)*, London, September 2009.

[PRR08] M. Pasquet, J. Reynaud, and C. Rosenberger. Secure payment with nfc mobile phone in the smarttouch project. In *International Symposium on Collaborative Technologies and Systems*, pages 121 –126, May 2008.

[PRS+06] Andrew Parker, Sasank Reddy, Thomas Schmid, Kevin Chang, Ganeriwal Saurabh, Mani Srivastava, Mark Hansen, Jeff Burke, Deborah Estrin, Mark Allman, and Vern Paxson. Network system challenges in selective sharing and verification for personal social, and urban-scale sensing applications. In *In HotNets*, 2006.

[RA10] Suoranta Risto and Lappeteläinen Antti. Operator's dilemma - how to take advantage of mobile internet, May 2010. Available online at `http://www.notava.com/notava/uploads/Whitepapers/Internet_growth_V10.pdf`.

[RB07] O. Riva and C. Borcea. The urbanet revolution: Sensor power to the people! *IEEE Pervasive Computing*, 6(2):41–49, April 2007.

[RDD+03] Kurt Rothermel, Dominique Dudkowski, Frank Dürr, Martin Bauer, and Christian Becker. Ubiquitous Computing - More than Computing Anytime Anyplace? In *Proceedings of the 49. Photogrammetrische Woche.* Stuttgart: ifp, September 2003.

[RES10] Sasank Reddy, Deborah Estrin, and Mani Srivastava. Recruitment Framework for Participatory Sensing Data Collections. In Patrik Floréen, Antonio Krüger, and Mirjana Spasojevic, editors, *Proceedings of the 8th International Conference on Pervasive Computing*, pages 138–155, Berlin, Heidelberg, May 2010. Springer Berlin Heidelberg.

[RKS+03] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, and F. Yu. Data-centric storage in sensornets with GHT, a geographic hash table. *Mob. Netw. Appl.*, 8(4):427–442, August 2003.

[RLBCE11] Nikodin Ristanovic, Jean-Yves Le Boudec, Augustin Chaintreau, and Vijay Erramilli. Energy Efficient Offloading of 3G Networks. In *Proceedings of the 8th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (Best Student Paper Award)*, 2011.

[RM04] K. Römer and F. Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, 2004.

[RMB$^+$10] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Trans. Sen. Netw.*, 6:13:1–13:27, March 2010.

[RNC05] Paul Rudman, Steve North, and Matthew Chalmers. Mobile pollution mapping in the city. In *Proc. UK-UbiNet workshop on eScience and ubicomp, Edinburgh*, May 2005.

[ROE09] Mika Raento, Antti Oulasvirta, and Nathan Eagle. Smartphones: An emerging tool for social scientists. *Sociological Methods Research*, 37(3):426–454, February 2009.

[Rot08] Jörg Roth. Extracting line string features from gps logs. In *5. GI/ITG KuVS Fachgespräch "Ortsbezogene Anwendungen und Dienste"*. Schriftenreihe der Georg-Simon-Ohm-Hochschule Nürnberg, Nürnberg, September 2008.

[SBG99] Albrecht Schmidt, Michael Beigl, and Hans-Werner Gellersen. There is more to context than location. *Computers & Graphics Journal*, 23(6):893–902, December 1999.

[Sky11] Skyetec. Skyemodule m9, September 2011. Available online at `http://www.skyetek.com/Portals/0/Documents/Products/SkyeModule_M9_DataSheet.pdf`.

[SM08] Anthony Steed and Richard Milton. Using tracked mobile sensors to make maps of environmental effects. *Personal Ubiquitous Comput.*, 12(4):331–342, 2008.

[SMR05] I. Stepanov, P.J. Marron, and K. Rothermel. Mobility modeling of outdoor scenarios for manets. In *Proceedings. 38th Annual Simulation Symposium*, pages 312 – 322, April 2005.

[SMR06] Dan Smith, Ling Ma, and Nick Ryan. Acoustic environment as an indicator of social and physical context. *Personal Ubiquitous Comput.*, 10:241–254, March 2006.

[SMUS06] N. Shrivastava, R. Madhow, Mudumbai U., and S. Suri. Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, SenSys '06, pages 251–264, New York, NY, USA, 2006. ACM.

[SOP$^+$04] Robert Szewczyk, Eric Osterweil, Joseph Polastre, Michael Hamilton, Alan Mainwaring, and Deborah Estrin. Habitat monitoring with sensor networks. *Commun. ACM*, 47:34–40, June 2004.

[SRJ10] Ruben Stranders, Alex Rogers, and Nick Jennings. A decentralised coordination algorithm for maximising sensor coverage in large sensor networks. In *The Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 1165–1172, 2010.

[Sto48] Harry Stockman. Communication by means of reflected power. In *Proceedings of the IRE*, volume 36, pages 1196– 1204, October 1948.

[Sum11] Summit. Sdc-cf10g - 802.11g compact flash module with antenna connectors, September 2011. Available online at `http://www.summitdata.com/SDC-CF10G.html`.

[TKK$^+$11] Muhammad Adnan Tariq, Boris Koldehofe, Gerald G. Koch, Imran Khan, and Kurt Rothermel. Meeting subscriber-defined QoS constraints in publish/-subscribe systems. *Concurrency and Computation: Practice and Experience*, 23:2140–2153, December 2011.

[TRL$^+$09] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 85–98, New York, NY, USA, 2009. ACM.

[VD08] Nitin Vaidya and Samir R. Das. Rfid-based networks: exploiting diversity and redundancy. *SIGMOBILE Mob. Comput. Commun. Rev.*, 12(1):2–14, 2008.

[VG06] Maarten Vergauwen and Luc Van Gool. Web-based 3d reconstruction service. *Mach. Vision Appl.*, 17(6):411–426, 2006.

[Wan11] Yi Wang. *Towards energy efficient mobile sensing.* PhD thesis, University of Southern California, 2011.

[WBT$^+$10] Daniel B. Work, Sébastien Blandin, Olli-Pekka Tossavainen, Benedetto Piccoli, and Alexandre M. Bayen. A traffic model for velocity data assimilation. *Applied Mathematics Research eXpress*, 2010(1):1–35, 2010.

[WDR09] Harald Weinschrott, Frank Dürr, and Kurt Rothermel. Efficient Capturing of Environmental Data with Mobile RFID Readers. In *Proceedings of the 10th International Conference on Mobile Data Management (MDM'09)*, pages 41–51, Taipei, Taiwan, May 2009. IEEE Computer Society Conference Publishing Services.

[WDR10a] Harald Weinschrott, Frank Dürr, and Kurt Rothermel. StreamShaper: Coordination Algorithms for Participatory Mobile Urban Sensing. In *Proceedings of the 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'10)*, pages 195–204, San Francisco, CA, USA, November 2010. IEEE.

[WDR10b] Harald Weinschrott, Frank Dürr, and Kurt Rothermel. Symbolic Routing for Location-based Services in Wireless Mesh Networks. In *Proceedings of the IEEE 24nd International Conference on Advanced Information Networking and Applications*, pages 851–858, Perth, Australia, April 2010. IEEE Computer Society.

[Wei91] Mark Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, September 1991.

[Whi87] R.M. White. A sensor classification scheme. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control Society*, 34(2):124–126, March 1987.

[WLA$^+$09] Yi Wang, Jialiu Lin, Murali Annavaram, Quinn A. Jacobson, Jason Hong, Bhaskar Krishnamachari, and Norman Sadeh. A framework of energy efficient mobile sensing for automatic user state recognition. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 179–192, New York, NY, USA, 2009. ACM.

[WWDR11] Harald Weinschrott, Julian Weisser, Frank Dürr, and Kurt Rothermel. Participatory Sensing Algorithms for Mobile Object Discovery in Urban Areas. In *Proceedings of the 9th Annual IEEE International Conference on Pervasive Computing and Communications*, pages 128–135, Seattle, WA, USA, March 2011. IEEE Computer Society.

[YKG10] Tingxin Yan, Vikas Kumar, and Deepak Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, MobiSys '10, pages 77–90, New York, NY, USA, 2010. ACM.

[YMG08] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52:2292–2330, August 2008.

[ZCM$^+$10] Farid Zaid, Diego Costantini, Parag Mogre, Andreas Reinhardt, Johannes Schmitt, and Ralf Steinmetz. Wbroximity: Mobile participatory sensing for wlan- and bluetooth-based positioning. In *Proceedings of the 5th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2010)*, pages 906–912. Omnipress, October 2010.

[ZS05] Thomas Zahn and Jochen Schiller. MADPastry: A DHT Substrate for Practicably Sized MANETs. In *Proc. of 5th Workshop on Applications and Services in Wireless Networks (ASWN2005)*, Paris, France, June 2005.

[ZWHS09] Yue Zhou, Jiangyan Wang, Di Huang, and Shengyang Sun. Pedestrian simulation modeling for world expo 2010 shanghai. *Journal of Transportation Systems Engineering and Information Technology*, 9(2):141 – 146, 2009.