Energiebewusstes und selbstorganisiertes Clustering drahtloser Sensornetzwerke

Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.) der Fakultät für Informatik und Elektrotechnik der Universität Rostock

vorgelegt von: Jakob Salzmann, geb. am 30. März 1981 in Rostock

urn:nbn:de:gbv:28-diss2012-0120-2

Gutachter:

- Prof. Dr. Dirk Timmermann (Universität Rostock, Fakultät für Informatik und Elektrotechnik, Institut für angewandte Mikroelektronik und Datentechnik)
- Prof. Dr. Stefan Fischer (Universität zu Lübeck, Institut für Telematik)
- Prof. Dr. Theo Ungerer (Universität Augsburg, Institut für Informatik)

Tag der Einreichung: 30.05.2012 Tag der Verteidigung: 05.11.2012

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit im Institut für Angewandte Mikroelektronik und Datentechnik der Universität Rostock.

An dieser Stelle möchte ich mich daher ausdrücklich bei meinem Doktorvater Herrn Prof. Dr.-Ing. Dirk Timmermann für die Möglichkeit bedanken, eine solche Arbeit zu verfassen. Jedoch geht auch Dank an seine Geduld seine motivierenden Worte und das angenehme Arbeitsklima, an dem er als Institutsdirektor erheblichen Anteil hat.

Danken möchte ich außerdem insbesondere meinem langjährigen Bürokollegen Ralf Behnke für die fruchtbare Zusammenarbeit und dem wechselseitigen Gedankenaustausch. Dank gilt selbstverständlich auch den anderen Kollegen und den Studenten, die zu dieser Arbeit beigetragen haben.

Dank gilt auch dem nicht wissenschaftlich arbeitendem Personal, welches für eine stets harmonische Arbeitsatmosphäre sorgte. Schließlich gilt mein Dank meinen Eltern, Anke und Holger Salzmann, die mich auf meinem nunmehr 24 Jahre währendem Bildungsweg unermüdlich begleitet haben und ohne deren Erziehung, Ratschläge und Weisheiten diese Arbeit wohl niemals zustande gekommen wäre.

Abschließend möchte ich allen Menschen danken, die mich auf meinem bisherigen Lebensweg begleitet und mich damit direkt und indirekt unterstützt haben.

Inhaltsverzeichnis

A	Abbildungsverzeichnis		
T	Tabellenverzeichnis17		
A	Abkürzungsverze	ichnis	18
K	Kurzreferat		20
A	Abstract		21
1	Einleit	ung	22
2	Grund	lagen	25
	2.1 Drahtlose	Kommunikation	
	2.1.1 Über	tragungsmedium	
	2.1.2 Bitül	pertragungsschicht	
	2.1.3 Siche	erungsschicht	
	2.1.4 Vern	nittlungsschicht	30
	2.1.5 Höhe	ere Schichten	30
	2.2 Grundlag	en drahtloser Sensornetzwerke	30
	2.2.1 Aufb	au drahtloser Sensornetzwerke	30
	2.2.2 Platz	ierung der Sensorknoten	
	2.2.3 Ener	gieaufnahme	
	2.2.4 Anw	endungsgebiete	
	2.3 Herausfor	derungen drahtloser Sensornetzwerke	39
	2.3.1 Orga	nisation	
	2.3.2 Loka	lisierung	40
	2.3.3 Med	enzugriff	
	2.3.4 Rout	ing	
	2.3.5 Kom	munikationsabdeckung	
	2.3.6 Sens	orabdeckung	
	2.4 Grundlag	en des Clusterings	
	2.4.1 Zeitl	iches Clustering	
	2.4.2 Geog	graphisches Clustering	
	2.5 Kurzzusa	mmenfassung	51
3	Selbsto	organisation in drahtlosen Sensornetzwerken	52
	3.1 Grundlag	en der Selbstorganisation	53
	3.2 Selbsthei	ung	54
	3.2.1 Präv	entive Heilung	56

	3.2.2 Reaktive Heilung	
	3.3 Kurzzusammenfassung	57
4	4 Evaluierung von Algorithmen für drahtlose Sensorr	netzwerke 58
	4.1 Netzwerksimulator	
	4.1.1 Anzahl simulierter Sensornetzwerke	
	4.1.2 MAC-Modell	59
	4.1.3 Routing-Modell	59
	4.1.4 Energiemodell	59
	4.1.5 Kanalmodell	60
	4.1.6 Szenarien	61
	4.1.7 Sensorreichweite	
	4.1.8 Randgebiete	
	4.2 Netzwerkbewertung	
	4.2.1 Realistic Case Coverage	
	4.2.2 Worst Case Coverage	
	4.2.3 Node efficiency	
	4.2.4 Lebenszeit	
	4.3 Kurzzusammenfassung	
5	5 Lokalisierung drahtloser Sensorknoten	69
	5.1 Heuristische Verfahren	
	5.1.1 Schwerpunktbestimmung	
	5.1.2 Gewichtete Schwerpunktbestimmung	
	5.1.3 Adaptive gewichtete Schwerpunktbestimmung	
	5.1.4 Vergleich der Algorithmen unter realitätsnahen Bedi	ngungen 72
	5.2 Exakte Verfahren	
	5.2.1 Trilateration	
	5.2.2 Multilateration und Methode der kleinsten Fehlerqua	drate 76
	5.2.3 Multilateration unter realitätsnahen Bedingungen	
	5.3 HyPAERLoc	79
	5.3.1 Geeignete Auswahl der Beaconpaare	
	5.3.2 Optimierung nicht plausibler Distanzschätzungen	
	5.3.3 Hybride Lokalisierung	
	5.4 Positionsschätzung unter realitätsnahen Bedingungen	
	5.5 Kurzzusammenfassung	89
6	6 Positionsbasiertes Clustering	
	6.1 XGAE	01
	6.1.1 Quadrat	
	6.1.1 Quadrat 6.1.2 Gleichseitiges Dreieck	

	6.1.4	Unregelmäßiges Fünfeck	94
	6.1.5	Vergleich der Zellformen	95
	6.2 N	MASCLE	. 100
	6.2.1	2 – MASCLE	. 100
	6.2.2	4 – MASCLE	. 104
	6.2.3	Hex – MASCLE	. 111
	6.3 V	Vergleich ausgewählter Algorithmen	116
	6.4 Z	Zwischenfazit	122
7		Lokalisierungsfreies Clustering	124
	7.1 H	Beacon Based Clustering	124
	7.1.1	Aufteilung der Areale	125
	7.1.2	Beaconabstand	126
	7.1.3	Minimale Sendereichweite der Sensorknoten	126
	7.1.4	Optimale Sendereichweite der Beacons	127
	7.2 H	Free-GAF	. 130
	7.2.1	Informationsphase	131
	7.2.2	Clusteringphase	. 131
	7.2.3	Bestätigungsphase	. 131
	7.2.4	Anwendung im Sensornetzwerk	. 132
	7.3 H	Free-CLASH	. 133
	7.3.1	Anpassen der Reichweite des Possible Clusterhead Broadcas	t 134
	7.3.2	Selektive Auswahl neuer Clusterheads	. 135
	7.3.3	Beschleunigung der Clusterheadwahl für eine gleichm	äßige
	Clust	erausbreitung	. 136
	7.3.4	Vermeiden von Cross-Couplings	. 137
	7.3.5	Anwendung im Sensornetzwerk	. 138
	7.4 V	Vergleich ausgewählter Algorithmen	. 141
	7.5 2	Zwischenfazit	146
8		Zusammenfassung und Ausblick	. 148
	8.1 A	Ausblick	149
L	literatur	verzeichnis	152
A			162
	Thesen		162
	Liste de	er wissenschaftlichen Veröffentlichungen	165

Abbildungsverzeichnis

Abbildung 1.1 Struktur der Arbeit; Rot hervorgehoben sind Abschnitte, die wesentliche eigene Beiträge enthalten. Gelb hinterlegt sind Algorithmen, welche im Zuge der Arbeit entwickelt und analysiert wurden
Abbildung 2.1 OSI-Modell für die Rechnerkommunikation
Abbildung 2.2 Beispiele und Größenvergleich aktueller Sensorknoten, (A) Java Sunspot, (B) Mica2Dot, (C) Tmote Sky
Abbildung 2.3 Schematischer Aufbau eines Sensorknotens
Abbildung 2.4 Energie/Bit im Vergleich zur eingestellten Sendeleistung am Transmitter nach Formel (2.5) und den Datenblättern [3], [7], [9]
Abbildung 2.5 Theoretisch erreichbare Sendereichweite im Vergleich zur eingestellten Sendeleistung am Transmitter für $G_T = 1$, $G_R = 1$
Abbildung 2.6 Benötigte Energie je Bit im Vergleicht zu theoretisch erreichbaren Sendereichweite für $G_T = 1$, $G_R = 1$
Abbildung 2.7 Clustering eines Sensornetzwerks, (A) Ungeclustertes Netzwerk, (B) Zeitliches Clustering nach [5], (C) geographisches Clustering mit einfacher Rollenzuweisung, (D) Geographisches Clustering mit komplexer Rollenzuweisung
Abbildung 2.8 Redundanzerkennung bei CPNS: (A) Sensorfläche und für die Redundanzerkennung relevante Nachbarn von K, (B) Knoten K empfängt Broadcast eines Nachbarn und ermittelt draus die Sponsored Coverage des Nachbarn, (C) Die Sponsored Coverages alle Nachbarn bedecken die gesamte Sensorfläche von K, K ist redundant
Abbildung 2.9 Angewandter GAF Algorithmus, (A) Ermittlung der maximalen Zellgröße, (B) Routing im geclustertem Netzwerk
Abbildung 3.1 Knoten unterschiedlicher Beschädigungsgrade und erlaubte Rollen
Abbildung 4.1 Vergleich der ermittelten Leistung am Empfänger bei verschiedenen Kanalmodellen; eingestellte Parameter: $f=900 \text{ MHz}, G_T=G_R=1$ $P_T=0 \text{ dBm}$

Abbildung 4.5 Wahrscheinlichkeit kompletter Netzwerkfunktionalität zur Laufzeit des Sensornetzwerks (Lebenszeit) für zwei Beispielalgorithmen...... 67

Abbildung 5.3 Boxplotdarstellung der Ergebnisse des AWCL-Verfahrens in verschiedenen Szenarien, (A) Flacher Sandstrand, (B) Trockenes Unterholz. 74

Abbildung 5.7 Boxplotdarstellung der Ergebnisse einer Multilateration in verschiedenen Szenarien, (A) Flacher Sandstrand (B) Trockenes Unterholz ... 79

Abbildung 5.10 Arithmetisches Mittel der Ergebnisse einer realitätsnahen Lokalisierung mittels Multilateration und angepassten nicht plausiblen Distanzen in den Szenarien (A) Flacher Sandstrand, (B) Trockenes Unterholz

Abbildung 6.1 Angewandter SXGAF Algorithmus, (A) Ermittlung der maximalen Zellgröße in Abhängigkeit von Sensor- und Sendereichweite, (B) Mögliche Anordnung der Zellen, Zellgröße basiert auf der Sensorreichweite 92

Abbildung 6.2 Angewandter TXGAF Algorithmus, (A) Ermittlung der maximalen Zellgröße in Abhängigkeit von Sensor- und Sendereichweite, (B) Mögliche Anordnung der Zellen, Zellgröße basiert auf der Sensorreichweite 93

Abbildung 6.10 Anteil von mit XGAF Varianten geclusterten Sensornetzwerken mit 100% Netzwerkfunktionalität (Worst Case Coverage) 98

Abbildung 6.19 Energieeffiziente Heilung in asymmetrischem 4-MASCLE mittels Assistance Arrangements, (A-D) mit 4 Zellen; (E) Von links nach rechts sinkend: Assistance Arrangements sortiert nach Energieeffizienz..... 110

Abbildung 6.23 Assistierte Fläche ausgefallener Zellen in (A) Hex-MASCLE³, (B) Hex-MASCLE⁴; (C) Vergleich der Flächen aller MASCLE-Varianten. 114

Abbildung 6.25 Realistic Case Coverage positionsbasierter Clusteringverfahren für ideale Bedingungen über verschiedene Anzahlen platzierter Knoten 117

Abbildung 7.2 Größe der entstehenden Zellen im Verhältnis zur Fläche, die von 4 Beacons aufgespannt wird, bei verschiedenen Sendereichweiten der Beacons. 126

Abbildung 7.5 Anteil komplett abgedeckter Sensornetzwerke mit BBC- oder SXGAF -Clustering für unterschiedlicher Anzahl platzierter Sensorknoten . 129

Abbildung 7.7 Clustering eines Sensornetzwerkes mittels Free-GAF (Initialwert $C_W=1$, $t_{Add}=1$, $t_{red}=1$), (A) Ausgangssituation, (B) Broadcast der Possible Clusterhead Nachricht, (C) Broadcast der Join Cluster Nachricht, (D) Broadcasts der Cluster Affirmation Nachrichten, (E-G) Zweiter Clusterhead wiederholt die Prozedur, (H) Fertig geclustertes Netzwerk...... 133

Abbildung 7.9 Nicht-selektive versus selektive Clusterheadwahl,
(A) Ausgangs-situation, (B) Entstehung von zwei weiteren Clusterheads,
(C) Loch in der Sensorabdeckung durch ungünstige Wahl aktiver Knoten,
(D) Selektion der Possible Clusterheads, (E-F) Entstehung von eng vermaschten Clustern ohne Möglichkeit für Löcher in der Sensorabdeckung135

Abbildung 7.11 Entstehung und Vermeidung von Cross-Couplings, (A) Beschleunigte Auswahl eines nahen Possible Clusterheads durch drei benachbarte Cluster, (B) Entstehung eines Cross Couplings, (C) Verzögerung von Possible Clusterheads dessen Nachbarn alles miteinander benachbart sind. Abbildung 7.12 Flussdiagramm der lokalisierungsfreien Clusteringalgorithmen Abbildung 7.13 Flussdiagramm der Regeln zur Auswahl eines geeigneten Clusterheads, (A) Free-GAF, (B) Free-CLASH, wie sie in Abbildung 7.13 Abbildung 7.14 Anwendung der lokalisierungsfreien Clusteringalgorithmen Beispielnetzwerk; (A) Zuordnung der Knoten zu ihren Clustern, auf ein der Clusterheads, (B) Voronoi-Diagramm (C) Voronoidiagramm der Abbildung 7.15 Realistic Case Coverage untersuchter Clusteringverfahren für Abbildung 7.16 Realistic Case Coverage untersuchter Clusteringverfahren für das Szenario Flacher Sandstrand über verschiedene Anzahlen platzierter Abbildung 7.17 Realistic Case Coverage untersuchter Clusteringverfahren für das Szenario Trockenes Unterholz über verschiedene Anzahlen platzierter Abbildung 7.18 Node Efficiency der lokalisierungsfreien Clusteringalgorithmen sowie weiterer Vergleichsalgorithmen über Abbildung 7.19 Wahrscheinlichkeit kompletter Netzwerkfunktionalität verschiedener Clusteringalgorithmen bei idealen Bedingungen über der Abbildung 7.20 Anteil aktiver Knoten zur Gesamtzahl platzierter Knoten verschiedener untersuchter Clusteringalgorithmen über der Laufzeit des Abbildung 7.21 Wahrscheinlichkeit kompletter Netzwerkfunktionalität verschiedener Clusteringalgorithmen im Szenario Flacher Sandstrand über die

Tabellenverzeichnis

TABELLE 2.1 EFFEKTE IM FUNKKANAL (NACH [2])				
TABELLE 2.2 EIGENSCHAFTEN AUSGEWÄHLTER SENSORKNOTEN* 33				
TABELLE 2.3 EIGENSCHAFTEN AUSGEWÄHLTER SENSOREN*				
TABELLE 3.1 EIGENSCHAFTEN SELBSORGANISIERTER SYSTEME NACH [4] 52				
TABELLE 4.1 KANALMODELL DER SIMULIERTEN SZENARIEN NACH [10]6				
TABELLE 5.1 ANPASSUNG DER NICHT PLAUSIBLEN DISTANZSSCHÄTZUNGEN 83				
Tabelle 6.1 Eigenschaften von XGAF Zellen bei verschiedenen Optimierungen				
Tabelle 6.2 Simulationsumgebung für den Vergleich dei Eigenschaften von XGAF Zellen unter idealen Bedingungen				
TABELLE 6.3 COVERAGE-EIGENSCHAFTEN EINER MAXIMAL DIMENSIONIERTEN XGAF ZELLE (DIMENSION IN ABHÄNGIGKEIT VON DER SENSORREICHWEITE). 99				
Tabelle 6.4 Simulationsumgebung für den Vergleich dei				
EIGENSCHAFTEN VON VERSCHIEDENEN POSITIONSBASIERTEN				
CLUSTERINGALGORITHMEN				
Tabelle 7.1 Simulationsumgebung für den Vergleich dei				
Eigenschaften von BBC und SXGAF Zellen unter idealer				
BEDINGUNGEN				
Tabelle 7.2 Simulationsumgebung für den Vergleich dei				
EIGENSCHAFTEN VON VERSCHIEDENEN REICHWEITENBASIERTEN				
Clusteringalgorithmen				

Abkürzungsverzeichnis

2-MASCLE	Dual Phase-Mutual Assistance in a Cluster Environment
4-MASCLE	Quadruple Phase-Mutual Assistance in a Cluster Environment
AODV	Ad hoc On-Demand Distance Vector
AWCL	Adaptive Weighted Centroid Localization
СДМА	
CL	
CPNS	Coverage Preserving Node Scheduling
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
FDMA	
Free-CLASH	LocalizationFree-Clusteringwith Approximation to Symmetric Hexagons
Free-GAF	Localization Free-Geographical Adaptive Fidelity
GAF	Geographical Adaptive Fidelity
HEED	Hybrid Energy-Efficient Distributed Clustering
Hex-MASCLE	Hexagon based- Mutual Assistance in a Cluster Environment
HXGAF	
HyPAERLoc	
LEACH	Low-Energy Adaptive Clustering Hierarchy
MAC	
MPL	
NBL	Next Beacon as Linearizer
NBLAD	Next Beacon as Linearizer with Adjusted Distancies
NIN	
OSI-Modell	Open Systems Interconnection Reference Model
PXGAF	
RSSI	
sDLS	scalable Distributed Least Squares
SDMA	Space Division Multiple Access
SIN	
S-MAC	Sensor-MAC
SWL	
SXGAF	
TDMA	
TIN	
TXGAF	Triangle XGAF
WCL	
XGAF	Extended Geographical Adaptive Fidelity

Kurzreferat

Drahtlose Sensornetzwerke sind eine der bedeutendsten Technologien des 21. Jahrhunderts, da sie es uns erlauben, neue Sichtweisen auf verschiedenste Phänomene durch eine weiträumige Erfassung zu gewinnen.

Hauptkomponente des drahtlosen Sensornetzwerks ist der Sensorknoten, welcher Daten erfasst und Nachrichten über detektierte Phänomene mit Hilfe anderer Sensorknoten intelligent zu einer zentralen Senke leitet.

Ein Sensorknoten vereint dabei Sensoreinheit, Funkeinheit, Batterie und Microcontroller auf nicht mehr als ein paar Kubikzentimetern. Erst die Entwicklungen der letzten Jahre im Bereich der Mikroelektronik haben den Bau solcher Miniaturgeräte möglich gemacht.

Da eine möglichst geringe Größe eines Sensorknotens und damit sein Einsatzspektrum häufig durch die Größe seiner Batterien begrenzt ist, ist es unerlässlich, Konzepte zu entwickeln und zu optimieren, welche den Energiebedarf der Knoten auf ein Minimum reduzieren. Gleichzeitig darf die Fähigkeit eines Sensornetzwerks, Phänomene zu detektieren, dabei nicht eingeschränkt werden.

Zentrales Ergebnis dieser Arbeit sind daher Konzepte des Clusterings, welche durch intelligentes, temporäres Abschalten redundanter Knoten, die Energie der abgeschalteten Knoten für einen späteren aktiven Einsatz sparen. Auf Basis der zur Verfügung stehenden Informationen wurden die Verfahren dabei in positionsbasierte und lokalisierungsfreie Algorithmen unterteilt.

Bei den positionsbasierten Algorithmen gelang es, mit den entwickelten MASCLE (Mutual Assistance in a Cluster Environment) Verfahren eine Lebenszeitsteigerung von bis zu 105% gegenüber vergleichbaren Verfahren zu erreichen.

Bei den lokalisierungsfreien Verfahren wurde mit dem reichweitenbasierten Free-CLASH (Localization Free Clustering with Approximation to Symmetric Hexagons) eine Lebenszeitsteigerung um 30% erreicht.

Die entwickelten Algorithmen basieren dabei auf geographischen Clustern mit untereinander austauschbaren Knoten, so dass auch ein effizientes Routing davon profitiert.

Ein weiterer Beitrag dieser Arbeit ist ein hybrides Lokalisierungsverfahren für drahtlose Sensornetzwerke, welches es ermöglicht, in realistischen Umgebungen Ausreißer bei der Lokalisierung zu minimieren und gegenüber Vorgängerverfahren eine Genauigkeitssteigerung von mindestens 15% zu erzielen.

Abstract

Wireless Sensor Networks are one of the key technologies of the 21th century and allow us to gain detailed information about phenomena from a completely different measurement perspective.

Main component of a wireless sensor network is the sensor node, which collects data and forwards the collected data with the help of other participants of the network to a central computer.

Each sensor node combines sensor system, wireless communication, microcontroller and battery on only a few cubic centimeters. Only the recent developments in the area of microelectronics allow the production of such tiny electronic devices.

Due to the fact, that the minimum size and hence the field of application of a sensor node is mostly limited by the size of its batteries, it is essential to develop and to optimize concepts, which reduce the energy consumption to a minimum. These energy conserving concepts must not decrease the sensor or communication abilities of the overall network.

Accordingly, the main achievement of this thesis are clustering concepts, which allow a temporary switch-off of a fraction of nodes and save their energy for a later role as active participant of the network. Based on the available information, the concepts are divided into position-based and localization-free clustering algorithms.

By applying the position-based MASCLE (Mutual Assistance in a Cluster Environment) algorithms, a lifetime increase of 105% compared to similar preceding algorithms is possible. In the category of localization-free algorithms, the developed algorithm Free-CLASH (Localization Free Clustering with Approximation to Symmetric Hexagons) achieves a lifetime increase of 30%.

All developed clustering algorithms are based on geographic clustering with exchangeable nodes in each cluster. Hence, an energy-aware routing can also benefit from the algorithms.

An additional contribution of the thesis is the development of a hybrid localization algorithm, which allows eliminating outliers in realistic environments by combination of heuristic and exact localization algorithm. Furthermore, the algorithm achieves an average localization precision increase of at least 15% compared to preceding algorithms.

1 Einleitung

Die Fortschritte in der Miniaturisierung und der Mikroelektronik führen zu immer leistungsfähigeren, billigeren und kleineren Mikroprozessoren. Gleichzeitig sinkt die notwendige Größe von Sensoren und Bauelementen für eine drahtlose Kommunikation. Werden Mikroprozessor, Energieversorgung, drahtlose Kommunikation und Sensoren auf einem einzelnen Gerät integriert, so spricht man von einem Sensorknoten. Die Größe derartiger Sensorknoten liegt zur Zeit bereits im Kubikzentimeter Bereich, Zukunftsvisionen gehen aber sogar von staubkorngroßen Knoten aus, der auch als "Smart Dust" bezeichnet wird [12].

Die Möglichkeit, solche Miniatursensorsysteme herzustellen, führte bereits Ende des letzten Jahrhunderts zum Begriff der Sensornetzwerke, also Netzwerke, welche aus drahtlos kommunizierenden und möglichst selbstorganisierten Sensorknoten bestehen. Die Idee ist dabei, in einem Gebiet zahlreiche Sensorknoten zu platzieren, die deutlich differenziertere Messungen aufnehmen und Phänomene erkennen können, als es von einer zentralen Stelle aus der Fall wäre. Die Verwendung von drahtlos kommunizierenden Sensorknoten bietet dabei zudem den enormen Vorteil, dass sie auch in unzugänglichen Gebieten platziert werden können, beispielsweise durch den Abwurf aus einem Flugzeug. Da so auf eine kostenintensive Verdrahtung verzichtet werden kann, können im Vergleich zu drahtgebundenen Lösungen enorme Kosten gespart werden. Zusätzlich kann die Ausbringung praktisch ohne zeitlichen Aufwand erfolgen. Ohne Verbindungsleitungen und dank ihrer geringen Größe ist außerdem gewährleistet, dass die Sensorknoten nur minimalen Einfluss auf die Umgebung haben und nicht allein durch ihre Anwesenheit das zu überwachende System empfindlich stören.

Dank ihrer hohen Flexibilität sind drahtlose Sensornetzwerke ideal geeignet, um Überwachungsaufgaben im Außenbereich zu übernehmen. Zahllose Szenarien sind vorstellbar, beispielsweise das Aufspüren und Melden von entstehenden Waldbränden mittels Gas- und Temperatursensoren, die Überwachung von Deichen mittels Feuchtigkeitssensoren, die Kontrolle von Agrarflächen mittels der Messung von Düngergehalt oder Bodenfeuchte, und nicht zuletzt das Aufspüren und Verfolgen von Fahrzeugen.

Prototypisch sind Sensornetzwerke bereits zum Einsatz gekommen. So befindet sich das wohl bekannteste eingesetzte Sensornetzwerk auf Great Duck Island, um die Brutplätze und den Lebensraum der dort heimischen Vögel zu überwachen [13]. Ein weiteres Sensornetzwerk wurde in Ecuador rund um den Vulkan Tungurahua installiert, um dort seismische Aktivitäten zu überwachen und aufzuzeichnen [14].

Doch obwohl diese sowie weitere Projekte bereits die Effizienz und die Einsetzbarkeit von Sensornetzwerken unter Beweis gestellt haben, ist man aufgrund verschiedener Herausforderungen derzeit noch weit davon entfernt, hunderte von sich selbstorganisierenden Sensorknoten zufällig im Observationsgebiet zu platzieren.

Motivation der Arbeit

Eine der größten Herausforderungen für den massiven Einsatz selbstorganisierter Sensorknoten ist die Energieversorgung. So kann der nicht unerhebliche Leistungsbedarf von Mikroprozessor, der Funkeinheit und teilweise auch der Sensoreinheiten kaum durch Versorgungsleitungen sichergestellt werden, sondern wird durch Batterien gewährleistet. Da diese jedoch nur über begrenzte Kapazität verfügen, ist die Lebenszeit eines Sensornetzwerks begrenzt. Würde man leistungsfähigere Batterien verwenden, nähmen die Größe der Batterien und damit das Gewicht und die Größe des Sensorknotens entsprechend zu. Dies steht konträr zu der Idee der aus der Luft abgeworfenen, unauffälligen Sensorknoten. Zudem stellt die Batterie eine erhebliche Schadstoffbelastung für die Umwelt dar. Ziel dieser Arbeit ist es daher, die Lebensdauer von Sensornetzwerken auf algorithmischer Ebene zu erhöhen. Dabei konzentriert sich die Arbeit hauptsächlich auf die Erkennung und Ausnutzung von Redundanzen im Sensornetzwerk, welche durch günstige oder ungünstige Platzierungen von Knoten entstehen. Durch geschicktes Clustering nahe beieinander liegender Knoten wird so erkannt, welche Knoten für den laufenden Netzbetrieb benötigt werden und welchen Knoten es erlaubt werden kann, sich temporär in einen energiesparenden Schlafmodus zu begeben. Zusätzlich werden Strategien beschrieben, mit denen sich die Energiebalance im Gesamtnetz verbessern lässt. Mit den beschriebenen und analysierten Methoden ist es möglich, die Lebensdauer und Zuverlässigkeit eines selbstorganisierten Sensornetzwerks im Vergleich zu bisherigen Strategien erheblich zu steigern.

Struktur der Arbeit

Die vorliegende Arbeit unterteilt sich in 4 Abschnitte, welche in Abbildung 1.1 dargestellt sind. Die mit Rot hervorgehobenen Kapitel enthalten wesentliche eigene Beiträge.

Abschnitt I führt den Leser in die Thematik ein. Nach einer Einleitung in Kapitel 1 beginnt Kapitel 2 damit, dem Leser die für das Verständnis dieser Arbeit nötigen Grundlagen zu vermitteln. Kapitel 3 erklärt den Begriff der Selbstorganisation im Bezug auf Sensornetzwerke. Abschnitt II widmet sich den Voraussetzungen für eine realitätsnahe Analyse entwickelter Clusteringverfahren. Kapitel 4 beschreibt Simulationsmethoden und entwickelt Parameter für Simulationsszenarien, mit denen energieeffiziente Algorithmen für Sensornetzwerke evaluiert werden. Kapitel 5 beschäftigt sich mit existenten Algorithmen sowie einem im Zuge dieser Arbeit entwickelten Algorithmus zur Lokalisierung von Sensorknoten, welche eine Grundvoraussetzung für viele Clusteringalgorithmen darstellt.

Abschnitt III widmet sich den entwickelten Clusteringverfahren. Dabei befasst sich Kapitel 6 mit dem positionsbasierten Clustering und stellt die in diesem Zusammenhang entwickelten Algorithmen vor. Kapitel 7 befasst sich mit dem lokalisierungsfreien Clustering.

Abschnitt IV schließt die Arbeit mit einer Zusammenfassung in Kapitel 8. Diese Zusammenfassung sowie ein Ausblick über mögliche zukünftige Forschungsarbeiten in dem Themenbereich runden die Arbeit ab.



Abbildung 1.1 Struktur der Arbeit; Rot hervorgehoben sind Abschnitte, die wesentliche eigene Beiträge enthalten. Gelb hinterlegt sind Algorithmen, welche im Zuge der Arbeit entwickelt und analysiert wurden.

2 Grundlagen

Dieses Kapitel erläutert die technischen Grundlagen, die zum Verständnis der vorliegenden Arbeit notwendig sind. Kapitel 2.1 diskutiert zunächst die Grundlagen der Funkübertragung, welche in Sensornetzwerken als Kommunikationsschnittstelle verwendet wird. In Kapitel 2.2 wird der Begriff der drahtlosen Sensornetzwerke definiert. Das Kapitel nennt wichtige Anwendungsgebiete und betrachtet den Energieverbrauch von typischen Sensorknoten. In Kapitel 2.3 wird erläutert, welche Herausforderungen bei dem Einsatz drahtloser Sensornetzwerke zu erwarten sind. Abschließend geht Kapitel 2.4 auf den Stand der Technik von effizienten Clusteringalgorithmen ein.

2.1 Drahtlose Kommunikation

Die drahtlose Kommunikation mittels elektromagnetischer Wellen wurde seit deren Entdeckung Ende des 19. Jahrhunderts möglich [15]. Ein Hilfsmittel der modernen drahtlosen Datenübertragung bietet das seit 1983 standardisierte *Open System Interconnection Reference Model* (kurz: OSI-Modell), welches für die Datenübertragung entwickelt wurde [16]. Das OSI-Modell beschreibt die Datenübertragung als einen Prozess, der verschiedene Schichten durchläuft, wie er in Abbildung 2.1 dargestellt ist. Wenn ein Computer mit einem anderen Computer kommuniziert, werden beim Sender die verwendeten Protokollebenen von oben nach unten durchlaufen, bis die Nachricht schließlich übertragen wird. Der Empfänger der Nachricht arbeitet den Protokollstapel hingegen von unten nach oben ab, um die empfangenen Daten zu interpretieren und sinnvoll zu nutzen. Die drahtgebundene unterscheidet sich von der drahtlosen Kommunikation hauptsächlich durch das Übertragungsmedium und die unteren drei Schichten, die für das Verständnis der vorliegenden Arbeit nachfolgend im Einzelnen erläutert werden.

2.1.1 Übertragungsmedium

Werden bei der drahtgebundenen Kommunikation Kabel aus leitenden Materialien wie Kupfer oder Aluminium verwendet, dient die Luft bei der drahtlosen Kommunikation als Ausbreitungsmedium, bei der Nachrichten in Form elektromagnetischer Wellen übertragen werden.

Während bei der drahtgebundenen Kommunikation die Länge des Kabels gleichzeitig die maximale Reichweite bestimmt, hängt bei der Funkkommunikation dagegen die erzielte Reichweite davon ab, welche Leistung am Empfänger noch erkannt und vom allgegenwärtigen Rauschen unterschieden



Abbildung 2.1 OSI-Modell für die Rechnerkommunikation

werden kann. Die Empfangsleistung P_R hängt dabei von der Sendeleistung P_T , der Ausbreitungsgeschwindigkeit der Lichtgeschwindigkeit im Medium c, dem Abstand von Sender und Empfänger d, der Frequenz f und den Antennengewinnen der Sende- und der Empfangsantenne G_T und G_R ab. Für eine ungestörte Übertragung ist die Empfangsleistung mit der Friis'schen Gleichung (2.1) bestimmbar.

$$P_{R} = P_{T} * \left(\frac{c}{4*\pi * d*f}\right)^{2} * G_{T} * G_{R}$$
(2.1)

Statt mit Leistungsgrößen zu rechnen, wird bei der drahtlosen Kommunikation häufig mit Pegeln L in der Einheit Dezibel gearbeitet, bei der als Bezugsgröße 1 mW gewählt wurde. Die Friis'sche Gleichung stellt sich dann wie in Gleichung (2.2) beschrieben dar.

$$L_{R}[dBm] = L_{T}[dBm] + 20 \log_{10}\left(\frac{c}{1m} * \frac{1}{4\pi f}\right) + 20 \log_{10}(G_{T}G_{R}) - 20 \log_{10}\left(\frac{d}{1m}\right) \quad (2.2) \text{ (a)}$$

$$L_R = 10 \log_{10}(P_R/1mW); L_T = 10 \log(P_R/1mW)$$
 (2.2) (b)

Wird für einen Transceiver die maximale Frequenz f, der Pegel der Abstrahlleistung L_T und eine Empfängerempfindlichkeit E_R angegeben, ist die theoretische Reichweite ermittelbar, indem E_R mit dem Empfangspegel L_R verglichen wird. Solange L_R größer als E_R ist, ist theoretisch der Empfang einer Nachricht möglich.

Da eine reale Übertragung in der Regel nicht ungestört abläuft, sind die in TABELLE 2.1 aufgezählten Effekte in die Berechnung des empfangenen

EFFEKT	AUSWIRKUNG
Abschattung	Direkte Ausbreitung durch Hindernis verhindert
Beugung	An Kanten werden Wellen gebeugt
Brechung	Ablenkung der Welle bei Durchlauf durch Materialien mit unter- schiedlichen Brechungsindizes
Dämpfung	Schwächung des Signals
Doppler	Frequenzverschiebung durch Relativ-Bewegung von Sender und Empfänger
Reflexion	Spiegelung der Wellen an glatten Oberflächen
Streuung	Diffuse Reflexion an zur Wellenlänge rauen Oberflächen
Polarisation	Schwingungsrichtung der Wellen verändert sich an einer Grenz- fläche

TABELLE 2.1 EFFEKTE IM FUNKKANAL (NACH [2])

Leistungspegels mit einzubeziehen. Dies geschieht mit Hilfe von Kanalmodellen. Im Folgenden werden die beiden in der vorliegenden Arbeit verwendeten Modelle *Unit disk graph* und *Log-Normal Shadow Fading* beschrieben.

Unit disk graph Modell

Als einfachstes zur Verfügung stehendes Kanalmodell wird mittels *Unit disk graph* eine Sendereichweite R_T definiert. Um zu überprüfen, ob ein Empfänger *E* Nachrichten eines Senders *S* empfangen kann, wird der Abstand $d_{E,S}$ zwischen *S* und *E*, mit der Sendereichweite verglichen. Ein möglicher Empfänger *E*, dessen Abstand kleiner R_T ist, ist in der Lage, Nachrichten von *S* zu empfangen, während ein Empfänger mit einem größeren Abstand als R_T keine Nachrichten von *S* empfangen kann. Weiterhin wird beim *Unit disk graph* Modell keinerlei Dämpfung berücksichtigt. Die Empfangsleistung P_R ergibt sich damit für das *Unit disk graph* Modell aus der Sendeleistung P_T gemäß Gleichung (2.3).

$$P_R = \begin{cases} P_T, & d_{E,S} \le R_T \\ 0, & d_{E,S} > R_T \end{cases}$$
(2.3)

Das *Unit disk graph* Modell kann verwendet werden, um Algorithmen ähnlicher Funktionsweise ohne großen Aufwand miteinander zu vergleichen. Für Vergleiche von Algorithmen unterschiedlicher Funktionsweise, beispielsweise positionsbasiertes Clustering verglichen mit reichweitenbasiertem Clustering, ist dieses Modell dagegen aufgrund der vereinfachten Annahmen nicht geeignet [17].

Log-Normal Shadow Fading Modell

Als ausreichend genaues Modell für den Leistungsverlust bei der drahtlosen Kommunikation hat sich das Log-Normal Shadow Fading Modell erwiesen.

Dieses Modell ist an die Friis'sche Freiraumgleichung angelehnt und beschreibt den zu erwartenden Signalstärkepegel am Empfänger wie in Gleichung (2.4) dargestellt [10, 18].

$$L_{R}[dBm] = L_{T}[dBm] + 20 \log_{10}\left(\frac{c}{1m * 4\pi f}\right) + 20 \log_{10}(G_{T}G_{R})$$

$$-n * 10 \log_{10}\left(\frac{d}{1m}\right) + X_{\sigma}$$
(2.4)

Im Gegensatz zur Freiraumdämpfung enthält das *Log-Normal Shadow Fading* Modell den Pfadverlustexpontenen *n* zur Darstellung einer gegebenenfalls höheren Dämpfung und eine normalverteilte Zufallsvariable X_{σ} mit dem Erwartungswert $\mu=0$ und einer Standardabweichung σ zur Beschreibung von Abschattungseffekten. Sowohl *n* als auch σ müssen für jede spezielle Umgebung experimentell ermittelt werden. Beispiele für experimentell ermittelte und für drahtlose Sensornetzwerke relevante Pfadverluste und Standardabweichungen werden in [10] beschrieben.

Neben den Modellen *Unit disk graph* und *Log-Normal Shadow Fading* gibt es weitere Kanalmodelle wie beispielsweise in [19, 20] beschrieben. Diese verwenden zusätzliche Informationen wie die Antennenhöhe oder von der Distanz abhängige Pfadverlustexponenten, um noch realistischer Dämpfungseffekte zu beschreiben. Jedoch benötigen aussagekräftigere Modelle für eine bessere Vorhersage des Signalverhaltens auch konkretere Aussagen über die Umgebung der verwendeten Sensorknoten, welche ohne genaue Kenntnisse über das Einsatzgebiet des Sensornetzwerks nicht möglich sind.

2.1.2 Bitübertragungsschicht

Die Bitübertragungsschicht regelt, wie einzelne Bits über das verwendete Medium übertragen werden. Da bei der drahtgebunden Datenübertragung meist Kupferkabel als Übertragungsmedium dienen, ist es möglich, digitale Zeichen durch veränderte Spannungspegel oder Spannungswechsel zu erkennen.

Um digitale Zeichen während der Funkübertragung zu übertragen, werden verschiedene Modulationstechniken verwendet. So wird beispielsweise bei Bluetooth [2] das Verfahren der Frequenztastung angewendet, bei dem unterschiedliche Frequenzen verschiedene Sendesymbole darstellen. Weitere Verfahren sind die digitale Phasentastung sowie die Amplitudentastung [21].

2.1.3 Sicherungsschicht

Die Sicherungsschicht hat die Aufgabe, eine fehlerfreie Übertragung von Daten zwischen Sender und Empfänger zu garantieren. Eine der Hauptaufgaben der Sicherungsschicht ist die Verwaltung des Zugriffs auf das Übertragungsmedium (engl. Medium Access Control, kurz: MAC). Da mehrere Kommunikationsteilnehmer gleichzeitig auf ein Medium zugreifen könnten, verwendet die Sicherungsschicht Zugriffsprotokolle, die Kollisionen durch das gleichzeitige Senden von Nachrichten verhindern. Diese Protokolle werden auch als Mehrfachzugriffsverfahren bezeichnet, die im Folgenden erläutert werden.

Mehrfachzugriff durch Raummultiplexing (engl. Space Division Multiple Access, kurz: SDMA)

Sofern die Ausdehnung in einem Netzwerk groß genug ist, können mehrere Sender an mehrere Empfänger ungestört gleichzeitig senden, solange gewährleistet ist, dass die Empfänger nur für sie bestimmte Nachrichten hören können. Ausgenutzt wird diese Art des Mehrfachzugriffs beispielsweise im zellularem Mobilfunk und im Satellitenfunk.

Mehrfachzugriff durch Frequenzmultiplexing (engl. Frequency Division Multiple Access, kurz: FDMA)

Bei diesem Zugriffsverfahren können die Sender auf verschiedenen Frequenzen senden, um Nachrichten zu übermitteln. Dieses ist über eine Vergabe von Frequenzen oder Frequenzsequenzen für verschiedene Sender möglich. Verwendung findet dieses Verfahren beispielsweise im klassischen UKW-Radio.

Mehrfachzugriff durch Zeitmultiplexing (engl. Time Division Multiple Access, kurz: TDMA)

Beim Zeitmultiplexing erhält jeder Sender einen Zeitschlitz, der für ihn reserviert ist. So ist gewährleistet, dass das Signal störungsfrei übertragen wird. Dabei wird zwischen synchronem und asynchronem Zeitmultiplexing unterschieden. Bei synchronem Zeitmultiplexing wird jedem Sender ein fester Zeitschlitz zugeordnet, während beim asynchronen Zeitmultiplexing die Sendezeiten bei Bedarf ermittelt werden können.

Mehrfachzugriff durch Codemultiplexing (engl. Code Division Multiple Access, kurz: CDMA)

Dieses Zugriffsverfahren zeichnet sich dadurch aus, dass ein Sendesignal auf mehreren Frequenzen parallel gesendet wird. Die verwendeten Frequenzen sind für jedes Signal individuell in einem Code festgelegt. Werden vom Empfänger die empfangenen Signale mittels dieses Codes gefiltert, wird das uncodierte Signal zurückgewonnen.

Neben den einzelnen Mehrfachzugriffsverfahren ist es zudem möglich, mehrere Verfahren miteinander zu kombinieren. Im UKW-Radio wird beispielsweise Frequenzmultiplexing verwendet, um einzelne Sender zu unterscheiden. Durch Raummultiplexing ist es außerdem möglich, Frequenzbänder in bestimmten räumlichen Abständen mehrfach zu vergeben. Ein anderes Beispiel ist das Frequency Hopping bei Bluetooth [2], bei dem gleichzeitig ein Frequenz- und Zeitmultiplexing stattfindet. Dieses dient hier zusätzlich dazu, die Störanfälligkeit durch frequenzselektive Störungen zu verringern.

2.1.4 Vermittlungsschicht

Aufgabe der Vermittlungsschicht ist es, Wege zwischen zwei Kommunikationspartnern zu finden und bereit zu stellen. Im Vergleich zur drahtgebundenen Kommunikation herrscht zumeist eine höhere Dynamik der möglichen Kommunikationspartner und somit der Routen. Diese wird zum einen durch Fading, zum anderen durch eine mögliche Beweglichkeit der Netzwerkteilnehmer verursacht.

2.1.5 Höhere Schichten

Für die höheren Schichten des OSI-Modells unterscheidet es sich grundsätzlich nicht, ob es sich um ein drahtloses oder ein drahtgebundenes Netzwerk handelt. Die Aufgaben dieser Schichten sind daher nicht Bestandteil dieser Arbeit und können in [16] nachgelesen werden.

2.2 Grundlagen drahtloser Sensornetzwerke

Nachdem in Kapitel 2.1 die für die drahtlose Kommunikation notwendigen Grundlagen diskutiert wurden, widmet sich Kapitel 2.2 den drahtlosen Sensornetzwerken, beginnend mit dem Aufbau eines solchen sowie dem Aufbau eines einzelnen Sensorknotens, im Folgenden auch als Knoten bezeichnet. Anhand von Beispielknoten und aktueller Sensortechnik wird dem Leser ein Überblick über die Leistungsaufnahme der einzelnen Komponenten eines Sensorknotens gegeben. Weiterhin werden Anwendungsgebiete für Sensornetzwerke erläutert.

2.2.1 Aufbau drahtloser Sensornetzwerke

Drahtlose Sensornetzwerke bestehen im Allgemeinen aus einer Anzahl von Sensorknoten und mindestens einer datenverarbeitenden Senke. Während die Senke ein gewöhnlicher PC sein kann, sind Sensorknoten speziell entwickelte Geräte, die trotz geringer Größe in der Lage sein müssen, die verschiedenen elektronischen Komponenten in sich zu vereinen. Abbildung 2.2 zeigt prototypische Exemplare von Sensorknoten. Je nach gewünschter Funktionalität ist man derzeit in der Lage, Sensorknoten in der Größe von Münzen herzustellen. Der schematische Aufbau eines Sensorknotens ist in Abbildung 2.3 zu sehen. Ein typischer Sensorknoten beinhaltet immer die folgenden Komponenten:



Abbildung 2.2 Beispiele und Größenvergleich aktueller Sensorknoten, (A) Java Sunspot, (B) Mica2Dot, (C) Tmote Sky

Microcontroller, Batterie, Sensoreinheit und Transceiver, deren Aufgaben im Folgenden kurz erläutert werden.

Der Microcontroller eines Sensorknotens übernimmt die Steuerung des Knotens. Er entscheidet, wann und wie oft gemessen und gesendet wird und ist in der Lage, Transceiver und Sensoreinheit von der Batterie zu trennen, um so Energie zu sparen.

Mittels der Transceivereinheit kommunizieren Sensorknoten mit benachbarten Knoten und nur durch diese Funkkommunikationsschnittstelle ist der Aufbau eines Netzwerks überhaupt erst möglich.

Die Sensoreinheit besteht meist aus einem A/D-Wandler und einem oder mehreren angeschlossenen Sensoren. Da die Sensoren auf jedem einzelnen Knoten installiert sind, ermöglicht ein Sensornetzwerk eine flächendeckende Überwachung einer bestimmten Messgröße.

Die Batterie nimmt bei heutigen Sensorknoten häufig den größten Platz ein. Da Sensorknoten aufgrund ihrer geringen Größe und ihrer Platzierung oft nicht in der Lage sind, genügend Energie aus der Umgebung zu gewinnen, wird unter Umständen auf eine wiederaufladbare Energiequelle verzichtet und ein Knoten wird unweigerlich ausfallen, sobald die Energie der Batterie erschöpft ist.



Abbildung 2.3 Schematischer Aufbau eines Sensorknotens

Ein komplettes Sensornetzwerk besteht nicht zwangsläufig aus einer Senke und einer Anzahl identischer Knoten, obwohl die Verwendung einheitlicher Knoten die Wartung deutlich vereinfacht. Allerdings entstehen selbst bei Sensornetzwerken aus identischen Knoten durch verschiedene Entfernungen zur Senke, Schwankungen in der Batteriekapazität und durch Umgebungseinflüsse Inhomogenitäten in der Sensorfähigkeit, der Aktivität, der Sendereichweite und schließlich der Lebenszeit der einzelnen Knoten.

2.2.2 Platzierung der Sensorknoten

Ähnlich dem genauen Aufbau eines Sensornetzwerks ist auch die Anordnung der einzelnen Knoten im Sensornetzwerk nicht vorgegeben. So ist es mittels koordinierter Platzierung möglich, die Knoten gezielt an Orten hohen Interesses zu platzieren oder die Knoten gleichmäßig zu verteilen, um eine möglichst großflächige Sensorabdeckung und Konnektivität zu erreichen. Im Gegensatz dazu können die Knoten auch zufällig verteilt werden. Dieses kann geschehen um entstehende Kosten bei der Platzierung zu sparen oder aber die Platzierung in unzugänglichen Gebieten überhaupt erst zu ermöglichen, beispielsweise durch den Abwurf der Knoten aus einem Flugzeug.

2.2.3 Energieaufnahme

Da es Ziel der vorliegenden Arbeit ist, die verfügbare Energie in Sensornetzwerken effizient auszunutzen, ist es notwendig zu ermitteln, an welcher Stelle im Netzwerk Energie verbraucht wird, also welchen Anteil die einzelnen Komponenten am Energieverbrauch eines Knotens haben. Da sich die vorliegende Arbeit nicht auf die Energieaufnahme eines speziellen Sensorknotens stützt, wird nur der Trend des Energieverbrauchs aufgezeigt, statt im Detail auf die Leistungsaufnahme einzelner Bauteile, also spezieller Sende- und Empfangsschaltungen, Filter oder die Leistungsfähigkeit der verwendeten Microcontroller einzugehen.

EIGENSCHAFT		Sensorknoten	I		
	CC1010	MICA2	Tmote Sky		
Versorgungsspannung	3 V	3 V	3 V		
Sendefrequenz	868 MHz	915 MHz	2400 MHz		
Transceiver					
Bitrate Br	76800 Bit/s	38400 Bit/s	250000 Bit/s		
Datenempfang	35,7 mW	21 mW	59,1 mW		
Sendung mit $L_t = -15 dBm$	72,6 mW	27,9 mW	29,7 mW		
Sendung mit $L_t = 0 dBm$	96,3 mW	50,4 mW	52,2 mW		
Schlafend	0.0006 mW	0,003 mW	0,006 mW		
Microcontroller					
Aktiv	44,4 mW	36 mW	5,4 mW		
Schlafend	0,03 mW	0,03 mW	0,015 mW		

TABELLE 2.2 EIGENSCHAFTEN AUSGEWÄHLTER SENSORKNOTEN*

* Werte aus den Datenblättern der entsprechenden Knoten abgeleitet [3], [7], [9]

Die Leistungsaufnahme sowie weitere Eigenschaften des Transceivers und des Microcontrollers verschiedener beispielhafter Knoten, dem *CC1010* [3], dem *MICA2* [7] und dem *Tmote Sky* [9], sind daher in TABELLE 2.2 dargestellt. Anhand der TABELLE 2.2 ist zu erkennen, dass alle vorgestellten Knoten einen ähnlichen Energieverbrauch aufweisen. Die Energieaufnahme der einzelnen Komponenten wird im Folgenden genauer erläutert.

Transceiver

Die Transceivereinheit aktueller Knoten kann in drei verschiedenen Modi betrieben werden: Empfangs-, Sende- oder Schlafmodus. Während sich Empfangs- und Sendeleistung in der Größenordnung kaum unterscheiden, nimmt ein Transceiver im Schlafmodus rund 10⁵ mal weniger Leistung auf, wird aber auf eingehende Signale nicht reagieren. Um Energie zu sparen, wird ein Knoten also so häufig wie möglich in den Schlafmodus wechseln und unnötige Kommunikation vermeiden.



Abbildung 2.4 Energie/Bit im Vergleich zur eingestellten Sendeleistung am Transmitter nach Formel (2.5) und den Datenblättern [3], [7], [9]

Aus TABELLE 2.2 geht hervor, dass ein Knoten im Sendemodus mehr Energie verbraucht, wenn seine abgestrahlte Leistung L_t steigt, während im Empfangsmodus die aufgenommene Leistung als konstant angesehen wird.

Um die Transceivereinheiten der einzelnen Knoten zu bewerten, genügt es nicht, die Leistungsaufnahme der Transmitter für bestimmte Reichweiten zu vergleichen. Zum einen muss während der Datenübertragung ein empfangender Knoten im Datenempfangsmodus sein, verbraucht demzufolge ebenfalls Energie. Zum anderen ist es entscheidend, welche Datenrate von einem entsprechenden Transceiver erreicht wird, da dieses ausschlaggebend für die benötigte Zeit ist, in welcher Transmitter und Receiver zweier kommunizierender Knoten aktiv sind. Um eine Vergleichsmetrik zu erhalten, wurde in der vorliegenden Arbeit die aufgenommene Energie je übertragenes Bit E_{Bit} für eine Punkt zu Punkt Verbindung und eine bestimmte Sendereichweite anhand Gleichung (2.5) berechnet.

$$E_{Bit}[J/Bit] = 1/Br * (P_{trans} + P_{rec})$$
(2.5)

Br entspricht dabei der maximal möglichen Bitrate, P_{trans} der Leistungsaufnahme des Transmitters bei entsprechend eingestellter Sendeleistung und Prec entspricht der Leistungsaufnahme am Receiver. Es ist zu beachten, dass die gewählte Metrik nur Best-Case Informationen über den Energieverbrauch liewährend fert, da auf weitere energieverbrauchende Prozesse der Kommunikation nicht eingegangen wird. So wird weder das Abhören des Kanals, noch die Neuübertragung im Fehlerfall, noch der Energieverbrauch während des An- und Abschalten des Transceivers berücksichtigt. Dennoch ist es möglich, mit dieser Metrik die Knoten der TABELLE 2.1 untereinander zu

vergleichen, wie Abbildung 2.4 zeigt. Die Grafik zeigt deutlich, dass der *Tmote Sky* aufgrund seiner hohen Sendefrequenz und damit erzielten hohen Bitrate deutlich weniger Energie verbraucht als die anderen beiden Sensorknoten. Gleichzeitig ist die maximale Sendeleistung geringer als die der anderen Sensorknoten. Interessanterweise steigt bei allen Knoten E_{Bit} unverhältnismäßig gering zur Sendeleistung an, was zum einen am einbezogenen Energieverbrauch des Empfängers liegt, zum anderen jedoch an den jeweiligen Transmittern, welche auch für eine geringe Abstrahlleistung eine relativ hohe Leistung benötigen.

Von der Energie je Bit darf jedoch nicht fälschlicherweise darauf geschlossen werden, dass der Tmote Sky für eine abgestrahlte Sendeleistung von -20 dBm bis 5 dBm den anderen Sensorknoten überlegen ist. Dieses zeigt die Ermittlung der theoretisch maximalen Sendereichweite mit der Früs'schen Gleichung (2.2). Werden den Knoten entsprechende Empfänger-empfindlichkeiten und Sendefrequenzen eingesetzt, erhält man die theoretische maximale Sendereichweite, wie in Abbildung 2.5 zu sehen ist. Während der Tmote Sky nur eine theoretische Reichweite von 500m erzielt, erreichen die anderen Sensorknoten circa zehnfach höhere Reichweiten. In Abbildung 2.6 ist die Energie je Bit im Verhältnis zur theoretisch möglichen Sendereichweite dargestellt. Man erkennt, dass ein Tmote Sky deutlich günstiger Bits überträgt. Werden jedoch hohe Entfernungen benötigt, muss bei einer Übertragung mittels Tmote Sky auf Multihop zurückgegriffen werden, was bei idealem Abstand der Knoten der Multihop-Strecke mindestens zu einem linearen Anstieg von E_{Bit} zur Sendereichweite führt, und durch die Trendlinie dargestellt ist. Ab einer theoretischen Sendereichweite von ca. 7300m wird der Tmote Sky in jedem Fall ineffizienter als die beiden Konkurrenzknoten. Der auf hoher Frequenz sendende *Tmote Sky* ist dementsprechend nur energieeffizienter, solange das



Abbildung 2.5 Theoretisch erreichbare Sendereichweite im Vergleich zur eingestellten Sendeleistung am Transmitter für $G_T = 1$, $G_R = 1$



Abbildung 2.6 Benötigte Energie je Bit im Vergleicht zu theoretisch erreichbaren Sendereichweite für $G_T = 1$, $G_R = 1$

Netzwerk über eine geringe räumliche Ausdehnung verfügt.

Microcontroller

Da es sich bei Microcontrollern im Gegensatz zum Transceiver um digitale CMOS (Complementary Metal Oxide Semiconductor) Schaltungen handelt, sind die für CMOS geltenden Gesetze für die Leistungsaufnahme auch für Sensorknoten anwendbar. Wie in Gleichung (2.6) beschrieben, setzt sich die Verlustleistung demnach aus der dynamischen Verlustleistung P_{dyn} , der statischen Verlustleistung P_{stat} und der Verlustleistung durch Kurzschlussströme P_{SC} zusammen.

$$P_{CMOS} = P_{dyn} + P_{stat} + P_{sc} \tag{2.6}$$

Dabei ist die dynamische Verlustleistung P_{dyn} ferner abhängig von der Versorgungsspannung U_{DD} , der Lastkapazität der aufzuladenden Lastkapazitäten C_L , der Schaltwahrscheinlichkeit $\alpha_{0\to 1}$ und der Taktfrequenz f, wie in Gleichung (2.7) beschrieben.

$$P_{dyn} = U_{DD}^2 * C_L * f * \alpha_{0 \to 1}$$
(2.7)

Die zweite Komponente der Verlustleistung ist die statische Verlustleistung P_{stat} , die durch verschiedene Arten von Leckströmen entsteht. Diese treten zum einen an gesperrten PN-Übergängen auf (I_{PN}) , zum zweiten können Elektronen durch die Gates von Transistoren tunneln (I_G) und zum dritten gibt es den Anteil der sogenannten *Sub-Threshold* – Ströme von Drain zu Source (I_{ST}) . Auch hier hängt der Anteil der Verlustleistung von der anliegenden Spannung ab, wie in Gleichung (2.8) dargestellt ist.
$$P_{stat} = U_{DD} * (I_{PN} + I_G + I_{ST})$$
(2.8)

Die dritte Komponente P_{SC} der Verlustleistung entsteht durch die Kurzschlussströme, die während des Umschaltvorgangs fließen. Diese Komponente hat in aktuellen Schaltungen nur einen geringen Anteil auf die Verlustleistung.

Um die Verlustleistung einer integrierten Schaltung zu senken, verfügen moderne Microcontroller über einen Schlafmodus, bei dessen Ausführung weite Teile des Microcontrollers abgeschaltet werden. Reaktiviert wird der Microcontroller dann durch einen integrierten Timer oder ein externes Signal. Der Schlafmodus ermöglicht für derzeitige Schaltungen eine Energieersparnis um den Faktor 100 bis Faktor 1000, wie bereits in TABELLE 2.2 dargestellt wurde.

Sensoren

Die Energieaufnahme der Sensoreinheit eines Sensorknotens hängt vor allem von dem Einsatzszenario und damit von den verwendeten Sensoren ab. TABELLE 2.3 zeigt exemplarisch verschiedene Sensortypen. So gibt es äußerst energiearme Beschleunigungs- oder Drucksensoren. Deren Leistungsaufnahme ist im Vergleich zum Transceiver und Microcontroller vernachlässigbar gering und die verbrauchte Leistung wird hauptsächlich durch Verstärkerschaltungen verursacht. Im Gegensatz dazu stehen selektive Gassensoren, deren Leistungsaufnahme während einer aktiven Messung die Leistungsaufnahme des gesamten Sensorknotens dominiert. In dem in TABELLE 2.3 gewählten Beispiel eines Gassensors wird der hohe Leistungsverbrauch durch die Verwendung von Leuchtdioden verursacht. Andere Schaltungen, die für die Detektion von Gasen Heizelemente verwenden, verbrauchen zum Teil noch deutlich mehr Leistung [22].

Genau wie bei den anderen Komponenten eines Sensorknotens kann die Sensoreinheit abgeschaltet werden, um den Energieverbrauch zu senken. Zu beachten ist, dass Sensoren mit Heizelementen relativ viel Zeit im Bereich von

EIGENSCHAFT	SENSOR						
	BMP085 [1]	BMA150 [6]	SHT21 [8]	SensorChip TM CO ₂ Sensor [11]			
Sensorfähigkeit	Druck	Beschleu- nigung	Luftfeuchtigkeit und Temperatur	Kohlendioxid			
Aktive Messung	7,5 μW	600 µW	900 μW	70000 μW			
Schlafmodus	0,25 μW	3 µW	0,5 µW	keine Angabe			

TABELLE 2.3 EIGENSCHAFTEN AUSGEWÄHLTER SENSOREN^{*}

* Werte aus den Datenblättern der entsprechenden Sensoren abgeleitet

Minuten bis Stunden benötigen, um ihre Betriebstemperatur zu erreichen. Sie sind aufgrund dieser Anlaufzeit und ihres hohen Energiebedarfs daher für ressourcenbegrenzte Sensorknoten eher ungeeignet.

2.2.4 Anwendungsgebiete

Wie im vorhergehenden Abschnitt beschrieben, besteht ein Sensornetzwerk aus mindestens einer Senke und zahlreichen batteriebetriebenen Knoten. Da die minimale Größe der Knoten derzeit bis in den Bereich von Kubikzentimetern reicht, sind die Einsatzmöglichkeiten mannigfaltig. Im Prinzip kann jede sensorische Aufgabe wahrgenommen werden, welche die Ergebnisse eines Sensorfelds erfordert. Die folgende Auflistung verschiedener Anwendungsgebiete stellt daher nur einen Teil der möglichen Anwendungen dar, veranschaulicht jedoch die Bedingungen, die in einem Sensornetzwerk herrschen.

Umweltüberwachung

Als Prototyp einer nicht militärischen Sensornetzwerkanwendung gilt die Überwachung des Verhaltens des Lebensraums von Lebewesen. Auf Great Duck Island wurde bereits 2003 ein Sensornetzwerk installiert, welches die Brutplätze und die Umgebung der dort heimischen Vögel überwacht [13]. Eine weitere Anwendungsmöglichkeit ist die Überwachung von Agrarflächen. So wird mit Hilfe von Sensornetzwerken der Dünger- und Wassergehalt im Boden gemessen und überwacht [23]. Mittels gezielter Bewässerung oder Düngung ist es demzufolge möglich, die Erträge zu steigern und die Umwelt zu schonen.

Identifizierung von Naturkatastrophen

Besonderes Einsatzpotential für Sensornetzwerke bietet sich im Bereich des Katastrophenschutzes, da dort häufig große Gebiete dauerhaft überwacht werden müssen, um eine Naturkatastrophe rechtzeitig zu erkennen. So sind Sensornetzwerke beispielsweise zur Überwachung von Hochwassern geeignet. Hier ist es zum einen möglich, durch eine verteilte Messung von Wasserdruck, Regenmenge und Temperatur auf die Entstehung von Hochwassern zu schließen [24], zum anderen können in künstlichen Deichen platzierte Sensorknoten mittels Wasserstands- oder Feuchtigkeitsmessungen die Stabilität der Deiche erkennen und rechtzeitig vor eventuellen Deichbrüchen warnen [25].

Ein weiteres Aufgabenfeld im Katastrophenschutz ist die Überwachung von Wäldern und die frühzeitige Erkennung von Waldbränden mittels Temperatur-, Luftfeuchte-, Wind- und Rauchsensoren. Wie in [26] vorgestellt, bietet ein Sensornetzwerk im Gegensatz zu herkömmlichen Überwachungsmethoden eine kostengünstige und wetterunabhängige Alternative.

Auch zur Überwachung vulkanischer Aktivitäten gibt es bereits Forschungsarbeiten und prototypische Sensornetzwerkinstallationen. So wurden bereits mehrere Sensornetzwerke in Ecuador rund um den Vulkan Tungurahua installiert, um dort seismische Aktivitäten zu überwachen und aufzuzeichnen [14].

Identifizierung und Verfolgung mobiler Objekte

Ein weiteres Anwendungsfeld für drahtlose Sensornetzwerke ist die Verfolgung von Personen und mobilen Objekten. Im Bereich der medizinischen Assistenzsysteme ist es beispielsweise möglich, alte oder kranke Menschen mittels Sensornetzwerk zu überwachen und so das Pflegepersonal zu unterstützen [27]. Weiterhin ist im militärischen Bereich die Identifizierung und Verfolgung von feindlichen Fahrzeugen oder Soldaten mittels drahtloser Sensornetzwerke Thema aktueller Forschungen [28], da Sensorknoten aufgrund ihrer einfachen Reproduzierbarkeit und geringen Größe im Gegensatz zu Drohnen oder Satelliten kein lohnendes militärisches Ziel bieten und zudem relativ wetterunabhängig sind.

2.3 Herausforderungen drahtloser Sensornetzwerke

Nachdem in Kapitel 2.2 mögliche Anwendungsszenarien vorgestellt wurden, widmet sich Kapitel 2.3 den technischen Herausforderungen, die durch den Einsatz von Sensorknoten in solchen Szenarien entstehen. Zum Teil sind diese Herausforderungen bereits durch das OSI-Modell beschrieben. So müssen für einen effizienten Betrieb eines Sensornetzwerks geeignete Medienzugriffs- und Routingverfahren gewählt werden. Zusätzlich existieren sensornetzwerk-spezifische Herausforderungen. Dazu zählen eine für Sensornetzwerke geeignete Organisationsstruktur, die Lokalisierung der einzelnen Knoten und detektierter Phänomene, sowie eine ausreichende Kommunikations- und Sensorabdeckung im Fall eines energieeffizienten Betriebs. Das vorliegende Kapitel erläutert die einzelnen Herausforderungen und verweist auf mögliche Lösungsansätze.

2.3.1 Organisation

Sofern in einem drahtlosen Sensornetzwerk, welches aus Hunderten von Knoten besteht, jeder Knoten einzeln nach seinen Sensordaten abgefragt wird, kommt es sowohl zu einer hohen Verzögerung als auch zu einer enormen Energieverschwendung. Knoten in der Nähe der Senke hätten ununterbrochen die energieaufwändige Aufgabe, Statusnachrichten für und von anderen Knoten weiterzuleiten. Da die Übertragungsrate begrenzt ist, sinkt zudem die Antwortzeit jedes einzelnen Knoten mit der Anzahl der Knoten im Netzwerk. Um die Knoten energieeffizient und ohne hohe Verzögerungen zu verwalten, ist es notwendig, die direkte Kontrolle jedes einzelnen Knotens von der Senke aus durch eine Art der Selbstorganisation zu ersetzen. Die Konzepte der Selbstorganisation werden in Kapitel 3 erläutert.

2.3.2 Lokalisierung

Innerhalb eines drahtlosen Sensornetzwerks ist es nicht nur entscheidend herauszufinden, dass und wann ein Phänomen aufgetreten ist, sondern auch, dessen Position und räumliche Ausdehnung zu kennen. Die Lokalisierung befasst sich mit dieser Problematik und wird ausführlich in Kapitel 5 diskutiert.

2.3.3 Medienzugriff

Der Medienzugriff zur Datenübertragung innerhalb eines Sensornetzwerks unterliegt den gleichen Herausfroderungen wie in anderen drahtlosen Netzwerken. Entsprechende Protokolle sind dementsprechend mit Problematiken wie einer effizienten Kollisionsvermeidung oder dem Hidden Terminal Problem konfrontiert. Diese werden beispielsweise im WLAN (engl. Wireless Local Area Network) mittels RTS/CTS (engl. ready to send/clear to send) Nachrichten auf ein annehmbares Niveau reduziert [29]. Zusätzlich zu diesen Problematiken hat ein effizientes MAC-Protokoll die Bedürfnisse von Sensornetzwerken zu erfüllen. Demzufolge ist ein energieeffizientes Arbeiten unerlässlich, beispielsweise indem auf unnötiges Abhören des Mediums und unnötige Kommunikation verzichtet wird. Gleichzeitig ist es Aufgabe eines MAC-Protokolls für Sensornetzwerke, eine geringe Verzögerung einer Nachricht vom Knoten in Richtung Senke zu erzielen, um beispielsweise schnell einen Waldbrand detektieren zu können. Zusätzlich zu diesen Aufgaben stehen MAC-Protokolle für drahtlose Sensornetzwerke vor der Herausforderung, eine große Anzahl an Knoten zu verwalten, ohne die Datenübertragung zu beeinträchtigen.

Aufgrund dieser Herausforderungen sind MAC-Protokolle Gegenstand aktueller Forschungen. Dabei wird zwischen konkurrierenden und nicht-konkurrierenden Verfahren unterschieden. Nicht-konkurrierende Verfahren basieren dabei auf einer TDMA Variante [30]. Der Vorteil ist, dass der Kanal nur in bestimmten Zeiträumen abgehört werden muss, da andere Knoten nur zu diesen Zeiten senden dürfen. Nachteilig sind die notwendige Synchronisation und die durchschnittliche Verzögerung zwischen der Detektion eines Phänomens und der Benachrichtigung der Senke, welche sich mit steigender Anzahl an Kommunikationspartnern mit eigenem Zeitschlitz vergrößert.

Bei den konkurrierenden Verfahren, wie dem in [29] verwendete CSMA/CA (engl. *Carrier Sense Multiple Access/Collision Avoidance*), greifen mehrere Knoten gleichzeitig auf den Kanal zu. Durch vorheriges Abhören des Kanals werden dabei Kollisionen weitestgehend vermieden. Vorteil des konkurrierenden Verfahrens ist, dass eine Nachricht schnell durch das Netzwerk gelangen kann. Nachteilig ist, dass die Kollisionen mit steigender Anzahl an Kommunikationspartnern zunehmen und Knoten zu einer schnellen Weiterleitung von Daten permanent den Kanal abhören müssen.

Sowohl konkurrierende als auch nicht-konkurrierende Verfahren haben mit wachsender Anzahl möglicher Kommunikationspartner starke Performanceeinbußen. Kombinierte Verfahren sind in der Lage, diese Einbußen abzuschwächen und gleichzeitig auf die Erfordernisse von Sensornetzwerken einzugehen. Daher werden für drahtlose Sensornetzwerke meist speziell entwickelte MAC-Protokolle verwendet, die eine Kombination aus nicht-konkurrierenden und konkurrierenden Verfahren darstellen.

In S-MAC (engl. *Sensor-MAC*) [31] gehen beispielsweise alle Knoten periodisch in einen Schlafmodus. Benachbarte Knoten kennen ihre Schlafperiode und sind so in der Lage, Nachrichten entsprechend weiterzuleiten. Um Kollisionen zu vermeiden, wird dann das CSMA/CA Verfahren angewendet.

Die in [32] und [33] beschriebenen MAC-Protokolle gehen bereits von einem Routingbaum mit der Senke als Wurzel aus und optimieren den Kanalzugriff für eine schnelle und energieeffiziente Datenübertragung in Richtung Senke.

Weitere MAC-Protokolle sind in [34] zusammengefasst und stehen nicht im Fokus der vorliegenden Arbeit. Zusammenfassend ist festzustellen, dass ein MAC-Protokoll für ein Sensornetzwerk so auszuwählen ist, dass die Aufgabenstellung des Sensornetzwerks optimal unterstützt wird. Für Aufgaben wie Langzeit-Datenaggregation ist ein energieeffizientes MAC-Protokoll mit hoher Verzögerung daher eher geeignet, während sich für ein Sensornetzwerk zur Verfolgung mobiler Objekte MAC-Verfahren anbieten, die Nachrichten auf Kosten eines erhöhten Energieverbrauchs schnell weiterleiten und die Senke möglichst zeitnah informieren.

2.3.4 Routing

Genau wie das MAC-Protokoll steht das Routing in einem drahtlosen Sensornetzwerk vor den gleichen Herausforderungen der gewöhnlichen drahtlosen Kommunikation.

Effiziente Routingprotokolle für drahtlose Sensornetzwerke profitieren von den Eigenschaften der Sensorknoten. Diese sind zum einen relativ immobil, so dass sich in der Nachbarschaft, außer durch ausgefallene Knoten, wenig ändert. Zum anderen besteht in vielen Szenarien für drahtlose Sensornetzwerke hauptsächlich ein Bedarf an Routen von jedem einzelnen Sensorknoten zur Senke, während Routen zwischen den Sensorknoten oftmals vernachlässigt werden können. So bieten sich proaktive Protokolle an, die von der Senke aus einen Routingbaum aufbauen. Zum einen ermöglichen diese Routingbäume, dass der MAC Zugriff so optimiert wird, dass Nachrichten möglichst schnell bzw. energieeffizient die Senke erreichen. Die Routingbäume verhindern weiterhin, dass auf jedem Knoten aufwendige Routingtabellen gespeichert werden, wie es bei herkömmlichen Routingalgorithmen wie beispielsweise AODV-Routing (*engl. Ad-Hoc On-Demand Distance Vector-Routing*) [35] der Fall ist .

Zudem gehen Routingbäume auf die Erfordernisse von Sensorknoten ein. Stehen beispielsweise Knoten mit unterschiedlichem Restenergieniveau oder Cluster mit unterschiedlicher Mitgliederzahl zur Verfügung, ist der Routingbaum so zu optimieren, dass eine Energiebalance im gesamten Sensornetzwerk entsteht.

Neben bereits bestehenden Algorithmen [36, 37] wurde auch im Zuge der vorliegenden Arbeit ein Algorithmus entwickelt, der in der Lage ist, den Routingbaum während des Aufbaus an das Energieniveau der routenden Knoten anzupassen [38].

Um zu verhindern, dass ein Routingbaum zusammenbricht, sobald Knoten ausfallen, muss der Baum entweder periodisch oder nach jedem Knotenausfall von der Senke aus neu aufgebaut werden oder adaptiv angepasst werden [39].

Neben den proaktiven Routingbäumen stehen für das Routing in einem drahtlosen Sensornetzwerk reaktive oder kombinierte Routingverfahren zur Verfügung [40]. Diese stehen jedoch nicht im Fokus der Arbeit und werden daher hier nicht weiter betrachtet.

2.3.5 Kommunikationsabdeckung

Zahlreiche energieeffiziente Algorithmen für Sensornetzwerke nutzen die Möglichkeit aus, Knoten in einen Schlafmodus zu versetzen. Eine der Hauptherausforderungen dabei ist es, dennoch eine komplette Kommunikationsabdeckung zu gewährleisten. Es muss dementsprechend für ein funktionierendes Sensornetzwerk gewährleistet sein, dass eine Nachricht jedes Knotens die Senke erreichen kann. Ist ein Knoten in einem bestimmten Gebiet nicht in der Lage, die Senke zu erreichen, wird dieses Gebiet in der vorliegenden Arbeit als Kommunikationloch bezeichnet.

Die in Kapitel 6 und Kapitel 7 entwickelten Clusteringalgorithmen betrachten eine komplette Kommunikationsabdeckung ohne Kommunikationslöcher als Grundvoraussetzung für ein funktionierendes Clustering.

2.3.6 Sensorabdeckung

Neben der drahtlosen Kommunikation besitzen Sensorknoten die Fähigkeit, Phänomene mit Hilfe von Sensoren zu detektieren. Diese Grundfähigkeit eines Sensornetzwerks setzt voraus, dass genügend Knoten und deren Sensoren aktiv sind, um das ihnen zugewiesene Gebiet ausreichend zu überwachen. Können in einem Gebiet Phänomene nicht detektiert werden, so wird dieses Gebiet im Zuge dieser Arbeit als Sensorloch bezeichnet. Zusammen mit der Kommunikationsabdeckung gehört eine Sensorabdeckung zu den Restriktionen, die bei einer vorübergehenden Abschaltung von Knoten durch einen energieeffizienten Algorithmus zu beachten sind. Die in Kapitel 6 und Kapitel 7 entwickelten Clusteringalgorithmen betrachten die Sensorabdeckung ebenfalls als Grundvoraussetzung für ein funktionierendes Clustering.

2.4 Grundlagen des Clusterings

Nachdem in Kapitel 2.3 die Herausforderungen drahtloser Sensornetzwerke definiert wurden, geht das vorliegende Kapitel auf das Clustering ein. Clustering bezeichnet in drahtlosen Sensornetzwerken die allgemeine Strategie, die Sensorknoten eines Sensornetzwerk zu gruppieren, also in Cluster aufzuteilen [41].

Hauptziel des Clusterings ist zumeist ein energieeffizienter Betrieb des Sensornetzwerks mit dem Ziel, die Netzwerklebenszeit zu maximieren, ohne die Netzwerkfunktionalität einzuschränken. Das Clustering nutzt dazu verschiedene Methoden, um Energieeinsparungen zu ermöglichen. Die für diese Arbeit bedeutendsten Methoden werden nachfolgend erläutert.

Hierarchische Strukturen

Clustering wird häufig dazu verwendet, das Organisationsproblem in Sensornetzwerken zu lösen. Anstatt jedem einzelnen Knoten Aufgaben zuzuweisen, ist es möglich, mittels Clustering hierarische Strukturen zu erschaffen, sodass die Anzahl der Kontrollnachrichten sich auf Nachrichten zwischen der Senke und der nächsten Stufe in der Hierarchie beschränkt.

Datenaggregation

Benachbarte Knoten erzielen oftmals ähnliche oder identische Messergebnisse. Sind die Cluster so gewählt, dass benachbarte Knoten demselben Cluster angehören, kann eine Datenaggregation innerhalb eines Clusters bereits für eine extrem verringerte Datenlast sorgen.

Erkennung und temporäre Abschaltung redundanter Knoten

Eine der energieeffizientesten Maßnahmen im drahtlosen Sensornetzwerk ist es, die für den Netzwerkbetrieb redundanten Knoten ausfindig zu machen und temporär abzuschalten. Gerade bei Sensorknoten mit zufälliger Position entstehen meist zwingend Redundanzen, da hier im Normalfall deutlich mehr als die Mindestmenge notwendiger Knoten ausgebracht werden muss, um Gebiete ohne Kommunikations- oder Sensorabdeckung infolge einer ungünstigen Knotenverteilung zu vermeiden.

Ist es möglich, mittels geeignetem Clusteringalgorithmus die redundanten Knoten zu erkennen und abzuschalten, kann zum einen deren Energie zu einem späteren Zeitpunkt genutzt werden, zum anderen belasten sie das Übertragungsmedium nicht durch das Versenden von Nachrichten, so dass die umliegenden, aktiven Knoten ungestört senden können.

Heilungsmechanismen

Hat eine erfolgreiche Redundanzerkennung stattgefunden, kann ein geeignetes Clustering auch dabei helfen, geeignete Kandidaten für eine Reaktivierung zu finden, sollte einer der aktiven Knoten spontan oder erwartet ausfallen. Dieses erspart eine aufwändige Suche nach einem geeigneten Kandidaten, wodurch Sensorlöcher, also Gebiete ohne Sensorabdeckung im Sensornetzwerk, schnell geschlossen werden können.

Energiebalance

Cluster sind in der Lage, ihr eigenes Energieniveau anhand der verbleibenden Knoten des Clusters oder deren Energie zu bestimmen. Diese unterschiedlichen Niveaus sind nutzbar, indem Knoten starker Cluster beispielsweise als Routingknoten verwendet werden [38]. Die somit erzielte Energiebalance erlaubt häufig eine längere Lebensdauer des gesamten Netzwerks.

Es gibt verschiedene Arten, Knoten eines Sensornetzwerks zu Clustern zu gruppieren. Die im Zuge dieser Arbeit betrachteten Varianten sind das zeitliche sowie das geographische Clustering.

Eine Unterteilung der jeweiligen Verfahren wurde anhand des verfügbaren Wissens über die geographischen Positionen der Knoten im Netzwerk geschaffen. Ist einem Knoten seine Position im Netzwerk bewusst und wird diese Position zum Clustern verwendet, handelt es sich um einen positionsbasierten Algorithmus. Basiert ein Clusteringalgorithmus nicht auf den Positionen der Knoten, so handelt es sich um einen lokalisierungsfreien Algorithmus.

Kapitel 2.4.1 befasst sich mit den zeitlichen Clusteringalgorithmen und ihren Vertretern k-Cover [5] und CPNS (engl. *Coverage Preserving Node Scheduling*) [42]. Kapitel 2.4.2 befasst sich mit dem geographischen Clustering durch Algorithmen wie GAF (engl. *Geographical adaptive fidelity*) [43], LEACH (engl. *Low-Energy Adaptive Clustering Hierarchy*) [44] und HEED (engl. *Hybrid Energy-Efficient Distributed Clustering*) [45].

Weitere Arten des Clusterings, beispielsweise das Gruppieren von Knoten mit ähnlichen Sensoreigenschaften, stehen nicht im Fokus der vorliegenden Arbeit und werden daher nicht betrachtet.

2.4.1 Zeitliches Clustering

Als zeitliches Clustering wird eine Strategie bezeichnet, bei der die Knoten einem Cluster zugehörig sind, die während eines bestimmten Zeitraums eine bestimmte Aufgabe erfüllen. Die Knoten eines zeitlichen Clusters sind möglichst über das gesamte abzudeckende Gebiet verteilt. Somit kann bereits ein einzelner aktiver Cluster ausreichen, um komplette Funktionalität des Sensornetzwerks zu gewährleisten. Zeitlich clusternde Algorithmen können sowohl positionsbasiert als auch lokalisierungsfrei sein. Als Beispiel für ein lokalisierungsfreies zeitliches Clustering wird zunächst der k-Cover Algorithmus vorgestellt, bevor anschließend der CPNS Algorithmus als Beispiel für ein positionsbasiertes Clustering beschrieben wird.

k-Cover Algorithmus

Der k-Cover Algorithmus [5] gruppiert ein Sensornetzwerk in *k* Cluster, welche als Cover bezeichnet werden. Um zu ermöglichen, dass jeder Cover Knoten aus jedem Bereich des Sensornetzwerks enthält, wählt jeder Knoten zufällig aus, zu welchem Cover er gehört. Nach dieser Initialisierung ist im laufenden Betrieb des Sensornetzwerks zu jedem Zeitpunkt nur ein als Cover bezeichneter Cluster aktiv, während sich die anderen Cluster in einem Schlafmodus befinden. Nach einer bestimmten Zeit wechselt der aktive Cover in einen Schlafzustand, während der nachfolgende Cover aktiviert wird. Je mehr Cover dementsprechend gebildet werden, desto höher ist die Energieersparnis, da weniger Knoten gleichzeitig aktiv sind. Jedoch sinkt mit steigender Anzahl an Covern die Wahrscheinlichkeit, einen Netzwerkbetrieb ohne Sensor- oder Kommunikationslöcher aufrecht zu erhalten.

In [46] wird die zufällige Wahl des Covers für einen Knoten durch ein heuristisches Verfahren verbessert. Anstatt dass sich Knoten alle gleichzeitig für einen



Abbildung 2.7 Clustering eines Sensornetzwerks, (A) Ungeclustertes Netzwerk, (B) Zeitliches Clustering nach [5], (C) geographisches Clustering mit einfacher Rollenzuweisung, (D) Geographisches Clustering mit komplexer Rollenzuweisung

Cover entscheiden, geschieht dieses für jeden Knoten nach einer Zufallszeit. Nachdem ein Knoten *i* sich für einen der *k* möglichen Cover entschieden hat, sendet er einen Broadcast B_i an alle Knoten in der Umgebung, der die Information über den gewählten Cover k_i enthält. Ein Knoten *j*, der einen oder mehrere dieser Broadcasts *B* von Knoten in seiner Umgebung empfangen hat, wählt dann zufällig aus einem der Cover aus, welche am Geringsten in seiner Umgebung vertreten sind und sendet seinerseits einen Broadcast B_j aus. Durch diesen Mechanismus wählen sich Knoten, die bereits mehrere Broadcasts empfangen haben, mit einer höheren Wahrscheinlichkeit einen günstigen Cover, sodass sich jeder Cover somit vorteilhafter über das gesamte Netzwerk verteilt. Zur Veranschaulichung ist eine mögliche Einteilung eines Sensornetzwerks in 2 Cover in Abbildung 2.7 (B) dargestellt.

Coverage Preserving Node Scheduling

Um die Position der aktiven Knoten nicht dem Zufall zu überlassen, sondern gezielt die Knoten auszuschalten, die nicht benötigt werden, wurde in [42] der *Coverage Preserving Node Scheduling* (CPNS) Algorithmus entwickelt. Dieser sieht vor, nach der Platzierung zunächst alle Knoten zu aktivieren. Dabei hat jeder Knoten die Aufgabe, ein hier als seine Sensorfläche bezeichnetes Gebiet komplett abzudecken. Die Sensorfläche ist dabei die Kreisscheibe, die von der Sensorreichweite von *K* aufgespannt wird, wie in Abbildung 2.8 (A) dargestellt ist. Um herauszufinden, ob ein Knoten in einen Schlafmodus versetzt werden kann, sendet jeder aktive Knoten einen Broadcast mit seiner Position, bei welchem die Sendereichweite R_T mindestens der Sensorreichweite R_S entspricht. Ein Knoten *K*, der einen solchen Broadcast empfängt, kann mit Hilfe des Abstands, sowie seiner Position und der Position des sendenden Knotens ein als *Sponsored Coverage* bezeichnetes Gebiet errechnen, wie Abbildung 2.8 (B) zeigt. Dieses ist ein Kreissektor, welcher je nach Abstand des Nachbarn einen



Abbildung 2.8 Redundanzerkennung bei CPNS: (A) Sensorfläche und für die Redundanzerkennung relevante Nachbarn von K, (B) Knoten K empfängt Broadcast eines Nachbarn und ermittelt draus die Sponsored Coverage des Nachbarn, (C) Die Sponsored Coverages alle Nachbarn bedecken die gesamte Sensorfläche von K, K ist redundant.

Mittelpunktswinkel von $120^{\circ} \le \alpha < 180^{\circ}$ und einen Radius von $r = R_s$ besitzt. Nachdem jeder Nachbar einen Broadcast gesendet hat, berechnet Knoten *K* für jeden Nachbarn die entsprechende *Sponsored Coverage*. Der Knoten *K* gilt dann als redundant, wenn die *Sponsored Coverages* der Nachbarn von Knoten *K* die Sensorfläche komplett abdecken, wie Abbildung 2.8 (C) zeigt. Ist ein Knoten redundant, schaltet er sich temporär ab. Um zu vermeiden, dass sich benachbarte Knoten gleichzeitig abschalten, informiert ein redundanter Knoten *K* vorher seine Nachbarn ebenfalls mittels Broadcast mit Sendereichweite $R_T = R_S$, und die Nachbarn entfernen die Sponsored Coverage von Knoten *K* für ihre eigene Redundanzerkennung.

Nach bestimmten Zeiten findet auch hier ein Rollentausch statt: Alle schlafenden Knoten erwachen, um die aktiven Knoten zu ersetzen und erneut zu bestimmen, welche Knoten redundant sind. Welche Knoten nach jedem Rollentausch zuerst in einen Schlafmodus versetzt werden, kann beispielsweise anhand der verbleibenden Energie eines Knotens festgemacht werden.

2.4.2 Geographisches Clustering

Geographisches Clustering bezeichnet Cluster, deren Knoten durch räumliche Nähe zueinander ausgezeichnet sind. Die maximale räumliche Distanz eines Clusters ist von der Sendereichweite abhängig und überschreitet diese nur selten [41]. Im Gegensatz zum zeitlichen Clustering werden die im Sensornetzwerk zu absolvierenden Aufgaben beim geographischen Clustering häufig verschiedenen Knoten zugewiesen, so dass dementsprechend eine Rol-Dabei mindestens lenzuweisung erfolgt. existieren 2 Rollen. der Clustermember und der Clusterhead, wie in Abbildung 2.7 (C) dargestellt ist. Die Aufgaben eines Clustermembers bestehen in der Regel nur aus dem Abtasten der Umgebung mit den Sensoren und der Übermittlung der gemessenen Daten zum Clusterhead. Der Clusterhead nimmt mehrere zusätzliche Aufgaben wahr. Zum einen aggregiert er, falls möglich, die Daten seiner Sensorknoten, und sorgt für eine Weiterleitung der Daten in Richtung Senke. Zusätzlich kann ein Clusterhead dafür verantwortlich sein, die Funktionalität seines Clusters zu überprüfen und gegebenenfalls seine Rolle abzugeben, wie in Kapitel 3 beschrieben wird. Falls die Aufgaben eines Clusterheads so energieintensiv sind, dass sie einen häufigen Rollentausch erzwingen, können die Aufgaben eines Clusterheads auch auf mehrere Clustermember aufgeteilt werden, wie Abbildung 2.7 (D) veranschaulicht.

In der Literatur existieren verschiedene Ansätze, geographische Cluster und deren Einzugsbereiche zu bestimmen. Diese können ebenfalls lokalisierungsfrei- oder lokalisierungsbasiert sein. LEACH ist dabei der erste bekannte Algorithmus, welcher das Ziel hat, eine Energiebalance in einem drahtlosen Sensornetzwerk zu erreichen. Mit HEED wurde der Ansatz erweitert, in dem die Energiebalance verbessert und die Unterstützung von Multihop-Routen ergänzt wurde.

Der lokalisierungsbasierte Ansatz GAF, auf dem die in dieser Arbeit entwickelten Algorithmen basieren, verbindet die in LEACH und HEED etablierten Konzepte und kombiniert diese mit einer verbesserten Detektion der Clusterheads. Die drei geographisch clusternden Algorithmen LEACH, HEED und GAF werden im Folgenden vorgestellt.

LEACH

Der Low-Energy Adaptive Clustering Hierarchy (LEACH) Algorithmus wurde ursprünglich entwickelt, um zu verhindern, dass jeder einzelne Sensorknoten seine gesammelten Sensordaten zur entfernten Senke sendet [44]. Stattdessen werden die Daten zu einem Clusterhead gesendet, welcher die Daten aggregiert und gebündelt zur Senke schickt. Die Clusterheads werden dabei mit einem Zufallsprinzip ermittelt, bei welchem sich jeder Knoten zufällig und unabhängig entscheidet, ob er Clusterhead wird oder nicht. Die Clusterhead-Wahrscheinlichkeit P_{CH} ist dabei vorher festgelegt. Ein ebenfalls in [44] ermittelter Wert von $P_{CH}=5\%$ erweist sich dabei in als günstig.

Mittels eines Broadcasts benachrichtigt dann jeder Clusterhead seine benachbarten Knoten über seine Rolle. Empfängt ein Sensorknoten mehr als einen Clusterhead, entscheidet er sich für den näheren Clusterhead.

Um zu verhindern, dass die Clusterheads ihre Energie zu schnell verbrauchen, wird das Protokoll periodisch angewandt. Nach einer gewissen Zeit, hier Runde genannt, wird die entstandene Clusterstruktur aufgelöst. Zu Beginn der nächsten Runde werden die Clusterheads erneut durch Anwendung des Zufallsprinzips ermittelt. Mit dieser Einteilung der Lebenszeit des Netzwerks in Runden verteilen sich die Clusterheads wahrscheinlich im Laufe der Zeit über das gesamte Netzwerk, so dass eine gewisse Energiebalance zwischen den einzelnen Knoten erreicht wird.

Der LEACH Algorithmus enthält jedoch mehrere energieineffiziente Schwächen. Zum ersten kann es passieren, dass Sensorknoten mehrfach hintereinander zum Clusterhead gewählt werden, was zu einem frühzeitigen Knotenausfall jener führen kann. Zum zweiten ist es möglich, dass mehrere benachbarte Sensorknoten keinen Clusterhead erreichen und somit selbst mit der Senke kommunizieren müssen, was unnötig Energie verschwendet. Zum dritten ist eine direkte Kommunikation aller Clusterheads mit der Senke im Vergleich zu einer Multihop-Kommunikation Richtung Senke ineffizient, oder bei größeren Ausdehnungen eines Sensornetzwerks sogar unmöglich.

HEED

Der *Hybrid Energy-Efficient Distributed Clustering* (HEED) Algorithmus in [45] widmet sich den Schwächen durch einige Änderungen in LEACH, die im Folgenden kurz beschrieben werden.

Als erste Änderung wird für jeden Knoten die Clusterhead-Wahrscheinlichkeit P_{CH} anhand seiner verbliebenen Energie E_{Rest} nach Gleichung (2.9) angepasst.

$$P_{CH,HEED} = P_{CH} * \frac{E_{Rest}}{E_{Max}}$$
(2.9)

Da ein Knoten, welcher bisher als Clusterhead agiert hat, im späteren Verlauf deutlich weniger Energie zur Verfügung hat, sinkt seine Wahrscheinlichkeit bei einer erneuten Auswahlrunde, erneut als Clusterhead aktiv zu werden. Energiearme Knoten werden demzufolge mit einer höheren Wahrscheinlichkeit als bei LEACH geschont.

Genau wie bei LEACH benachrichtigt dann jeder Clusterhead benachbarte Knoten per Broadcast über seine Rolle. Empfängt ein Sensorknoten mehr als einen Clusterhead, entscheidet er sich ebenfalls für den näheren Clusterhead.

Im Gegensatz zu LEACH ist bei HEED der Prozess zur Clusterheadbestimmung damit nicht abgeschlossen. Knoten, die zu dem Zeitpunkt weder einen Broadcast empfangen haben noch selber Clusterhead sind, wiederholen den Prozess so lange, bis sie entweder selber Clusterhead sind oder einen Clusterhead in Reichweite haben. Dabei wird bei jedem erneuten Ausführen des Prozesses zur Clusterheadbestimmung die eigene Clusterhead-Wahrscheinlichkeit P_{CH} verdoppelt, so dass schließlich jeder Knoten entweder selbst Clusterhead wird oder einen benachbarten Knoten empfängt. Somit wird vermieden, dass bei der ersten Clusterheadauswahl übriggebliebene Knoten unnötig eigenständig als Clusterheads agieren.

Als weitere entscheidende Änderung zu LEACH wird in HEED eine direkte Kommunikation jedes Clusters zur Senke vermieden. Stattdessen wird eine Routingstruktur aufgebaut, welche einem Clusterhead erlaubt, mittels Multihop über andere Clusterheads mit der Senke zu kommunizieren. Im Gegensatz zu LEACH, bei der jeder Clusterhead direkt zur Senke kommuniziert, kann somit eine erhebliche Menge von Energie gespart werden.

Sowohl LEACH als auch HEED weisen die Schwäche auf, keine Garantie über die Dichte von Clusterheads geben zu können. Entstehen zu viele Clusterheads, ist der Energievorteil des Weiterleitens aggregierter Daten gering. Entstehen zu wenig Clusterheads, kann eine Multihop-Kommunikation nicht mehr gewährleistet werden.



Abbildung 2.9 Angewandter GAF Algorithmus, (A) Ermittlung der maximalen Zellgröße, (B) Routing im geclustertem Netzwerk

GAF

Geographical adaptive fidelity (GAF) [43] ist ein für Ad-hoc Netzwerke entwickeltes Clusteringverfahren, welches die oben genannten Nachteile von LEACH und HEED behandelt. Ziel dieses Verfahrens ist es, für die Datenübertragung redundante Kommunikationsknoten zu finden und die Weiterleitung über diese zu vermeiden. Dazu wird in GAF ein virtuelles zwei-dimensionales Gitter verwendet, welches über die Netzwerkfläche gelegt wird. So entstehen im Netzwerk virtuelle quadratische Zellen. Ein Knoten in jeder dieser Zellen wird zum Clusterhead ernannt, wodurch, bei ausreichender Anzahl besetzter Zellen, eine gewisse Dichte von Clusterheads gewährleistet ist. Um die Zellen zu ermitteln, wird in GAF davon ausgegangen, dass eine garantierte Sendereichweite R_T vorliegt. Die maximale Seitenlänge L_{max} einer entstehenden Zelle ist nach [43] in Gleichung (2.10) definiert und in Abbildung 2.9 (A) visualisiert.

$$L_{max,GAF} = \frac{R_T}{\sqrt{5}} \tag{2.10}$$

Solange die maximale Seitenlänge einer Zelle den Wert L_{max} nicht überschreitet, ist dementsprechend garantiert, dass zwei Kommunikationsknoten, welche orthogonal benachbart sind, miteinander kommunizieren können. Das bietet den entscheidenden Vorteil, dass bis auf einen Knoten innerhalb eines Clusters, den Clusterhead, sämtliche Knoten innerhalb einer Zelle temporär abgeschaltet werden können.

Solange genug Zellen besetzt sind, kann mittels der aktiven Clusterheads dennoch eine Nachricht von jedem Punkt des Netzwerks jeden anderen erreichen, wie Abbildung 2.9 (B) zeigt.

Da die zum Routing benötigten Knoten temporär abgeschaltet werden können, verfügt der Algorithmus über das Potential, erheblich weniger Energie zu benötigen als die bisher beschriebenen geographischen Clusteringalgorithmen. Die Energieersparnis hängt dabei direkt von der Knotendichte ab.

Der GAF Algorithmus setzt für jeden Knoten zwei Dinge voraus. Zum einen benötigt der Knoten Wissen über die Lage der virtuellen Zellen, zum anderen

muss die Position jedes Knotens so genau bestimmbar sein, dass er sich der richtigen, ihm zugehörigen Zelle zuordnen kann.

Aufgrund der einfachen Umsetzung und des Potentials des GAF Algorithmus dient dieser als Basis für eine Reihe der im Laufe dieser Arbeit entwickelten Algorithmen.

2.5 Kurzzusammenfassung

Dieses Kapitel hatte die Aufgabe, einen Einstieg in das Gebiet der drahtlosen Sensornetzwerke zu gewähren. Zunächst wurden dazu in Kapitel 2.1 die für das Verständnis der Arbeit notwendigen kommunikationstechnischen Grundlagen erläutert. Anschließend wurde in Kapitel 2.2 der Begriff der drahtlosen Sensornetzwerke erläutert, deren Analyse Thema der Arbeit ist. Darauf aufbauend folgte in Kapitel 2.3 die Diskussion der Herausforderungen, die bei dem Einsatz eines drahtlosen Sensornetzwerks zu bewältigen sind. Im Anschluss dazu wurde in Kapitel 2.4 auf den Begriff des Clusterings und bekannte Clustering-Verfahren aus der Literatur eingegangen, mit welchen den Herausforderungen begegnet werden kann.

Mit dieser Struktur wurde somit das Verständnis für die nachfolgenden Kapitel geschaffen.

3 Selbstorganisation in drahtlosen Sensornetzwerken

Die aus Hunderten von Sensorknoten bestehenden drahtlosen Sensornetzwerke lassen sich bereits durch die schiere Anzahl autonom agierender Komponenten mit biologischen Systemen wie Vogelschwärmen oder Ameisenkolonien vergleichen. Zusätzlich können Sensorknoten, ähnlich den Mitgliedern eines solchen biologischen Systems, häufig nur aufwändig Wissen über ihre Umgebung, ihre Nachbarn, oder die Gesamtsituation des Systems erhalten.

Aufgrund der begrenzten Energieressourcen findet daher in nahezu jedem drahtlosen Sensornetzwerk mehr oder minder ein Prozess statt, der ähnlich dem biologischen Vorbild als Selbstorganisation bezeichnet wird.

Das vorliegende Kapitel dient dazu, den Begriff Selbstorganisation in drahtlosen Sensornetzwerken zu erläutern. Während sich Abschnitt 3.1 mit der allgemeinen Selbstorganisation im Kontext drahtloser Sensornetzwerken und deren Anwendung beschäftigt, widmet sich Abschnitt 3.2 der Selbstheilung, welche die Grundlage für die in Kapitel 6 und Kapitel 7 entwickelten Clusteringalgorithmen bildet.

3.1 Grundlagen der Selbstorganisation

Selbstorganisierte Systeme sind allgegenwärtig, und können in nahezu allen biologischen Systemen beobachtet werden. Als populäres Beispiel gilt der Vogelschwarm [47], der selbstorganisiert eine Flugformation auswählt, in welcher der gesamte Schwarm für die Fortbewegung deutlich weniger Energie benötigt, als die gleiche Anzahl unor-

ganisierter Individuen es täte. In [4] werden einem selbstorganisierten System die in

TABELLE 3.1 genannten Eigenschaften zugeordnet, welche das System von einem zentral gesteuerten System unterscheiden.

Wendet man die vorliegenden Eigenschaften auf ein drahtloses Sensornetzwerk an, so erkennt man, dass die Eigenschaften "Emergente Strukturen" und "Entstehende Komplexität" erfüllt werden. Bei einem geclusterten Netzwerk beispielsweise entsteht eine Struktur aus Sensorknoten und Clusterheads, ohne dass mit den an der Senke vorliegenden Informationen vorhergesagt oder berechnet werden kann, welcher Knoten welche Rolle einnimmt.

Bei den Eigenschaften "Keine zentrale Kontrollinstanz" und "Hohe Skalierbarkeit" weicht ein Sensornetzwerk jedoch von der Definition ab.

Ein drahtloses Sensornetzwerk besitzt für gewöhnlich eine zentrale Instanz, die Senke. Diese ist in der Lage, Anweisungen in das Netzwerk einzugeben, und so das Verhalten des Sensornetzwerks zu beeinflussen. Allerdings ist die Senke nicht in der Lage, jeden einzelnen Knoten direkt zu kontrollieren.

Auch die hohe Skalierbarkeit ist nur bedingt erfüllt. Werden einem drahtlosen Sensornetzwerk zusätzliche Individuen, also Sensorknoten, außerhalb des bereits überwachten Gebietes hinzugefügt, so müssen mehr Hops zurückgelegt werden, um diese Knoten zu erreichen. Das belastet die näher an der Senke liegenden Knoten stärker und schränkt unter Umständen die Gesamtperformance ein.

Obwohl die Eigenschaften selbstorganisierter System nur bedingt erfüllt sind,

EIGENSCHAFT	BESCHREIBUNG		
Keine zentrale Kontrollinstanz	Es existiert kein globales Kontrollsystem und es sind keine glo- balen Informationen verfügbar. Jedes Subsystem muss komplett autonom agieren		
Emergente Strukturen	Das globale Verhalten des Systems emergiert zu erkennbaren Strukturen		
Entstehende Komplexität	Selbst wenn die einzelnen Individuen eines Systems nur einfac Regeln befolgen können, ist das Verhalten des Gesamtsystem komplex und unvorhersehbar		
Hohe Skalier- barkeit	Es existiert kein Performanceverlust, falls zusätzliche Individ zu dem System hinzugefügt werden, und das System verhält trotzdem wie gefordert.		

 TABELLE 3.1 EIGENSCHAFTEN SELBSORGANISIERTER SYSTEME NACH [4]

wird im Zuge dieser Arbeit für Sensornetzwerke dennoch der Begriff selbstorganisiert verwendet, da die Tatsache, dass einzelne Individuen nicht von einer zentralen Instanz aus kontrolliert werden, maßgeblich für die entwickelten Algorithmen ist.

Im Zuge der Entwicklung energieeffizienter Algorithmen für drahtlose Sensornetzwerke, insbesondere Clusteringalgorithmen, wird die Art solch selbstorganisierender Algorithmen als *Localized Algorithms* [48, 49] beschrieben. Diese Klasse der Algorithmen zeichnet sich dadurch aus, dass jedes Individuum nur eine Kommunikation mit seinen direkten Nachbarn benötigt, während die zentrale Instanz in der Lage ist, notwendige Anweisungen in das Netzwerk hinein zu geben. Als Resultat ist zwar nicht vorhersehbar, welches Individuum aufgrund der Anweisungen der Senke welchen Zustand angenommen hat, jedoch kann davon ausgegangen werden, dass die Anweisung der Senke den Möglichkeiten entsprechend umgesetzt wurde.

Die in Kapitel 6 sowie in Kapitel 7 entwickelten selbstorganisierten Clusteringalgorithmen zählen alle zur Gruppe der *Localized Algorithms*, da zwar vorgegeben wird, wo oder wie die Cluster entstehen sollen, jedoch die aktuelle Aufgabe jedes einzelnen Knotens selbstorganisiert ermittelt wird.

3.2 Selbstheilung

Das Prinzip der Selbstheilung als Teilkonzept der Selbstorganisation erlaubt einem System, seine Funktionalität in einem Fehlerfall mittels der Aufwendung von zusätzlichen Systemressourcen autonom wiederherzustellen.

Im Kontext der Sensornetzwerke ist die angestrebte Funktionalität, überall im zu überwachenden Gebiet Phänomene detektieren zu können und im Falle der Detektion zeitnah eine Nachricht an die Senke senden zu können. Durch den Einsatz eines Sensornetzwerks in unzugänglichen Gebieten sind die einzelnen Sensorknoten verschiedenen Gefahrenquellen ausgesetzt, die zu einem Ausfall eines Knotens führen können. Als Beispiele seien hier der Abwurf der Knoten aus einem Flugzeug und die damit verbundene mechanische Belastung oder der Einfluss der Witterung auf die Knoten genannt.

Durch die verschiedenen Fehlerquellen fallen Knoten nicht sofort aus, vielmehr ist es möglich, den Grad der Beschädigung eines Knotens in die folgenden vier Kategorien zu unterteilen [50].

Unbeschädigte Knoten (engl. Non impaired node, kurz: NIN)

Unbeschädigte Knoten sind Knoten, deren Sensoren und Transmitter den Erwartungen entsprechen. Diese Knoten sollten die Mehrheit der Sensorknoten im Netzwerk ausmachen.

Sensorbeschädigte Knoten (engl. Sensor impaired node, kurz: SIN)

Ein Knoten dieser Kategorie ist entweder nicht mehr in der Lage, Phänomene zu detektieren, oder detektiert Phänomene, obwohl gar kein Phänomen vorliegt. Gründe können eine defekte Sensorik oder eine unglückliche Platzierung sein.

Übertragungsbeeinträchtigte Knoten (engl. Transceiver impaired node, kurz TIN)

In diese Kategorie fallende Knoten sind nicht mehr in der Lage, sämtliche erreichbare Knoten zu erreichen, die für einen NIN in einer ähnlichen Position erreichbar wären. Grund für einen TIN können beschädigte Antennen oder ebenfalls eine ungünstige und stark abgeschattete Knotenplatzierung sein.

Ausgefallene Knoten

Knoten dieser vierten Kategorie besitzen entweder keine Möglichkeit (mehr), mit ihrer Umgebung zu kommunizieren, oder sind beispielsweise aufgrund verbrauchter Batterien komplett ausgefallen. In beiden Fällen können die Knoten nicht mehr am Netzwerk partizipieren, beeinflussen jedoch auch das restliche Netzwerk nicht mehr.

Im Laufe des Netzbetriebs entstehen aufgrund der verbrauchten Batterien immer mehr ausgefallene Knoten, bis schließlich kein NIN mehr vorhanden ist. Um eine Aussage über den Anteil von SIN und TIN zur Laufzeit zu treffen, müssen die Einsatzumgebung des Sensornetzwerks und die Empfindlichkeit der Sensorknoten bekannt sein. Je nach Einsatzumgebung wird ein Knotenzur Laufzeit mehr oder weniger mechanischem Stress, Feuchtigkeit oder anderen funktionsbeeinträchtigen Effekten ausgesetzt sein, was zu zusätzlichen SIN und TIN führen kann.

Mittels redundant platzierter Knoten ist ein Sensornetzwerk in der Lage, sowohl auf ausgefallene als auch auf beschädigte Knoten entsprechend reagieren zu können.

Für eine selbstständige Heilung muss das Netzwerk zuerst in der Lage sein, beschädigte und ausgefallene Knoten zu erkennen. Das Erkennen eines ausgefallenen Knotens erscheint relativ einfach, und kann mittels periodisch empfangener *Life Sign* Nachrichten ermittelt werden. Das Erkennen eines beschädigten Knotens dagegen ist weitaus komplexer, da man herausfinden muss, ob beispielsweise ein Knoten permanent Phänomene detektiert, oder eine beschädigte Sensorik besitzt. Strategien zur Erkennung fehlerhafter Knoten werden in [51] und in dem im Zuge dieser Arbeit entwickelten Algorithmus ELDEN (Efficient Localized Detection of Erroneous Nodes) [52] beschrieben, stehen jedoch nicht im Fokus der vorliegenden Arbeit.

Nachdem ein beschädigter oder ausgefallener Knoten identifiziert wurde, hat das Netzwerk die Möglichkeit entsprechend zu reagieren. Während das Erken-

	Clustermember	Clusterverwalter	Aggregator	Router
	•	V	A	R
Unbeschädigter Knoten	*	~	V	v
Sensorbeschädigter Knoten	*	v	*	v
Übertragungsbeeinträchtigter Knoten	*	v	*	×
Ausgefallener Knoten	*	×	×	×

Abbildung 3.1 Knoten unterschiedlicher Beschädigungsgrade und erlaubte Rollen

nen eines ausgefallenen Knotens nur möglich ist, nachdem der Knoten tatsächlich ausgefallen ist, bietet das frühzeitige Erkennen eines beschädigten Knotens die Möglichkeit, den Knoten mit einer anderen Funktionalität weiter zu betreiben. Aufgrund dieser beiden unterschiedlichen Strategien zur Detektion beschädigter bzw. ausgefallener Knoten unterscheidet man zwischen präventiver und reaktiver Heilung, auf die im Folgenden kurz eingegangen wird.

3.2.1 Präventive Heilung

Ist es möglich, beschädigte Knoten weiter in den Netzbetrieb zu integrieren, so kann man dies als präventive Heilung [50] bezeichnen. Diese Art der Heilung ist insbesondere in einem geclusterten Netzwerk möglich, bei welchem den Knoten unterschiedliche Aufgaben zugewiesen worden sind. Wird beispielsin Abbildung 2.7 (D) geclusterte Netzwerk als Referenz weise das herangezogen, so erkennt man die Einteilung in Sensorknoten, Router, Aggregator und Clusterverwalter mit den entsprechenden Aufgaben. Sollte die Clustergröße so gewählt sein, dass ein problemloser Rollentausch möglich ist, könnte ein Knoten mit beschädigter Sensorik beispielsweise die Aufgabe des Clusterverwalters übernehmen, während die Netzwerkfunktionalität nicht beeinträchtigt würde. Ebenso könnte ein übertragungsbeeinträchtigter Knoten möglicherweise noch die Aufgabe des Sensorknotens oder Clusterverwalters übernehmen, da diese nur innerhalb ihres Clusters kommunizieren müssen. Eine vollständige Auflistung der in Kategorien eingeteilten Knoten ist in Abbildung 3.1 zu sehen.

Die im Zuge dieser Arbeit entwickelten Clusteringalgorithmen verwenden prinzipiell nur 2 verschiedene Rollen. Den Clusterhead, welcher die Clusterverwaltung, die Datenaggregation, das Weiterleiten der Daten und die Umweltüberwachung mittels Sensorik vereint, und den Clustermember, welcher sich zumeist in einem Schlafmodus befindet und nur zeitweise aufwacht, um das Vorhandensein seines Clusterheads zu überprüfen und sich zu synchronisieren.

Sollte das Vorhandensein beschädigter Knoten festgestellt werden, so können den in Abbildung 3.1 verwendeten Rollen entsprechend Knoten zugewiesen werden, falls damit die Lebensdauer des gesamten Clusters erhöht werden kann. Eine präventive Heilung ist daher bei allen in Kapitel 6 und Kapitel 7 entwickelten Algorithmen möglich.

3.2.2 Reaktive Heilung

Der Begriff der reaktiven Heilung wurde als *Rehabilitation* ebenfalls in [50] eingeführt. Diese Art der Selbstheilung versucht, ausgefallene Knoten durch bereits vorhandene, bislang inaktive Knoten zu ersetzen. Sollte das Netzwerk in Cluster aufgeteilt sein, deren Mitglieder untereinander problemlos austauschbar sind, so ist dies ohne größeren Kommunikations- oder Rechenaufwand möglich. In [53] zeigen die Autoren, dass ein selbstorganisierter Ansatz dabei eine deutlich energieeffizientere Heilung ermöglicht als ein zentralisierter Ansatz, insbesondere wenn sich der Cluster mehrere Hops von der Senke entfernt befindet.

Durch die maximale Größe der in Kapitel 6 und Kapitel 7 entwickelten Cluster ist garantiert, dass der Austausch eines Knotens innerhalb eines Clusters ohne weitere Einschränkungen möglich ist, die reaktive Heilung daher ebenso wie die präventive Heilung anwendbar ist. Insbesondere muss hierbei nur ein einzelner Knoten innerhalb desselben Clusters aktiviert werden, um einen ausgefallenen Knoten vollständig zu heilen.

Darüber hinaus bieten die in Kapitel 6 entwickelten Algorithmen teilweise die Möglichkeit, eine reaktive Heilung auf Clusterebene durchzuführen.

3.3 Kurzzusammenfassung

Das vorliegende Kapitel hatte die Aufgabe, eine Einordnung der Struktur von Sensornetzwerken in den Kontext Selbstorganisation zu geben. Kapitel 3.1 stellt dazu die Grundzüge der Selbstorganisation dar und vergleicht diese mit den Eigenschaften drahtloser Sensornetzwerke. Kapitel 3.2 geht auf die Bedeutung der Selbstheilung ein.

Mit den Prinzipien der Selbstorganisation und insbesondere der Selbstheilung ist es möglich, die Lebenszeit der in den nachfolgenden Kapiteln entwickelten Clusteringverfahren drastisch zu steigern.

4 Evaluierung von Algorithmen für drahtlose Sensornetzwerke

Ein für Sensornetzwerke entwickelter idealer Clusteringalgorithmus zeichnet sich gegenüber anderen Algorithmen dadurch aus, dass ein Sensornetzwerk damit besonders lange besonders effizient seine Aufgabe erfüllen kann. Das vorliegende Kapitel motiviert, nach welchen Gesichtspunkten die entwickelten Algorithmen sinnvoll evaluiert werden können. Da ein periodisches Ausbringen Hunderter realer Sensorknoten ein kosten- und zeitineffizienter Prozess ist, verwendet man für die Evaluierung von Sensornetzwerken Sensornetzwerksimulatoren. In Abschnitt 4.1 werden daher der verwendete Netzwerksimulator und die zur Simulation verwendeten Modelle erläutert, während sich Abschnitt 4.2 den Evaluierungskriterien widmet, um schließlich auch unterschiedlich arbeitende Algorithmen und deren Effizienz miteinander vergleichen zu können.

4.1 Netzwerksimulator

Um die bereits existierenden und in der Arbeit entwickelten Algorithmen vergleichen zu können, ist es notwendig, diese in einer gemeinsamen Simulationsumgebung einzuordnen. Die für die vorliegende Arbeit betrachteten Netzwerksimulatoren ns-2 [54], J-Sim [55], Opnet [56], Glomosim [57], Prowler [58] und Rmase [59] bieten teilweise nur bedingt die Fähigkeit, mehrere hundert Sensorknoten effektiv zu simulieren. Prowler (*engl. Probalistic Wireless Network Simulator*) erschien jedoch aus mehreren Gründen als die geeignetste Wahl, um die entwickelten Algorithmen simulieren und analysieren zu können.

Zum einen bietet Prowler verschiedene Kanalmodelle, eine MAC-Schicht in Form von CSMA, und die Fähigkeit, mehrere Hundert Knoten zu simulieren. Zum anderen ist der Quellcode in Matlab verfasst, bietet also die Möglichkeit, schnell auf die Bedürfnisse angepasst zu werden. Die einzelnen verwendeten Modelle werden im Folgenden beschrieben.

4.1.1 Anzahl simulierter Sensornetzwerke

Um eine Aussage über die Leistungsfähigkeit eines Algorithmus treffen zu können, wurde die Analyse jedes untersuchten Algorithmus' in einer Vielzahl von Sensornetzwerken durchgeführt. Durch eine zufällige Anordnung der Knoten in jedem einzelnen Szenario konnten Durchschnittswerte über verschiedene Knotenanordnungen ermittelt werden. Je nach Algorithmus wurden zwischen 100 und 500 Sensornetzwerke simuliert, was geeignet genug ist, um Aussagen über die Performance treffen zu können, und gleichzeitig nicht den Simulations- und Auswertungsaufwand überschreitet.

Für die Lokalisierungsalgorithmen in Kapitel 5 wurden aufgrund der schnellen Berechnungszeiten für jeden Datenpunkt mindestens 1000 Lokalisierungen simuliert.

4.1.2 MAC-Modell

Da die Untersuchung eines geeigneten Medienzugriffs nicht im Fokus dieser Arbeit stand und ein effizienter MAC für jeden Clusteringalgorithmus ähnliche Performance erzielen würde, wurde der MAC auf ein kollisionsfreies TDMA vereinfacht.

4.1.3 Routing-Modell

Da die Untersuchung eines geeigneten Routings ebenfalls nicht im Fokus dieser Arbeit stand, wird davon ausgegangen, dass für sämtliche geclusterten Netzwerke von der Senke aus ein Routingbaum aufgebaut wird, welcher die notendigen Daten verlustfrei an die Senke übertragen kann, und für jeden Knoten eine günstige Anzahl an notwendigen Hops wählt. Diese Vereinfachung erlaubt es, die Algorithmen ohne den Einfluss verschiedener Routingstrategien zu vergleichen.

4.1.4 Energiemodell

Die Performance eines Clusteringalgorithmus hängt entscheidend mit der verbrauchten Energie zusammen. Da Prowler kein eigenes Energiemodell bietet, wurde im Zuge dieser Arbeit ein stark vereinfachtes Energiemodell implementiert. Dieses basiert auf den Erkenntnissen der Tabellen TABELLE 2.2 und TABELLE 2.3 in Kapitel 2.2 und sieht Folgendes vor:

- 1.) Sensorknoten, die sich in einem Schlafzustand befinden, verbrauchen keinerlei Leistung; $P_{Sleep}=0$
- 2.) Die Leistung, die zum Initialisieren des Sensornetzwerks sowie zum eventuellen Lokalisieren der Sensorknoten benötigt wird, ist sehr gering im Verhältnis zu der Gesamtenergie eines Knotens, und kann daher vernachlässigt werden; $P_{init}=0$
- 3.) In den untersuchten Sensornetzwerken wird davon ausgegangen, dass Phänomene, die detektiert werden, unverzüglich die Senke erreichen müssen. Daher befindet sich jeder aktive Knoten permanent im Idle Listening Zustand und kann sich nicht leisten, periodisch in einen Schlafzustand zu fallen. Daraus abgeleitet ergibt sich, dass jeder aktive Knoten dieselbe Leistung verbraucht; $P_{akt}=1$

4.) Die Leistung, die für die interne Organisation und den Rollentausch innerhalb eines Clusters, sowie eine wie in Kapitel 6.2 beschriebene Heilung benötigt wird, ist ebenfalls sehr gering, und wird daher ebenfalls vernachlässigt; $P_{Rest}=0$

Jeder Knoten verbraucht also entweder keine oder maximale Leistung. Dies vereinfacht die Berechnung einer Lebenszeit des Netzwerks, obwohl die Realitätsnähe natürlich leidet, da beispielsweise Knoten nahe der Senke entweder häufiger senden oder stärker aggregieren müssen. Dies betrifft jedoch zum einen alle Clusteringalgorithmen gleichermaßen, sodass die Vereinfachung dennoch einen Vergleich der unterschiedlichen Algorithmen erlaubt. Zum anderen kann davon ausgegangen werden, dass aktive Sensorknoten sich ohnehin permanent im Datenempfangsmodus befinden müssen, damit die Senke zeitnah über erkannte Phänomene informiert werden kann.

4.1.5 Kanalmodell

Für viele einfache Simulationen und das Testen der Algorithmen wurde im Zuge dieser Arbeit das *Unit disk graph* Modell verwendet. Um realitätsnähere Simulationen zu erhalten und die Performance der in den nachfolgenden Kapiteln entwickelten Algorithmen zu vergleichen, wurde jedoch auch ein Kanalmodell benötigt, welches Abschattung und Dämpfung realitätsnah simuliert.

Das von Prowler zur Verfügung gestellte Kanalmodell ähnelt vom Verhalten dem der Freiraumdämpfung. Das Verhältnis von Empfangsleistung P_R und Sendeleistung P_T verringert sich quadratisch mit der Distanz *d* zwischen Sender und Empfänger, wie Gleichung (4.1) zeigt.



Abbildung 4.1 Vergleich der ermittelten Leistung am Empfänger bei verschiedenen Kanalmodellen; eingestellte Parameter: f=900 MHz, $G_T=G_R=1 P_T=0 dBm$

$$P_R = P_T * \frac{1}{1 + d[m]^2} * (1 + X_\tau)$$
(4.1)

Die Effekte der Abschattung werden ähnlich dem *Log-Normal Shadow Fading* Modell durch eine normalverteilte, mittelwertfreie und mit Standardabweichung τ ermittelte Zufallszahl X_{τ} simuliert. Im Gegensatz zum *Log-Normal Shadow Fading* ist es so möglich, auch eine negative Empfangsleistung zu erhalten, die von Prowler entsprechend auf 0 gesetzt wird.

Da das Kanalmodell von Prowler im Gegensatz zu dem bereits in Kapitel 2.1.1 vorgestellten *Log-Normal Shadow Fading* Modell weder Frequenzen oder unterschiedliche Pfadverlustexponenten berücksichtigt, noch σ für verschiedene Szenarien bekannt sind, wurde das Prowler Kanalmodell im Zuge dieser Arbeit auf Log-Normal Shadow Fading umcodiert.

Als Antennengewinne wurden der Einfachheit halber $G_T=G_R=1$ angenommen Damit ergibt sich die resultierende Empfangssignalstärke wie in Gleichung (4.2) beschrieben.

$$P_{R} = P_{T} * \left(\frac{c}{4\pi f * 1m}\right)^{2} * \left(\frac{1}{d[m]}\right)^{n} * 10^{\frac{X_{\sigma}}{10}}$$
(4.2)

Die Frequenz wurde mit 900 Mhz auf eine für drahtlose Sensornetzwerke relevante Frequenz eingestellt [10]. Werte für den Pfadverlustexponent n und ein entsprechendes σ finden sich ebenfalls in [10].

Um den Unterschied der Kanalmodelle deutlich zu machen, wurden in Abbildung 4.1 die Freiraumdämpfung sowie die Modelle *Unit disk graph*, das voreingestellte Prowler Modell und *Log-Normal Shadow Fading* dargestellt. Zusätzlich ist das in den Simulationen verwendete Limit für eine minimal benötigte Signalstärke von *-98 dBm* am Empfänger eingetragen. Dies entspricht dem Wert aus dem Datenblatt des *MICA2* Knotens [7].

4.1.6 Szenarien

Um die Untersuchungen des Clusterings und der Lokalisierung realitätsnah zu gestalten, ist es nicht ausreichend, die idealen Übertragungsbedingungen, die

EIGENSCHAFT	Szenario				
	Ideale Bedingun- gen	Flacher Sandstrand	Trockenes Unterholz		
Pfadverlust- exponent	Reichweite ein- stellbar	4,2	3,6		
Standardab- weichung	0	2	2,9		

 TABELLE 4.1 KANALMODELL DER SIMULIERTEN SZENARIEN NACH [10]

durch ein Unit disk graph Modell gegeben sind, zu verwenden. Daher wurden neben dem als "Ideale Bedingungen" bezeichneten Szenario aus [10] zwei Szenarien ausgewählt, die für drahtlose Sensornetzwerke typisch sind. Diese sind der "Flache Sandstrand", also das Ausbringen eines Sensornetzwerks in einer relativ flachen, aber dafür welligen Umgebung und das "Trockene Unterholz", welches für Szenarien wie die Waldbranderkennung relevant ist.

Die aus [10] ermittelten Parameter für den Pfadverlustexponent und die Standardabweichung der Signalstärke sind in TABELLE 4.1 dargestellt. In Abbildung 4.1 ist zu erkennen, dass sich für den "Flachen Sandstrand" eine durchschnittliche Sendereichweite von 38,25 m ergibt, während sich für das Szenario "Trockenes Unterholz" mit den dort eingestellten Werten eine durchschnittliche Reichweite von 70,2 m ergibt.

4.1.7 Sensorreichweite

Ähnlich der Fähigkeit zu kommunizieren muss auch die Fähigkeit betrachtet werden, mit welcher ein Phänomen detektiert werden kann. Im Gegensatz zur Kommunikation existiert für die Detektion von Phänomenen jedoch kein explizites Modell. Im Zuge der vorliegenden Arbeit wird daher das *Unit disk graph* Modell verwendet, um die Sensorfähigkeit eines Knotens zu beschreiben. Jedem Knoten im Sensornetzwerk wird dabei eine Sensorreichweite R_S zugeordnet, welche die maximale Distanz darstellt, bis zu welcher ein Phänomen detektiert werden kann. In der Literatur wird die Sensorreichweite häufig als in einer ähnlichen Größenordnung liegend wie die Sendereichweite betrachtet [42]. Dementsprechend wird für die entwickelten Algorithmen die Sensorreichweite um Faktor 1 bis 2 kleiner als die Sendereichweite betrachtet.

4.1.8 Randgebiete

Aufgrund seiner begrenzten Ausdehnung besitzt jedes Sensornetzwerk ein Randgebiet, in welchem sich die Bedingungen vom Kerngebiet unterscheiden. So stehen den Knoten beispielsweise weniger Kommunikationspartner zur Verfügung, oder ein Routing ist nur in bestimmte Richtungen möglich. Dass diese veränderten Eigenschaften die Fähigkeiten eines Sensornetzwerks beeinflussen, wurde im Zuge dieser Arbeit beispielhaft in [60] gezeigt. Hier wird belegt, dass sich in Randgebieten eines Sensornetzwerks die Lokalisierungsgenauigkeit erheblich verschlechtert.

Obwohl die Untersuchung der Randgebiete ein Bestandteil der Evaluierung von Sensornetzwerkalgorithmen ist, stand sie nicht im Fokus dieser Arbeit, da zum direkten Vergleich das Verhalten im Kerngebiet von entscheidenderer Bedeutung ist. Die Randgebiete wurden daher je nach Simulation auf einer der zwei im Folgenden vorgestellten Wege eliminiert.



Abbildung 4.2 Simulationsumgebung zur Vermeidung von Randgebieten, (A) Torus-Welt, (B) Simulationsgebiet > Evaluationsgebiet

Torus-Welt

Die Idee der Torus-Welt ist es, eine quadratische Fläche so zu simulieren, als ob sie keine Kanten hätte. Funksignale, die auf einer Seite eine Kante der Fläche erreichen, treten an der gegenüberliegenden Kante wieder ein. Dieses Verhalten ähnelt der Bewegung entlang der Oberfläche eines Torus. Ein Beispiel für eine Torus-Welt ist in Abbildung 4.2 (A) dargestellt. Für ein sich aus dem Zentrum aufbauendes Clustering eignet sich die Torus-Welt jedoch nicht, da am Rand dennoch Cluster entstehen würde, die sich zu denen im Zentrum unterschieden.

Simulationsgebiet > Evaluationsgebiet

Die zweite Möglichkeit, die durch Ecken und Kanten der simulierten Fläche entstehende Effekte zu vermeiden, besteht darin, den Algorithmus auf einer simulierten Fläche mit Randeffekten ablaufen zu lassen, die Evaluation des Algorithmus jedoch auf einer deutlich kleineren Fläche durchzuführen, sodass Randeffekte nicht zum Tragen kommen. Ein Beispiel für ein solch kleineres Evaluationsgebiet ist in Abbildung 4.2 (B) dargestellt. Man erkennt, dass die simulierte Fläche so gewählt wurde, dass sie mindestens eine Sendereichweite in jede Richtung größer ist als die evaluierte Fläche. Da die gesamte simulierte Fläche genau wie die evaluierte Fläche mit Sensorknoten aufgefüllt werden muss, ist der Aufwand bei dieser Methode deutlich höher. Da Prowler jedoch die Torus-Welt zum einen nicht unterstützt, zum anderen die Algorithmen aus Kapitel 7 durch eine Torus-Welt in ihrem Verhalten beeinflusst würden, wurden die in Prowler durchgeführten Simulationen zur Vermeidung von Effekten durch Randgebiete durch das Verwenden einer größeren Simulationsfläche durchgeführt.

4.2 Netzwerkbewertung

Um die entwickelten Algorithmen vergleichen zu können, müssen Evaluierungskriterien festgelegt werden, nach welchen entschieden wird, wie gut und wie lange ein drahtloses Netzwerk in der Lage ist, die von ihm geforderten Aufgaben mittels der entwickelten Algorithmen zu erfüllen.

Die im Folgenden vorgestellten Evaluierungskriterien dienen der entsprechenden Bewertung der entwickelten Algorithmen.

4.2.1 Realistic Case Coverage

Bei den im Zuge dieser Arbeit entwickelten Clusteringalgorithmen wurden Cluster entwickelt, bei denen sich die Mehrheit aller im Sensornetzwerk verfügbaren Sensorknoten in einem Schlafmodus befindet. Das hier als *Realistic Case Coverage* bezeichnete Bewertungskriterium führt einen entsprechenden Clusteringalgorithmus aus, wie beispielsweise in Abbildung 4.3 (A) dargestellt, und deaktiviert jeweils entsprechend der Algorithmusvorgabe einen Anteil der Sensorknoten, siehe Abbildung 4.3 (B). Daraufhin wird ein engmaschiges Raster über die evaluierte Fläche gelegt, und für jeden Punkt des Rasters wird geprüft, ob er sich in der Sensorreichweite zumindest eines Knotens befindet,



Abbildung 4.3 Bewertung der Coverage-Fähgikeit eines Sensornetzwerks, (A) geclustertes Sensornetzwerk, (B) Deaktivierung aller Knoten außer der Clusterheads, (C) Raster über die zu evaluierende Fläche legen zur Ermittlung der Realistic Case Coverage, (D) Ermittlung der kleinsten garantiert abgedeckten Fläche eines Clusters, (E) garantiert abgedeckte Fläche im Netzwerk, (F) Raster über die zu evaluierende Fläche legen zur Ermittlung der Worst Case Coverage

wie in Abbildung 4.3 (C) gezeigt ist. Falls alle Rasterpunkte komplett abgedeckt sind, gilt auch das zu überwachende Gebiet als komplett abgedeckt. Mit der Realistic Case Coverage lässt sich also abbilden, wie viele Knoten nötig sind, um bei einem normal arbeitenden Sensornetzwerk komplette Funktionalität zu erreichen. Um mittels der Realistic Case Coverage eine möglichst genaue Aussage über die tatsächlich abgedeckte Fläche treffen zu können, wird im Zuge dieser Arbeit der Rasterabstand auf 1*m* gesetzt.

4.2.2 Worst Case Coverage

Die Realistic Case Coverage besitzt den Nachteil, dass ein aktivierter Knoten innerhalb eines Clusters zufällig aktiviert wird. Dadurch können potentielle Löcher in der Sensorabdeckung durch eine glückliche Auswahl aktiver Knoten kaschiert werden. Dies passiert insbesondere bei Clusteralgorithmen, deren Clustergröße zu groß gewählt wurde, um eine komplette Abdeckung garantieren zu können. Um die so kaschierten Löcher aufdecken zu können, berechnet die *Worst Case Coverage* in einem ersten Schritt die Schnittmenge der durch die Sensorreichweiten aufgespannten Kreise der Knoten eines Clusters, wie Abbildung 4.3 (D) zeigt. Anschließend dient diese Schnittmenge als Sensorfähigkeit jedes Clusters, siehe Abbildung 4.3 (E).

Zur Ermittlung der Abdeckung wird daraufhin wieder ein feinmaschiges Raster über die evaluierte Fläche gelegt. Ein Rasterpunkt im Sensornetzwerk gilt nur als abgedeckt, solange er sich mindestens in einer Schnittmenge befindet. Es sind also immer gleich viele oder weniger Rasterpunkte abgedeckt als bei der Realistic Case Coverage, wie auch Abbildung 4.3 (E) darstellt. Mit der Worst Case Coverage lässt sich also abbilden, wie viele Knoten nötig sind, um komplette Funktionalität zu erreichen, selbst wenn eine ungünstige Anordnung aktiver Knoten gewählt wird. Um Aussagen mittels Worst Case Coverage zu generieren, ist zusätzlicher Simulationsaufwand nötig. Gleichzeitig müssen für eine gute Abdeckung mehr Knoten simuliert werden, was zusätzlichen Zeitaufwand bedeutet. Daher wurde die Worst Case Coverage wurde im Zuge dieser Arbeit als Evaluierungskriterium hauptsächlich dort eingesetzt, wo die Realistic Case Coverage keine signifikanten Unterschiede bei den einzelnen Algorithmen ergab. Genau wie für die Realistic Case Coverage wurde auch für die Worst Case Coverage der Rasterabstand auf 1 m gesetzt.

4.2.3 Node efficiency

Um die Effizienz eines Algorithmus' zu bestimmen, ist es nicht nur notwendig, die minimale Anzahl platzierter Knoten zu kennen, sondern auch, welcher Anteil dieser Knoten aktiv ist, und welcher Anteil am Netzwerk mit den aktiven Knoten abgedeckt wird. Um dies bewerten zu können, wurde in [61] der Be-



Abbildung 4.4 Knotenanordnung zum Erhalt der maximalen Node Efficiency

griff der *Node efficiency* (kurz: *NE*) eingeführt, welcher sich wie in Gleichung (4.3) beschrieben ermittelt.

$$NE = \frac{A_{akt}}{N_{akt} * \pi * R_S^2} \tag{4.3}$$

Mit A_{akt} wird dabei die mittels Sensoren aktiver Knoten erfasste Fläche bezeichnet, N_{akt} bezeichnet die Anzahl aktiver Knoten und R_S die Sensorreichweite. Die *NE* beschreibt dementsprechend die tatsächlich abgedeckte Fläche im Verhältnis zur maximal möglichen abgedeckten Fläche.

Je höher der *NE*-Wert ist, desto effizienter werden die Sensorknoten im entsprechenden Algorithmus ausgenutzt. Der *NE*-Wert ist jedoch begrenzt und kann selbst unter idealen Bedingungen für räumlich unbegrenzte Netzwerke den Maximalwert *NE_{max}* nicht überschreiten. *NE_{max}* entsteht gemäß [61] genau dann, wenn die aktiven Knoten regelmäßig so angeordnet sind, dass benachbarte Knoten die Ecken von gleichseitigen Dreiecken mit einem Abstand von d_{NB} = $\sqrt{3} * R_S$ bilden, wie Abbildung 4.4 zeigt. Gemäß Gleichung (4.4) ergibt sich damit eine maximale Node efficiency von *NE_{max}* ≈ 82,7%.

$$NE_{max} = \frac{6(R_s^2 * \sqrt{3}/4)A_{akt}}{\pi * R_s^2} = \frac{3 * \sqrt{3}}{2\pi} = 0,827$$
(4.4)

4.2.4 Lebenszeit

In der Literatur werden häufig Methoden zu Bewertung der Lebenszeit eines drahtlosen Sensornetzwerks vorgestellt, die die Lebenszeit mit einem bestimmten Anteil verbleibender Knoten gleichsetzen. Jedoch ist diese Aussage äußerst ungeeignet, da sie nicht auf die erwünschte Funktionalität eingeht. Befinden sich beispielsweise alle Knoten permanent im Schlafmodus, hätte das Sensornetzwerk seine maximale Lebenszeit erreicht, ohne jedoch in der Lage zu sein, auch nur ein einziges Phänomen zu detektieren.

Um einen entsprechend fairen Vergleich der Performance verschiedener Algorithmen zu gewährleisten, wurde in der vorliegenden Arbeit die Lebenszeit als



Abbildung 4.5 Wahrscheinlichkeit kompletter Netzwerkfunktionalität zur Laufzeit des Sensornetzwerks (Lebenszeit) für zwei Beispielalgorithmen

die Zeit definiert, nach der das Netzwerk nicht mehr in der Lage ist, seine Aufgabe zu erfüllen. Kommt es demnach zu einem Loch in der Sensorabdeckung oder sind Knoten nicht mehr in der Lage, die Senke zu erreichen, gilt das Netzwerk zu diesem Zeitpunkt als nicht mehr als lebend.

Mittels dieser Definition ist es möglich, die entwickelten Clusteringalgorithmen fair zu vergleichen, ohne beispielsweise Algorithmen zu bestrafen, welche zusätzlich Knoten aktivieren, um eine Heilung ausgefallener Knoten zu ermöglichen.

Basiert ein Clusteringalgorithmus auf ungenauen Positionsdaten oder ist schon ein Großteil der Knoten ausgefallen, kann es zu dem Fall kommen, dass nur ein Teil der simulierten Sensornetzwerke in der Lage ist, komplette Netzwerkfunktionalität zu erreichen. In diesem Fall wird nur der Anteil der Gesamtlebenszeit des Netzwerks hinzugefügt, der dem Anteil der Sensornetzwerke entspricht, die komplette Netzwerkfunktionalität erreichen. Um dies zu verdeutlichen, vergleicht Abbildung 4.5 die Lebenszeiten zweier Algorithmen. In dem durch Algorithmus *B* geclusterten Netzwerk erreichen bei jedem Messpunkt nur 80% der simulierten Sensornetzwerke komplette Netzwerkfunktionalität, so dass eine Gesamtlebenszeit von $T_{life} = 20,7T$ entsteht. Somit entsteht bei Algorithmus *B* trotz längerer Laufzeit insgesamt eine kürzere Lebenszeit als bei Algorithmus *A*, welche 21,4T beträgt.

4.3 Kurzzusammenfassung

Das vorliegende Kapitel hatte die Aufgabe, die in der Arbeit verwendete realitätsnahe Bewertung von Clusteringalgorithmen vorzustellen. Dazu wird zunächst in Kapitel 4.1 über den Netzwerksimulator und die verwendeten vereinfachten Annahmen und Modelle diskutiert. Kapitel 4.2 beschreibt danach Methoden, um die Effizienz eines Clusteringalgorithmus' zu bewerten.

Mittels des verwendeten Netzwerksimulators und der Bewertungsmethoden ist es möglich, die im Verlaufe dieser Arbeit entwickelten Algorithmen im Simulator sinnvoll zu erproben und untereinander sowie mit bereits vorhandenen Ansätzen zu vergleichen.

5 Lokalisierung drahtloser Sensorknoten

Innerhalb der in Kapitel 2.2.4 beschriebenen Szenarien ist es nicht nur entscheidend, herauszufinden, dass und wann ein Phänomen aufgetreten ist, sondern auch, dessen Position und räumliche Ausdehnung im entsprechenden Bezugssystem zu kennen. Zusätzlich bietet das Wissen über die geographische Nähe von Sensorknoten die Möglichkeit, redundante Knoten zu erkennen. So wird die erkannte Position der Knoten beispielsweise durch die in Kapitel 6 entwickelten Clusteringverfahren ausgenutzt.

Um eine Lokalisierung durchzuführen, muss jeder messende Sensorknoten möglichst genau seine Position in einem Bezugssystem kennen. Da die Knoten häufig in unzugänglichen Gebieten oder in großer Zahl ausgebracht werden sollen, kann nicht davon ausgegangen werden, dass die Knoten per Hand platziert und ihnen ihre Position einprogrammiert wird. Im Bereich der Outdoorexistiert erdumspannend das satellitengestützte Global Lokalisierung Positioning System [62] (GPS), welches in der Lage ist, entsprechende Empfangsgeräte mit einer Genauigkeit von 5 m bis 10 m zu lokalisieren. Jedoch benötigen GPS-Empfänger zum einen eine nicht unerhebliche Energiemenge, zum anderen ist eine derart genaue Lokalisierung nur mittels direkter Sichtlinie zum Satelliten möglich. In urbanen Gebieten beispielsweise ist nicht auszuschließen, dass für einige Knoten mit extremen Ungenauigkeiten zu rechnen ist. Um eine GPS-unabhängige Lokalisierung durchzuführen, wird in der Literatur häufig vorgeschlagen, verhältnismäßig wenige Knoten als Referenz zu verwenden und deren Position im Bezugssystem mit geeigneten, gegebenenfalls aufwändigen Methoden im Vorfeld des Sensornetzbetriebs zu bestimmen [63], [64], [65]. Diese Knoten mit bekannter Position, nachfolgend als Beacons (engl. Leuchtfeuer) bezeichnet, senden zur Lokalisierung der anderen unbekannten Knoten eine Broadcast Nachricht aus, die die Positionsdaten des entsprechenden Beacons erhalten.

Die Knoten mit unbekannter Position, nachfolgend als *Unbekannte* bezeichnet, bestimmen ihre eigene Position mit Hilfe der empfangenen Beaconbroadcasts. Die dazu verwendeten Verfahren unterscheiden sich durch den Rechenaufwand und die erzielte Genauigkeit. In der vorliegenden Arbeit wird anhand der maximal erreichbaren Genauigkeit zwischen heuristischen und exakten Verfahren unterschieden.

Im Zuge dieser Arbeit wurden in Kapitel 5.1 drei heuristische Verfahren auf ihre Genauigkeit unter idealen und realitätsnahen Bedingungen in einem zweidimensionalen Bezugssystem untersucht. Desweiteren wurde die Genauigkeit der Multilateration als exaktes Verfahren in Kapitel 5.2 untersucht. Auf Basis dieser Untersuchungen wurde in Kapitel 5.3 ein hybrides Verfahren entwickelt, welches die erzielte Genauigkeit erhöht und die für die exakten Verfahren typischen Ausreißer minimiert.

5.1 Heuristische Verfahren

Als heuristische Verfahren gelten Methoden, die mit begrenzten Informationen und Aufwand zu geeigneten Lösungen kommen. Bezugnehmend auf die Lokalisierung werden in der vorliegenden Arbeit die Algorithmen als heuristische Verfahren bezeichnet, die zwar einfache, effiziente Regeln verwenden, aber auch mit hinreichend genauen Distanzschätzungen keine genaue Position eines zu schätzenden Knotens ermitteln können. Im Folgenden werden drei bekannte heuristische Verfahren beschrieben und im Hinblick auf die erreichbare Genauigkeit unter realitätsnahen Bedingungen untersucht.

5.1.1 Schwerpunktbestimmung

Die Schwerpunktbestimmung (engl. Centroid Localization, kurz: CL) ermöglicht es, die Position eines Knotens auf einfache Art und Weise zu ermitteln [63]. Um einen unbekannten Knoten mit unbekannter Position P mit CL zu lokalisieren, genügt es, die Positionsdaten B(x,y) jeder empfangenen Beaconnachricht *i* zu speichern. Der Unbekannte ermittelt daraufhin seine Position als Schwerpunkt aus den Positionen aller empfangenen Nachrichten *n* der benachbarten Beacons mittels Gleichung (5.1).

$$P(x, y) = \frac{1}{n} \sum_{i=1}^{n} B_i(x, y)$$
(5.1)

Der Vorteil des Algorithmus' ist die einfache Berechnung der Position. Zudem wird keine Signalstärke oder andere Form der Abstandsschätzung zu den empfangenen Beacons vorgenommen, was den Algorithmus unempfindlich gegen Fehler in der Abstandsschätzung macht und ihn auch einsetzbar macht, wenn eine Signalstärkemessung und damit Distanzschätzung nicht oder nur schwer möglich ist. Der Nachteil des Verfahrens ist der systematisch hohe Fehler, der durch die einfache Heuristik entsteht. Dieser ist besonders hoch, sobald sich ein Unbekannter außerhalb des von den Beacons aufgespannten Polygons befindet.

In [66] wurde der CL-Algorithmus für ein Szenario mit regelmäßig angeordneten Beacons optimiert. Dennoch entsteht selbst unter den in Kapitel 4.1.6 vorgestellten idealen Bedingungen ein Durchschnittsfehler der Lokalisierung mit CL von 17,5% der Sendereichweite.

5.1.2 Gewichtete Schwerpunktbestimmung

Die gewichtete Schwerpunktbestimmung (engl. Weighted Centroid Localization, kurz: WCL) wurde in [64] entwickelt, um das Ergebnis der Schwerpunktsbestimmung zu verbessern. WCL verwendet dazu einen Wichtungsfaktor w, der einem Knoten mit größerem Abstand zum Unbekannten eine geringere Bedeutung bei der Schwerpunktbestimmung zuweist, wie in Gleichung (5.2) für einen Unbekannten mit Position P und n empfangene Beacons zu erkennen ist.

$$P(x,y) = \frac{\sum_{i=1}^{n} (w_i * B_i(x,y))}{\sum_{i=1}^{n} w_i}$$
(5.2)

Ein geeigneter Wichtungsfaktor ist entscheidend für die Qualität von WCL. In der Literatur wird für die Wichtung beispielsweise auf die Empfangssignalstärke oder einen Link Quality Index zurückgegriffen [64]. Im Zuge der Untersuchungen der vorliegenden Arbeit wurde auf einen Wichtungsfaktor zurückgegriffen, der auf dem geschätzten Abstand zu dem entsprechenden Beacon *i* basiert, wie in Gleichung (5.3) gezeigt.

$$w_i = (d_i)^{-1} (5.3)$$

Der Abstand wird dazu, wie später beschrieben, aus der empfangenen Signalstärke geschätzt.

Zwar verliert WCL aufgrund des Einbeziehens der Empfangssignalstärke die Unempfindlichkeit gegenüber dieser, jedoch ist ein deutlicher Anstieg in der Lokalsierungsgenauigkeit gegenüber CL zu verzeichnen, wie Kapitel 5.1.4 zeigt.

5.1.3 Adaptive gewichtete Schwerpunktbestimmung

Die adaptive, gewichtete Schwerpunktbestimmung (engl. Adaptive Weighted Centroid Localization, kurz: AWCL) wurde in [67] entwickelt, um die Lokalisierung mittels WCL noch weiter zu verbessern.

Ist der Abstand vom Unbekannten zu jedem Beacon annähernd gleich groß, sinkt der Einfluss der zusätzlichen Wichtung und die Performance des WCL-Algorithmus' nähert sich CL an.

Grundidee von AWCL ist es daher, den Unterschied in den Wichtungen deutlicher in die Formel der Schwerpunktberechnung einfließen zu lassen. Daher werden die errechneten Gewichte für jeden Beacon wie folgt angepasst:

1.) Bestimmung des kleinsten Gewichts aus allen Beacons, wmin

2.) Berechnung eines Wertes für die Verringerung von allen Gewichte, der "reduction part" *q*. Simulationsergebnisse in [67] zeigen, dass ein günstiger Wert für *q* sich wie in Gleichung (5.4) beschrieben errechnet.

$$q = w_{min} * 0,55 \tag{5.4}$$

3.) Schwerpunktbestimmung mit angepassten Gewichten, wie in Gleichung (5.5) gezeigt.

$$P(x,y) = \frac{\sum_{i=1}^{n} ((w_i - q) * B_i(x, y))}{\sum_{i=1}^{n} (w_i - q)}$$
(5.5)

Im Ergebnis gelingt es AWCL, eine differenzierte Gewichtung der Beaconabstände vom Unbekannten in die Schwerpunktbestimmung einfließen zu lassen und somit in vielen Fällen ein genaueres Lokalisierungsergebnis zu erzielen.

5.1.4 Vergleich der Algorithmen unter realitätsnahen Bedingungen

Für den Vergleich der heuristischen Lokalisierungsalgorithmen wurde zunächst



Abbildung 5.1 Vergleich der heuristischen Lokalisierungsverfahren, (A) Unbekannter Knoten außerhalb der von den Beacons aufgespannten Fläche, (B) Unbekannter Knoten innerhalb der von den Beacons aufgespannten Fläche, (C) Ergebnis mit 20% Fehler einer Distanzschätzung
untersucht, unter welchen Parametern besonders große bzw. geringe Lokalisierungsfehler entstehen. Dabei stellte sich heraus, dass es von großer Bedeutung für die erzielte Genauigkeit ist, wo sich der Knoten im Verhältnis zu den von ihm empfangenen Beacons befindet.

Eine Eigenschaft der vorgestellten heuristischen Verfahren ist es, dass sich ein Knoten mittels dieser immer eine Position errechnet, die in der Fläche liegt, die von den für die Berechnung der Position verwendeten Beacons aufgespannt wird. Somit errechnet sich ein Knoten, welcher sich außerhalb dieser Fläche befindet, eine Position mit entsprechend großem Fehler, wie Abbildung 5.1(A) zeigt. Da sich ein Knoten mittels CL immer eine Position direkt im Schwerpunkt dieser Fläche errechnet, ist der Fehler im Vergleich zu WCL und AWCL entsprechend höher.

Befindet sich der Knoten innerhalb der aufgespannten Fläche, wie in Abbildung 5.1(B) gezeigt, so kann die Position mittels der gewichteten Verfahren relativ genau bestimmt werden. Fehler bei der Distanzschätzung, wie sie unter realistischen Bedingungen auftreten, wirken sich nur gering, oder im Falle von CL, gar nicht auf den erzielten Lokalisierungsfehler aus, solange noch alle Beaconnachrichten empfangen werden. Dies wird in Abbildung 5.1(C) gezeigt. Für den eigentlichen Vergleich der Algorithmen wurden Beacons und Unbekannte im Simulator PROWLER gleichverteilt in einem Gebiet platziert. Dabei wurden unterschiedliche Beacondichten so eingestellt, dass jede Lokalisierung mit einer bestimmten Anzahl empfangener Beaconnachrichten mindestens 1000 mal durchgeführt wurde, um aussagekräftige Ergebnisse zu erhalten.

Des weiteren wurden die Simulationen in den in Kapitel 4.1.6 beschriebenen realitätsnahen Szenarien durchgeführt. Um Randeffekte zu vermeiden, wurde nur die Positionsermittlung von Unbekannten evaluiert, die mehr als 100m von den Kanten des simulierten Gebiets entfernt lagen. Um die für AWCL und WCL benötigten Gewichte zu bestimmen, wurde dazu die Entfernung aus der



Abbildung 5.2 Arithmetisches Mittel der Ergebnisse einer realitätsnahen Lokalisierung mittels heuristischer Verfahren in verschiedenen Szenarien, (A) Flacher Sandstrand, (B) Trockenes Unterholz



Abbildung 5.3 Boxplotdarstellung der Ergebnisse des AWCL-Verfahrens in verschiedenen Szenarien, (A) Flacher Sandstrand, (B) Trockenes Unterholz

gemessenen Empfangssignalstärke mittels Gleichung (5.6) bestimmt.

$$d[m] = \sqrt[n]{\frac{P_T}{P_R} \left(\frac{c}{4\pi f * 1m}\right)^2}$$
(5.6)

Dabei geht man davon aus, dass an einem empfangenden Sensorknoten alle benötigten Werte, also sowohl Pfadverlustexponent n, als auch Sende- und Empfangsleistung P_T und P_R , sowie die gesendete Frequenz f vorliegen.

Aufgrund dieser Einstellungen kann es für jeden Knoten sowohl zu einer in Abbildung 5.1(A) als auch zu einer in Abbildung 5.1(B) dargestellten Anordnung von Beacons und Unbekanntem kommen.

Anhand des arithmetischen Mittelwertes wurde die erzielte Genauigkeit in Abbildung 5.2 der einzelnen Algorithmen miteinander verglichen. Dabei ist zu erkennen, dass die beiden gewichteten Verfahren dem einfachen CL überlegen sind. AWCL erzielt dabei insgesamt die beste Performance. In Abbildung 5.3 sind dieselben Ergebnisse für AWCL in Form eines Boxplot-Diagramms für beide untersuchten Szenarien dargestellt. Anhand der beiden Diagramme ist zu erkennen, dass der maximale Fehler circa das Vierfache des Durchschnittsfehlers beträgt. Zudem ist zu erkennen, dass sich in beiden Szenarien die erzielten Lokalisierungsgenauigkeiten nur wenig voneinander unterscheiden.

5.2 Exakte Verfahren

Als exaktes Verfahren wird in der vorliegenden Arbeit eine Methode zur Lokalisierung bezeichnet, die mit exakten Messwerten auch ein exaktes Lokalisierungsergebnis ermöglicht. Tri- und Multilateration, wie sie in der Literatur [65] beschrieben werden, zählen dabei zu den am weitesten verbreiteten Methoden und werden daher im Folgenden analysiert. Zusätzlich werden verschiedene Methoden der Multilateration vorgestellt und es wird untersucht, welche Faktoren die Ergebnisse der Multilateration beeinflussen.

5.2.1 Trilateration

Die Trilateration als exaktes Verfahren zur Lokalisierung verwendet die Beaconinformationen ,Position des Beacons' und ,Abstand zum Unbekannten', um die exakte Position des Unbekannten zu bestimmen. Veranschaulicht dargestellt ergibt sich die Position des Unbekannten $u(x_{ub}y_u)$ durch das Ermitteln der Parameter einer linearen Funktion mit dem Anstieg x_u und dem Schnittpunkt mit der h-Achse y_u in einem Bezugssystem (g,h), wie in Gleichung (5.7) gezeigt.

$$h = x_u * g + y_u \tag{5.7}$$

Um die Parameter dieser linearen Gleichung zu ermitteln, sind mindestens 2 Wertepaare g,h notwendig. Der x-Wert des Unbekannten ergibt sich daraufhin als Anstieg der linearen Funktion, wie in Gleichung (5.8) beschrieben.

$$x_u = \frac{h_2 - h_1}{g_2 - g_1} \tag{5.8}$$

Der *y*-Wert ergibt sich daraufhin als Schnittpunkt mit der *h*-Achse, wie in Gleichung (5.9) gezeigt.

$$y_u = h_1 - x_u * g_1 \tag{5.9}$$

Um auf die Werte für g und h zu kommen, werden die Gleichungen, die sich aus den Positionsinformationen der Beacons und der Distanzinformation zu diesen ergeben, wie folgt umgestellt. Mittels eines Beacons *i*, eines Unbekannten *u* und der ermittelten Distanz $d_{i,u}$ lässt sich nach Pythagoras folgende Gleichung (5.10) für einen Unbekannten *u* aufstellen.

$$(x_i - x_u)^2 + (y_i - y_u)^2 = d_{i,u}^2$$
(5.10)

Wird die Gleichung wie in Gleichung (5.11) umgestellt, ist es möglich, mittels eines zweiten Beacons *j* die Unbekannten Quadratterme x_u^2 und y_u^2 zu eliminieren und somit eine lineare Funktion zu erhalten, siehe Gleichung (5.12).

$$x_u^2 + y_u^2 = d_{i,u}^2 + 2x_u x_i + 2y_u y_i - x_i^2 - y_i^2$$
(5.11)

$$d_{j,u}^{2} + 2x_{u}x_{j} + 2y_{u}y_{j} - x_{j}^{2} - y_{j}^{2} = d_{i,u}^{2} + 2x_{u}x_{i} + 2y_{u}y_{i} - x_{i}^{2} - y_{i}^{2}$$
(5.12)



Abbildung 5.4 Trilateration eines Unbekannten, (A) Graphische Darstellung der Knotenpositionen, (B) Versuchsaufbau und Ergebnisse, (C) resultierende Funktion im (g,h)-Raum

Um die resultierende lineare Funktion in die Form der Gleichung (5.7) zu bringen, muss sie wie in Gleichung (5.13) dargestellt umgestellt werden.

$$\frac{d_{i,u}^2 - d_{j,u}^2 - x_i^2 - y_i^2 + x_j^2 + y_j^2}{2} = x_u(x_j - x_i) + y_u(y_j - y_i)$$
(5.13)

Im speziellen Fall $y_i = y_j$ kann mittels Gleichung (5.13) direkt der Wert für x_u berechnet werden. Für alle anderen Fälle wird der Term $y_i = y_j$ dividiert, sodass sich als Werte für g und h die folgenden Gleichungen (5.14) und (5.15) ergeben.

$$g_{i,j} = \frac{(x_j - x_i)}{(y_j - y_i)}$$
(5.14)

$$h_{i,j} = \frac{d_{i,u}^2 - d_{j,u}^2 - x_i^2 - y_i^2 + x_j^2 + y_j^2}{2(y_i - y_i)}$$
(5.15)

Man erhält also für jedes Beaconpaar einen Punkt im (g,h)-Bezugsraum. Ein Beispiel einer Trilateration ist in Abbildung 5.4 dargestellt. Dort ist deutlich zu erkennen, dass mittels exakt bestimmter Distanzen auch die exakte Position eines Unbekannten ermittelt werden kann.

5.2.2 Multilateration und Methode der kleinsten Fehlerquadrate

Die Multilateration mit anschließender Anwendung der Methode der kleinsten Fehlerquadrate, auch als "atomic multilateration" in [65] bezeichnet, bezeichnet eine Methode zur Schätzung einer Lösung eines unbestimmten linearen Gleichungssystems, wie es bei der Multilateration vorliegt. Angewendet auf das Lokalisierungsproblem, hat ein unbekannter Knoten möglichst sinnvolle Lösungen für seine unbekannten Koordinaten x und y mit Hilfe der von den Beacons erhaltenen Informationen zu finden. Bezogen auf den bereits vorgestellten (g,h)-Bezugsraum erhält man durch die Verwendung von mehr Beacons mehrere Punkte in diesem Raum. Da diese nur im Fall exakter Distanzmessungen direkt auf einer Linie liegen, müssen die Werte für x und y anhand eines Verfahrens bestimmt werden. Die Methode der kleinsten Fehlerquadrate entspricht in diesem Fall einer linearen Regression der Punkte im (g,h)-Bezugsraum.

Die Berechnung von x_u und y_u bezieht daher im Gegensatz zu den Gleichungen (5.8) und (5.9) mehr als 2 Punkte im (g,h)-Raum ein. Um zu einer Lösung für x_u und y_u zu kommen, werden bei diesem Verfahren im ersten Schritt die Durchschnittswerte \overline{g} und \overline{h} aus den n vorhandenen Punkten im (g,h)-Raum wie in Gleichung (5.16) beschrieben berechnet.

$$\bar{g} = \frac{\sum_{m=1}^{n} g_m}{n}; \ \bar{h} = \frac{\sum_{m=1}^{n} h_m}{n}$$
 (5.16)

Mittels dieser Durchschnittsgrößen werden dann gemäß einer linearen Regression die finalen Werte für x_u und y_u bestimmt, wie in den Gleichungen (5.17) und (5.18) beschrieben.

$$x_u = \frac{\sum_{m=1}^n (g_m - \bar{g}) (h_m - \bar{h})}{\sum_{l=1}^n (g_m - \bar{g})^2}$$
(5.17)

$$y_u = h - x_u * \bar{g} \tag{5.18}$$

Mit diesen Gleichungen ist es nun möglich, eine exakte Positionsbestimmung für mehr als drei empfangene Beacons anzuwenden. Unter idealen Bedingungen errechnet auch bei der Multilateration ein Knoten seine exakte Position.

5.2.3 Multilateration unter realitätsnahen Bedingungen

Werden Beacons und Knoten gleichverteilt in einem zu überwachenden Gebiet platziert, kommt es zu sämtlichen möglichen Kombinationen von Beaconanordnungen und Unbekannten. Zusätzlich entstehen aufgrund der unterschiedlichen Kanaleigenschaften zwischen den Beacons und einem Unbekannten auch Fehler in der Distanzschätzung. Ähnlich wie bei den heuristischen Verfahren wurde auch bei den exakten Verfahren untersucht, welche Parameter besonders großen bzw. kleinen Einfluss auf die erzielte Lokalisierungsgenauigkeit besitzen. Im Gegensatz zu den analysierten heuristischen Verfahren spielt es bei der Tri- oder Multilateration nur eine geringe Rolle, ob sich der Unbekannte innerhalb der von den Beacons aufgespannten Fläche befindet.



Abbildung 5.5 Trilateration eines Unbekannten mit 20% Distanzschätzfehler zu Beacon I, (A) Graphische Darstellung der Knotenpositionen, (B) Versuchsaufbau und Ergebnisse, (C) resultierende Funktion im (g,h)-Raum

In Abbildung 5.5 wurde derselbe Unbekannte wie in Abbildung 5.4 lokalisiert, und ein Fehler von 20% bei der in der Distanzschätzung zu einem Beacon angenommen. Obwohl der Unbekannte außerhalb der aufgespannten Fläche liegt, ist bei der gewählten Anordnung von Beacons und Unbekannten in diesem Fall nur ein geringer Lokalisierungsfehler zu verzeichnen.

Wird jedoch die Anordnung der Beacons so verändert wie in Abbildung 5.6 (A) dargestellt, wird die durch die Beacons aufgespannte Fläche deutlich geringer. Daraus ergeben sich Punkte im (g,h)-Raum, die relativ dicht beieinander liegen. In solch einem Fall kann auch schon ein geringer Fehler in der Distanzschätzung, der zu einer Veränderung eines *h*-Werts führt, zu einer gravierenden Veränderung des Anstiegs und des Schnittpunkts mit der *h*-Achse führen, wie in Abbildung 5.6 (C) dargestellt. Eine ungünstige Kombination von Beaconanordnung und Distanzschätzung kann also die Lokalisierung mittels Tri- oder Multilateration erheblich beeinflussen.

Da die Distanzschätzung mittels Gleichung (5.6) aus der gemessenen Emp-



Abbildung 5.6 Trilateration eines Unbekannten mit 20% Distanzschätzfehler zu Beacon I und ungünstiger Beaconanordnung, (A) Graphische Darstellung der Knotenpositionen, (B) Versuchsaufbau und Ergebnisse, (C) resultierende Funktion im (g,h)-Raum



Abbildung 5.7 Boxplotdarstellung der Ergebnisse einer Multilateration in verschiedenen Szenarien, (A) Flacher Sandstrand (B) Trockenes Unterholz

fangssignalstärke ermittelt wird, entstehen durch die Schwankung der empfangenen Signalstärke natürlich mehr oder minder große Fehler bei der Distanzschätzung, sodass die Anordnung der Beacons in Sendereichweite entscheidend für das Lokalisierungsergebnis ist.

Um die Ergebnisse der Multilateration mit denen der heuristischen Verfahren zu vergleichen, wurden in der Simulation dieselbe Simulationsumgebung wie in Kapitel 5.1.4 sowie dieselben Szenarien gewählt.

In Abbildung 5.7 ist der erzielte Lokalisierungsfehler einer Multilateration für beide untersuchten Szenarien in Boxplotdarstellung zu sehen. Während sich der Median nur unwesentlich von den heuristischen Verfahren unterscheidet, ist der sich nach oben streckende Whisker extrem hoch. Bei einer Kombination von ungünstiger Beaconanordnung und ungünstiger Abstandsschätzung können also Fehler bei der Lokalisierung auftreten, die einem Vielfachen der durchschnittlichen Sendereichweite entsprechen.

Zusammenfassend lässt sich aus Abbildung 5.7 ableiten, dass die Lokalisierung mittels exakter Multilateration im Vergleich zur heuristischen Lokalisierung unter ungünstigen Bedingungen Fehler generiert, die so hoch sind, dass jeglicher Bezug zur Realität verloren geht. Diese Fehlerquelle betrifft dabei jede der im Zuge dieser Arbeit untersuchten Varianten der Multilateration. Im folgenden Kapitel wird der Einfluss dieser Fehlerquelle durch einen hybriden Lokalisierungsansatz minimiert.

5.3 HyPAERLoc

Um die extremen Ausreißer der exakten Verfahren unter ungünstigen Abstandsschätzungen und Beaconanordnungen zu vermeiden und zusätzlich eine Genauigkeitssteigerung bei der Lokalisierung zu erreichen, wurde im Zuge dieser Arbeit das hybride Verfahren HyPAERLoc (Hybrid Plausible Approach for Error Reduced Localization) entwickelt, welches das Potential der heuristischen Verfahren, also die Robustheit gegen Ausreißer, mit den potentiell genaueren Lokalisierungsergebnissen der exakten Verfahren kombiniert [68]. Das Verfahren kann sowohl als eigenständige Lokalisierung als auch als Basis für eine in [65] beschriebene iterative Lokalisierung dienen, bei welcher bereits lokalisierte Knoten als Beacons dienen.

Kapitel 5.3.1 und Kapitel 5.3.2 widmen sich dabei der Analyse und Optimierung der Multilateration, während Kapitel 5.3.3 den hybriden Ansatz beschreibt, welcher im HyPAERLoc Algorithmus verwendet wird.

5.3.1 Geeignete Auswahl der Beaconpaare

In den vorhergehenden Kapiteln 5.2.1 und 5.2.2 wurde beschrieben, dass aus zwei Beacons jeweils ein Wertepaar für den (g,h)-Bezugsraum gebildet wird. Eigene Untersuchungen zeigten jedoch, dass die Auswahl der Beaconpaare nicht unbedeutend für die erzielte Genauigkeit ist. Folgende mögliche Strategien zur Wahl geeigneter Beaconpaare wurden daher im Rahmen einer Vorab-Untersuchung für die evaluierten Szenarien untersucht.

Maximale Punkte im (g,h)-Raum (engl. Maximum Point Linearization, kurz: MPL)

Bei dieser Linearisierung wird aus jeder möglichen Beaconkombination ein Wertepaar im (g,h)-Raum gebildet. Da dieses mathematisch einer Kombination ohne Wiederholung entspricht, werden aus *m* Beacons somit die in Gleichung (5.19) beschriebene Anzahl Punkte n_{MPL} im (g,h)-Raum erzeugt.

$$n_{MPL} = \frac{m(m-1)}{2}$$
(5.19)

Nachfolgende Beacons bilden Paare (engl. Sliding Window Linearization, kurz: SWL)

Aufgrund des quadratischen Aufwands von MPL bietet es sich an, über Alternativen nachzudenken. SWL bietet eine günstige Alternative. Hierbei werden Beaconpaare aus jeweils aufeinanderfolgend empfangenen Beacons gebildet. Zusätzlich wird der letzte empfangene Beacon dann noch mit dem zuerst empfangenen Beacon zu einem Beaconpaar verknüpft. Das Verfahren hat den Vorteil eines deutlich reduzierten Aufwands. Anstatt das eine Beaconinformation in m-1 Punkte einfließt, wird jede Beaconinformation nur noch genau in zwei Punkten im (g,h)-Raum verwendet. Für m empfangene Beacons ergeben sich daher n=m Punkte im (g,h)-Raum.



Abbildung 5.8 Arithmetisches Mittel der Ergebnisse einer realitätsnahen Lokalisierung mittels Multilateration mit verschiedenen Beaconpaarstrategien in den Szenarien, (A) Flacher Sandstrand, (B) Trockenes Unterholz

Dichtester Beacon als Linearisierer (engl. Next Beacon as Linearizer, kurz: NBL)

Die Verwendung des am nahesten detektierten Beacons als Linearisierer wird bereits in dem verteilten Lokalisierungsverfahren sDLS (scalable Distributed Least Squares) in [69] eingesetzt. Bei diesem Verfahren werden alle empfangenen Beaconnachrichten nach dem zum Unbekannten geschätzten Abstand sortiert. Der Knoten mit dem kürzesten geschätzten Abstand wird daraufhin als Linearisierer verwendet und bildet mit jedem anderen Knoten ein Beaconpaar. Aus *m* empfangenen Knoten werden somit n=m-1 Knoten im (g,h)-Raum gebildet.

Der arithmetische Mittelwert des Lokalisierungsfehlers im Verhältnis zu den empfangenen Beaconnachrichten ist in Abbildung 5.8 dargestellt. Deutlich ist zu erkennen, dass das NBL-Verfahren die besten Resultate erzielt, und somit am geeignetsten scheint. Grund dafür ist, dass eine Signalstärkeschwankung bei einer geringeren Entfernung zu einem geringeren Fehler bei der Abstandsschätzung führt. Der relativ genau ermittelbare Abstand zum Linearisierer, welcher in jede Gleichung zur Bestimmung der Position des Unbekannten eingeht, bekommt also erhebliche Bedeutung bei der Bestimmung der Knotenposition. Die in Kapitel 5.2.3 beschriebenen Ausreißer durch Verwendung der Multilateration lassen sich jedoch durch das NBL-Verfahren nicht reduzieren.

5.3.2 Optimierung nicht plausibler Distanzschätzungen

Für die Genauigkeit der Multilateration hat eine genaue Distanzschätzung eine erhebliche Bedeutung. Daher erscheint es lohnenswert, die mittels der recht ungenauen Methode der Distanzschätzung auf RSSI-Basis errechneten Distanzen auf Plausibilität zu überprüfen. Im Zuge dieser Arbeit wurde daher ein



Abbildung 5.9 Erkennung nicht plausibler Distanzen (A) Gap₁: Geschätzte Distanzen kleiner als Beaconabstand, (B) und (C) Gap₂ und Gap₃: Beaconabstand und eine Distanzschätzung kleiner der zweiten Distanzschätzung

Verfahren entwickelt und analysiert, mit welchem ein Knoten einige nicht plausible Distanzen erkennen und das es ihm erlaubt, die geschätzten Distanzen entsprechend abzuändern. Das Verfahren besteht aus zwei Schritten, der Erkennung und der Anpassung nicht plausibler Distanzen, die im Folgenden beschrieben werden.

Erkennung nicht plausibler Distanzen

Um nicht plausible Distanzen zu ermitteln, ermittelt ein Unbekannter U die Abstände zu zwei empfangenen Beacons I und J, woraus die geschätzten Distanzen $d_{i,u}$ und $d_{j,u}$ entstehen. Aus den mitgesendeten Positionen der Beacons kann sich der Unbekannte den Abstand der Beacons zueinander errechnen, hier als $d_{i,j}$ bezeichnet. Eine nicht plausible Lücke, hier als *Gap* bezeichnet, entsteht immer dann, wenn zwei dieser Distanzen zusammenaddiert kleiner sind als die dritte Distanz. Geometrisch betrachtet bedeutet dies, dass selbst wenn der Knoten auf einer Geraden mit den Beacons läge, zumindest eine der Abstandsschätzungen falsch ist.

BEDINGUNG	ANPASSUNG VON d _{i,u}	ANPASSUNG VON $d_{j,u}$
$Gap_1 > 0$	$d_{i,u}' = d_{i,u} \left(1 + \frac{Gap_1}{d_{i,u} + d_{j,u}} \right)$	$d_{j,u}' = d_{j,u} \left(1 + \frac{Gap_1}{d_{i,u} + d_{j,u}} \right)$
$Gap_2 > 0$	$d_{i,u}' = d_{i,u} \left(1 - \frac{Gap_2}{d_{i,u} + d_{j,u}} \right)$	$d_{j,u}' = d_{j,u} \left(1 + \frac{Gap_2}{d_{i,u} + d_{j,u}} \right)$
$Gap_3 > 0$	$d_{i,u}' = d_{i,u} \left(1 + \frac{Gap_3}{d_{i,u} + d_{j,u}} \right)$	$d_{j,u}' = d_{j,u} \left(1 - \frac{Gap_3}{d_{i,u} + d_{j,u}} \right)$

TABELLE 5.1 ANPASSUNG DER NICHT PLAUSIBLEN DISTANZSSCHÄTZUNGEN

Es gibt drei Arten von Gaps, die bei einem Beaconpaar und einem Unbekannten entstehen können. Diese werden durch Gleichung (5.20), (5.21) und (5.22) beschrieben und in Abbildung 5.9 graphisch dargestellt sind.

$$Gap_1 = d_{i,j} - (d_{i,u} + d_{j,u})$$
(5.20)

$$Gap_2 = d_{i,u} - (d_{i,j} + d_{j,u})$$
(5.21)

$$Gap_3 = d_{j,u} - (d_{i,j} + d_{i,u})$$
(5.22)

Für jedes Beaconpaar und Unbekannten kann immer maximal ein Gap auftauchen. Je größer das Gap, desto größer ist auch der Fehler, der bei zumindest einer der beiden Distanzschätzungen entstanden ist.

Anpassung nicht plausibler Distanzen

Nachdem ein Gap erkannt wurde, hat ein Unbekannter die Möglichkeit, die geschätzten Distanzen für ein Beaconpaar und somit für die Berechnung eines Punktes im (g,h)-Raum so anzupassen, dass das Gap entsprechend auf 0 gesenkt wird, die Abstandsschätzungen zu den Beacons also korrekt wären, wenn der Unbekannte auf einer Geraden mit dem Beaconpaar läge. Die Anpassungen sind in TABELLE 5.1 dargestellt. Die Anpassungen wurden wie dort beschrieben gewählt, weil sich im Rahmen der Simulationen gezeigt, dass sich eine der eigentlichen Distanzschätzung adäquate Anpassung der Distanzen als genauigkeitssteigender erweist, als ein Gap gleichmäßig auf die beiden dazugehörigen Distanzschätzungen aufzuteilen.

Die Optimierung der nicht plausiblen Distanzen kann nun unabhängig auf jedes Beaconpaar angewendet werden. Das bedeutet durchaus, dass ein Unbekannter zur Bestimmung der Punkte des (g,h)-Raums für denselben Beacon unterschiedliche Distanzen verwenden darf. Das Ergebnis der optimierten Distanzschätzung für das NBL-Verfahren, im Folgenden als NBLAD (engl. Next Beacon as Linearizer with Adjusted Distancies) bezeichnet, ist in Abbildung 5.10 dargestellt. Es ist zu erkennen, dass der arithmetische Mittelwert der



Abbildung 5.10 Arithmetisches Mittel der Ergebnisse einer realitätsnahen Lokalisierung mittels Multilateration und angepassten nicht plausiblen Distanzen in den Szenarien (A) Flacher Sandstrand, (B) Trockenes Unterholz

Lokalisierung für dieselbe Versuchsreihe insbesondere für das zweite Szenario verbessert wird. Ursache ist hier die größere Schwankung der Empfangssignalstärke und damit einhergehend der höhere Fehler bei der Entfernungsschätzung, welcher häufig zu nicht plausiblen Distanzen führt.

Da die Optimierung der nicht-plausiblen Distanzen jedoch nur begrenzt Fehler in der ermittelten Distanzschätzung erkennt, ist es nicht möglich, die extremen Ausreißer zu eliminieren, die bei einer Multilateration entstehen.

5.3.3 Hybride Lokalisierung

Um die bei der Lokalisierung mittels NBLAD entstehenden Ausreißer zu eliminieren, bietet es sich an, die ermittelte Position auf ihre Plausibilität zu überprüfen und gegebenenfalls zu korrigieren. Diese Methodik wurde in HyPAERLoc verwendet. HyPAERLoc führt die Methode in 2 Schritten durch.

Plausibilität der erhaltenen Position

Um die Plausibilität der lokalisierten Position eines Unbekannten zu überprüfen, wird zuerst aus der mittels NBLAD bestimmten Position x_u , y_u des Unbekannten der Abstand d_{NBLAD} zu jedem Beacon *i* berechnet. Dies ist in Gleichung (5.23) beschrieben.

$$d_{i,u,NBLAD} = \sqrt{(x_{u,NBLAD} - x_i)^2 + (y_{u,NBLAD} - y_i)^2}$$
(5.23)

Danach vergleicht HyPAERLoc den geschätzten Abstand d zu jedem Beacon i mit dem berechneten Abstand und summiert den entstehenden Fehler quadratisch auf, wie in Gleichung (5.24) beschrieben.

$$RATE_{NBLAD} = \sum_{i=1}^{n} (d_{i,u,NBLAD} - d_{i,u})^{2}$$
(5.24)



Abbildung 5.11 Lokalisierungsfehler von 100 mit 3 Beacons mit AWCL lokalisierten Unbekannten und dazugehörige Lokalisierungsfehler von WCL und NBL

Das Resultat, hier als $RATE_{NBLAD}$ bezeichnet, spiegelt wieder, wie stark die geschätzten Distanzen von denen der ermittelten Positionen abweichen. Je kleiner der Wert $RATE_{NBLAD}$, desto genauer stimmen die errechneten Distanzen mit den geschätzten Distanzen überein.

Korrektur mittels hybrider Lokalisierung

Um eventuell schlechte Lokalisierungsergebnisse von NBLAD zu korrigieren, die mit der vorhergehenden Bewertung gefunden werden, ist eine alternative geschätzte Position notwendig, welche der Unbekannte statt dessen annimmt. Im Rahmen dieser Arbeit wurden verschiedene Möglichkeiten untersucht, eine geeignete Alternativposition zu finden. Da die Alternativposition ebenfalls aus einer Lokalisierung heraus entstehen muss, ist es wichtig, dass diese Position nicht mit dem Fehler von NBLAD korreliert.

Abbildung 5.11 zeigt den Lokalisierungsfehler von 100 Unbekannten mit jeweils drei empfangenen Beaconnachrichten und dem angewendeten AWCL-Verfahren auf der Abszisse. Der Punkt auf der Ordinate wird durch den Lokalisierungsfehler bestimmt, der für den selben Unbekannten mit einem anderen Verfahren ermittelt wird. Man kann erkennen, dass die Ergebnisse von WCL sich leicht durch eine lineare Funktion annähern lassen, was für eine starke Korrelation der erzielten Lokalisierungsergebnisse spricht. Im Gegensatz dazu korreliert NBL nur äußerst schwach mit AWCL. Daher bietet es sich an, das Ergebnis des heuristischen Lokalisierungsverfahren AWCL als Alternativposition zu verwenden, da es neben der schwachen Korrelation zudem die höchste Genauigkeit und den kleinsten Maximalfehler aller untersuchten heuristischen Verfahren besitzt.

Um zu ermitteln, ob die mittels heuristischem oder exaktem Verfahren bestimmte Position genauer ist, wird für AWCL nun ebenfalls zuerst der Abstand d_{AWCL} jedes Beacons von der durch AWCL ermittelten Position errechnet und der Algorithmus entsprechend bewertet, wie in Gleichung (5.25) und (5.26) beschrieben.

$$d_{i,u,AWCL} = \sqrt{(x_{u,AWCL} - x_i)^2 + (y_{u,AWCL} - y_i)^2}$$
(5.25)

$$RATE_{AWCL} = \sum_{i=1}^{n} (d_{i,u,AWCL} - d_{i,u})^{2}$$
(5.26)

Um die finale Position mittels zu ermitteln, auf welcher ein Unbekannter sich schließlich schätzt, wird im HyPAERLoc Verfahren eine Strecke mit den Eckpunkten AWCL-Position und NBLAD-Position gelegt und der Unbekannte positioniert sich auf dieser Strecke. Dabei errechnet der Unbekannte U seine Position x_u, y_u wie in Gleichung (5.27) und (5.28) beschrieben so, dass er der Position des besser bewerteten Algorithmus näher liegt.

$$x_{u,HyPAERLoc} = x_{u,NBLAD} + \frac{RATE_{NBLAD}}{RATE_{NBLAD} + RATE_{AWCL}} * (x_{u,AWCL} - x_{u,NBLAD})$$
(5.27)

$$y_{u,HyPAERLoc} =$$

$$y_{u,NBLAD} + \frac{RATE_{NBLAD}}{RATE_{NBLAD} + RATE_{AWCL}} * (y_{u,AWCL} - y_{u,NBLAD})$$
(5.28)

Vergleich mit den Vorgängerverfahren

Um die Genauigkeitssteigerung von HyPAERLoc zu demonstrieren, wurde derselbe in Kapitel 5.1.4 beschriebene, simulierte Versuchsaufbau wie bei den heuristischen und exakten Verfahren gewählt.

In Abbildung 5.12 sind die arithmetisches Mittel von den in HyPAERLoc vereinten Algorithmen dargestellt. Hier ist klar zu erkennen, dass der hybride Ansatz HyPAERLoc in der Lage ist, die Position eines Unbekannten im Mittel deutlich genauer zu bestimmen als seine Vorgänger, und die sich ergebenden



Abbildung 5.12 Arithmetisches Mittel der Ergebnisse der Lokalisierung mittels HyPAERLoc und der Vorgängerverfahren in den Szenarien (A) Flacher Sandstrand, (B) Trockenes Unterholz

Positionen enthalten deutlich weniger Ausreißer als die Vorgängerverfahren. Weiterhin ist zu erkennen, dass ab circa fünf empfangenen Beaconnachrichten für das Szenario Flacher Sandstrand keine signifikante Genauigkeitssteigerung mehr erzielt wird, während im Szenario "Trockenes Unterholz" mehr empfangene Beacons auch weiterhin zu einer höheren Genauigkeit führen.

Der Vergleich von HyPAERLoc mit seinen Vorgängerfahren mittels Boxplotdarstellung ist in Abbildung 5.13 gezeigt.

Man erkennt, das HyPAERLoc dank des Einflusses von NBLAD geringfügig höhere Ausreißer hat als AWCL haben kann. Ansonsten ist jedoch der deutliche Genauigkeitsgewinn, den HyPAERLoc gegenüber seinen Vorgängerverfahren bietet, in dieser Darstellung ebenfalls zu erkennen.

Der erzielte Gewinn kommt jedoch mit erhöhtem Aufwand daher. Zwar bleibt der Kommunikationsaufwand identisch mit einer Nachricht pro Beacon, der Berechnungsaufwand ist jedoch höher als der addierte Aufwand der Vorgängerverfahren. Eine genaue Untersuchung oder Optimierung des Rechenaufwands stand jedoch nicht im Fokus der vorliegenden Arbeit.



Abbildung 5.13 Boxplotdarstellung der Ergebnisse von HyPAERLoc und der Vorgängerverfahren in den Szenarien (A) Flacher Sandstrand, (B) Trockenes Unterholz

Dieses Kapitel schafft die Grundlagen für die realitätsnahe Simulation der positionsbasierten Clusteringverfahren. Für diese Verfahren ist es notwendig, dass jeder Knoten in der Lage ist, seine Position mittels eines Lokalisierungsverfahrens zu bestimmen. Daher wird für realitätsnahe Bedingungen im Folgenden davon ausgegangen, dass genug Beaconknoten verfügbar sind, um jeden Sensorknoten hinreichend genau mittels HyPAERLoc zu lokalisieren. Wie Kapitel 5.3.3 zeigt, wird mit fünf empfangenen Beaconnachrichten im Szenario Flacher Sandstrand eine angemessene Genauigkeit erreicht, welche durch das zusätzliche Ausbringen von Beacons nur noch begrenzt gesteigert wird. Um zu ermitteln, welche Anzahl im Feld ausgebrachter Beacons mit zufälliger Position notwendig ist, damit wenigstens fünf Beacons von einem Großteil der Sensorknoten empfangen werden, wurden gemäß der in Kapitel 5.1.4 beschriebenen Simulationsumgebung verschiedene Beacondichten und die dazugehörige Anzahl empfangener Beacons untersucht. Das Ergebnis dieser Untersuchungen ist in Abbildung 5.14 dargestellt und zeigt, dass die Anzahl empfangener Beacons für eine eingestellte Beacondichte in etwa normalverteilt ist. Für eine ausreichend genaue Lokalisierung wird eine Beacondichte von 15 Beacons je Hektar als ausreichend angesehen. Bei dieser Dichte empfängt ein Knoten im Durchschnitt circa sieben Beaconsignale, und nur weniger als 20% der platzierten Knoten weniger als fünf Beaconsignale. Im Trockenen Unterholz führt diese Beacondichte dank des geringeren Pfadverlustexponenten zu durchschnittlich 24 empfangenen Beaconnachrichten, was mehr als ausreichend für eine Lokalisierung ist.

Für die folgenden Annahmen wird daher von einer Beacondichte von 15 Beacons je Hektar ausgegangen. Die Beacons werden zusätzlich zu den eigentlichen Sensorknoten zufällig im Sensornetzwerk platziert und spielen für die eigentliche Sensornetzwerkfunktionalität keine Rolle.



Abbildung 5.14 Relative Häufigkeit empfangener Beacons gegenüber gewählter Beacondichte im Szenario Flacher Sandstrand



Abbildung 5.15 Histogramm und errechnete Normalverteilung der x-Position lokalisierter Knoten von HyPAERLoc in den Szenarien (A) Flacher Sandstrand (B) Trockenes Unterholz

Bei den Untersuchungen des positionsbasierten Clustering in den folgenden Kapiteln wird weiterhin der Einfachheit halber davon ausgegangen, dass Knoten benachbarter aktiver Cluster miteinander kommunizieren können. Dadurch genügt es, für die Sensorabdeckung jedem aktiven Knoten einen zufälligen Fehler zuzuweisen. Geht man von einer mittelwertfreien Gleichverteilung der Lokalisierungsfehler aus, ergibt sich die in Abbildung 5.15 dargestellte Verteilung der Lokalisierungsfehler in x- oder y-Richtung. Diese lässt sich für beide Szenarien durch eine mittelwertfreie Normalverteilung annähern, deren Standardabweichungen als $\sigma=2,9 m$ (Flacher Sandstrand) bzw. $\sigma=4,2 m$ (Trockenes Unterholz) bestimmt wurden.

Mittels dieser Parameter kann in folgenden realitätsnahen Simulationen jedem Knoten neben seiner echten Position seine errechnete Position zugewiesen werden und so der Einfluss einer ungenauen Lokalisierung auf die Sensornetzwerkfunktionalität ermittelt werden.

5.5 Kurzzusammenfassung

Das vorliegende Kapitel analysiert Methoden zur Lokaliseirung von Sensorknoten unter realitätsnahen Bedingungen. Kapitel 5.1 analysiert bekannte heuristische Verfahren, die einen guten Kompromiss zwischen Genauigkeit und Aufwand vermitteln. Kapitel 5.2 befasst sich danach mit den exakten Verfahren, welche zwar genaue Ergebnisse liefern können, jedoch auch anfälliger gegenüber Fehlerquellen sind. In Kapitel 5.3 wird auf Basis der Erkenntnisse dieser Analysen die Entwicklung des im Zuge der Arbeit entwickelten Algorithmus' HyPAERLoc beschrieben. Dieser Algorithmus kombiniert die Robustheit der heuristischen Verfahren mit der möglichen Genauigkeit der exakten Verfahren, und ermöglicht somit einen erheblichen Genauigkeitsgewinn unter realistischen Bedingungen. In Kapitel 5.4 wird anhand einer geeigneten Zahl von platzierten Beaconknoten definiert, welche Lokalisierung in einer realitätsnahen Umgebung mittels HyPAERLoc zu erwarten ist. Diese Erkenntnisse werden in den folgenden Kapiteln für die Bewertung der positionsbasierten Clusteringverfahren genutzt.

6 Positionsbasiertes Clustering

Clusteringverfahren, bei denen die Mitglieder eines Clusters aufgrund ihrer Position in einem Bezugssystem bestimmt werden, spielen eine bedeutende Rolle bei der Entwicklung energieeffizienter Sensornetzwerke. Das vorliegende Kapitel dient dazu, dem Leser die im Zuge dieser Arbeit entwickelten positionsbasierten Clusteringalgorithmen näher zu bringen. In Kapitel 6.1 wird in einem ersten Schritt der bereits bestehende GAF-Algorithmus auf drahtlose Sensornetzwerke erweitert, analysiert und auf andere Zellformen übertragen. In Kapitel 6.2 werden darauf aufbauend weiterentwickelte Clusteringalgorithmen vorgestellt. Diese sind in der Lage, mittels Mechanismen der Selbstheilung im Netzwerk entstandene Löcher in der Sensorabdeckung im Rahmen ihrer Möglichkeiten selbstständig zu erkennen und zu beheben. Kapitel 6.3 vergleicht die entwickelten selbstheilenden Clusteringalgorithmen mit dem erweiterten GAF-Algorithmen und dem bereits vorgestellten positionsbasierten zeitlichen Clusteringalgorithmus CPNS.

6.1 XGAF

Der *Extended Geographical Adaptive Fidelity* (kurz: XGAF) Algorithmus wurde im Zuge dieser Arbeit in [38] entwickelt, um den für Ad-hoc Netzwerke entwickelten und in Kapitel 2.4.2 beschriebenen GAF Algorithmus [43] auf die Bedürfnisse von Sensornetzwerken anzupassen. Ziel dabei war es, nur einen Knoten pro XGAF Zelle im aktiven Zustand behalten zu müssen und trotzdem komplette Netzwerkfunktionalität zu gewährleisten.

Da es für Sensornetzwerke entscheidend ist, neben der Sendereichweite auch die Sensorreichweite zu berücksichtigen, wird diese auch für die maximale Ausdehnung d_{max} einer XGAF Zelle berücksichtigt. So gilt für jede XGAF Zelle Gleichung (6.1).

$$d_{max,XGAF} = min \left(d_{max,T}; d_{max,S} \right) \tag{6.1}$$

Hierbei entspricht $d_{max,T}$ der maximalen Ausdehnung einer Zelle auf Basis der Sendereichweite. $d_{max,S}$ entspricht der maximalen Ausdehnung einer Zelle auf Basis der Sensorreichweite. Da es wie bei GAF Ziel von XGAF ist, nur einen Knoten pro Zelle für die Funktion des Sensornetzwerks zu benötigen, muss jeder Knoten einer Zelle in der Lage sein, die Sensorabdeckung seiner Zelle zu gewährleisten. Somit entspricht die maximale Ausdehnung $d_{max,S}$ einer Zelle immer der Sensorreichweite R_S , wie Gleichung (6.2) zeigt.



Abbildung 6.1 Angewandter SXGAF Algorithmus, (A) Ermittlung der maximalen Zellgröße in Abhängigkeit von Sensor- und Sendereichweite, (B) Mögliche Anordnung der Zellen, Zellgröße basiert auf der Sensorreichweite

$$d_{max,S} = R_S \tag{6.2}$$

Die maximale Ausdehnung einer Zelle $d_{max,T}$ hängt im Gegensatz zu $d_{max,S}$ von den zwei Faktoren ab.

Zum Ersten ist es entscheidend, ob eine Zelle auf maximale Größe oder wie in [70] auf Konnektivität zu sämtlichen Nachbarn optimiert wird, wobei letzteres in einer geringeren maximalen Zellgröße resultiert. Um mit allen umgebenden Nachbarn in 1-Hop-Distanz zu kommunizieren, kann die maximal erlaubte Ausdehnung einer Zelle generell mit der in [70] verwendeten *Working Range R_W* beschrieben werden, die in Gleichung (6.3) definiert ist.

$$R_W = min(0.5 * R_T, R_S) = d_{max}$$
(6.3)

Zum Zweiten ist die Form der Zelle entscheidend. Im Zuge der Arbeit wurden in [70] und [71] mehrere mögliche Zellformen untersucht, welche, richtig angeordnet, eine lückenlose Fläche aus gleichgroßen Zellen ergeben.

Aufgrund der unterschiedlichen Formen entstehen unterschiedliche erlaubte Zellausdehnungen. Die im Zuge der Arbeit untersuchten Zellformen und die resultierenden maximalen Zelllängen werden im Folgenden beschrieben.

6.1.1 Quadrat

Für XGAF, welches auf Clustern mit quadratischer Zellform (engl. Square XGAF, kurz SXGAF) basiert, errechnet sich die maximale Zellgröße im Bezug auf die Sendereichweite wie bei GAF in Gleichung (2.10). Die maximale Länge auf Basis der Sensorreichweite ergibt sich, wie in Abbildung 6.1 (A) gezeigt, aus der Diagonale der Zelle, da diese die größte Ausdehnung innerhalb eines Quadrates besitzt. Die maximale Zelllänge kann demnach mittels trigonometrischer Funktionen errechnet werden und ergibt sich wie in Gleichung (6.4) gezeigt.



Abbildung 6.2 Angewandter TXGAF Algorithmus, (A) Ermittlung der maximalen Zellgröße in Abhängigkeit von Sensor- und Sendereichweite, (B) Mögliche Anordnung der Zellen, Zellgröße basiert auf der Sensorreichweite

$$L_{max,SXGAF} = min(0,447 * R_T; 0,707 * R_s)$$
(6.4)

Die Anwendung von SXGAF auf ein Sensornetzwerk ist in Abbildung 6.1 (B) gezeigt.

6.1.2 Gleichseitiges Dreieck

Die nächste untersuchte Zellform für den XGAF Algorithmus ist das gleichseitige Dreieck (engl. Triangle XGAF, kurz: TXGAF). Die maximale Ausdehnung einer solchen dreieckigen Zelle aufgrund der Sensorreichweite entspricht genau der Seitenlänge, wie Abbildung 6.2 (A) zeigt. Um die Kommunikation zu einem beliebigen Knoten einer Nachbarzelle zu garantieren, darf die maximale Seitenlänge nicht größer sein als die ebenfalls in Abbildung 6.2 (A) dargestellte Strecke L_{TXGAF} . Anhand trigonometrischer Funktionen kann die jeweilige maximale Reichweite ermittelt werden und ergibt sich wie in Gleichung (6.5) gezeigt.

$$L_{max,TXGAF} = \min(0.577 * R_T; R_S)$$
(6.5)

Eine mögliche lückenlose Anordnung der Zellen ist in Abbildung 6.2 dargestellt.

6.1.3 Gleichseitiges Sechseck

Die lückenlose Abdeckung einer Fläche mit gleichseitigen Sechsecken, wie sie von Honigwaben bekannt ist, wurde mit Hexagon XGAF (kurz: HXGAF) ebenfalls untersucht. Die maximale Kantenlänge des HXGAF Sechsecks ergibt sich dabei wie in Abbildung 6.3 (A) dargestellt, und kann dementsprechend mittels trigonometrischer Funktionen berechnet werden, was in der in Gleichung (6.6) gezeigen Formel resultiert.



Abbildung 6.3 Angewandter HXGAF Algorithmus, (A) Ermittlung der maximalen Zellgröße in Abhängigkeit von Sensor- und Sendereichweite, (B) Anordnung der Zellen, Zellgröße basiert auf der Sensorreichweite

$$L_{max,HXGAF} = min (0,277 * R_T; 0,5 * R_S)$$
(6.6)

Die Anordnung von HXGAF Zellen zu einer lückenlosen Fläche ist in Abbildung 6.3 (B) dargestellt.

6.1.4 Unregelmäßiges Fünfeck

Als letzte untersuchte Zellform wurde ein unregelmäßiges Fünfeck (engl. Pentagon XGAF, kurz: PXGAF) gewählt. Um ein flächendeckendes Mosaik aus Fünfecken bilden zu können, wurden Fünfecke gewählt, bei denen jeweils 4 benachbarte Zellen ein Sechseck bilden. Diese Sechsecke lassen sich dann reanordnen. Wie Abbildung 6.4 zeigt, sind verschiedene gelmäßig Dimensionierungen möglich. Da jedoch 2 der Innenwinkel des Fünfecks immer mit 90° vorgegeben sind, ist niemals ein regelmäßiges Fünfeck möglich, dessen Innenwinkel jeweils 108° beträgt. Für die Untersuchungen im Zuge dieser Arbeit und in [70] wurde die Variante aus Abbildung 6.4 (B) gewählt, bei der die beiden längeren Diagonalen D_1 und D_2 gleichlang sind. Abbildung 6.5 (A) zeigt, welche Bedingungen für die maximalen Kantenlängen eines PXGAF Fünfecks gelten. Mittels trignometrischer Funktionen ergeben sich die maximalen Kantenlängen L und K eines PXGAF Fünfecks wie in Gleichung (6.7)



Abbildung 6.4 Mögliche fünfeckige Zellformen für den PXGAF Algorithmus. Jeweils 4 Zellen ergeben ein Sechseck, Diagonale D₂ und Flächeninhalt jeweils gleich groß, (A) D₁>D₂, (B) D₁=D₂, (C) D₁<D₂



Abbildung 6.5 Angewandter PXGAF Algorithmus, (A) Ermittlung der maximalen Zellgröße in Abhängigkeit von Sensor- und Sendereichweite, (B) Anordnung der Zellen, Zellgröße basiert auf der Sensorreichweite

und (6.8) beschrieben.

$$L_{max,PXGAF} = \min(0,423 * R_T; 0.732 * R_S)$$
(6.7)

$$K_{max,PXGAF} = \min(0,299 * R_T; 0,518 * R_S)$$
(6.8)

Die Anordnung von PXGAF Fünfecken zu einer lückenlosen Fläche ist in Abbildung 6.5 (B) dargestellt.

6.1.5 Vergleich der Zellformen

Während die beschriebenen Zellgrößen so gewählt wurden, dass die Größen einer Zelle ihren maximalen Wert erreichen, und trotzdem ein Routing möglich ist, wurden in [70] die Konnektivität der Zelle optimiert und mit Hilfe der Working Range R_W eine generell gültige maximale Ausdehnung einer Zelle geschaffen. TABELLE 6.1 gibt daher einen Vergleich über die erreichte Zellfläche A_Z und die entsprechende Anzahl der Nachbarn an, während Abbildung 6.6 die verschiedenen Formen und die garantiert erreichten Nachbarzellen gra-

TABELLE 6.1 EIGENSCHAFTEN VON XGAF ZELLEN BEI VERSCHIEDENEN

Optimierungen

PARAMETER	ALGORITHMUS			
	SXGAF	TXGAF	HXGAF	PXGAF
Optimierte Konnektivität mittels R _W				
Garantierte Nachbarn	8	12	6	7
Maximale Zellgröße in Ab- hängigkeit von R_T	$0, 125R_T^2$	$0,108R_T^2$	$0,162R_T^2$	$0, 125R_T^2$
Optimierte Zellgröße mittels L _{max}				
Garantierte Nachbarn	4	3	6	3
Maximale Zellgröße in Ab- hängigkeit von R_T	$0,2 R_T^2$	$0,144R_T^2$	$0, 2R_T^2$	$0,167R_T^2$



Abbildung 6.6 Maßstabsgetreue XGAF Zellgrößen und garantiert erreichbare Nachbarn in Abhängigkeit der Sendereichweite, jeweils auf Konnektivität und Größe optimiert (A) Square XGAF, (B) Triangle XGAF, (C) Hexagon XGAF, (D) Pentagon XGAF

phisch darstellt.

Es ist zu erkennen, dass es sich insbesondere bei HXGAF nicht lohnt, die maximale Zellgröße mittels R_W zu beschränken, da auch bei einer größenoptimierten Zelle alle sechs Nachbarn garantiert erreicht werden.

Für den Vergleich, welche der Zellformen am geeignetsten für den Einsatz in Sensornetzwerken ist, wurden die Zellformen mit der in TABELLE 6.2 vorgestellten Simulationsumgebung untersucht. Abbildung 6.7 zeigt die Anzahl an Hops an, die für einen Knoten notwendig sind, um die Senke zu erreichen. Dabei ist es nur erlaubt, benachbarte Zellen als nächsten Hop Richtung Senke auszuwählen. Je geringer die Anzahl an benötigten Hops, desto energieärmer ist eine Datenübertragung zur Senke.

Dabei ist es von der Art der XGAF-Zellform abhängig, ob eine konnektivitätsoder größenoptimierte Zelle weniger Hops benötigt. Für HXGAF und PXGAF bietet es sich an, Zellen mit optimaler Größe zu bilden, da hier eine Größenbe-

TABELLE 6.2 SIMULATIONSUMGEBUNG FÜR DEN VERGLEICH DER EIGENSCHAFTEN VON XGAF ZELLEN UNTER IDEALEN BEDINGUNGEN

AGAF ZELLEN UN IER IDEALEN DEDINGUNGEN					
PARAMETER	WERT				
Simulierte Fläche	270 m*270 m				
Evaluierte Fläche	162 m*162 m				
Sendereichweite	54 m				
Sensorreichweite	27 m				
Simulierte Netzwerke	2500				
Kanalausbreitung	Unit disk graph				
Knotenposition	Exakt				



Abbildung 6.7 Durchschnittliche Anzahl an notwendigen Hops eines Knotens zur Senke im Verhältnis zum Abstand

schränkung mittels der Working Range zu konservativ gewählt ist. Für SXGAF und TXGAF lohnt es sich dagegen, Zellen so zu kreieren, dass sie garantiert mit allen Nachbarn kommunizieren können. Insgesamt besitzt die größenoptimierte Variante von HXGAF die günstigste Anzahl an Nachbarn und Zellgröße, um möglichst wenige Hops zu einer Senke zu benötigen.

Für Zellen, deren maximale Größe durch die Sensorreichweite limitiert ist, wurde im nächsten Schritt untersucht, wie viele Knoten nötig sind, um die zur Verfügung stehenden Zellen zu besetzen. Abbildung 6.8 (A) zeigt den Anteil der besetzten Zellen des mit dem jeweiligen Algorithmus geclusterten Netzwerks an. Man erkennt, dass die HXGAF Zellen aufgrund ihrer größeren Ausdehnung und daraus resultierenden geringeren Anzahl besonders schnell besetzt sind. Aus diesem Grund weist HXGAF auch eine besonders hohe *Node Efficiency* auf, wie in Abbildung 6.8 (B) ersichtlich ist. Bei einer Anzahl an platzierten Knoten, bei der alle Algorithmen 100% besetzte Zellen besitzen, aktiviert HXGAF dementsprechend besonders wenig Knoten und funktioniert daher besonders gut, wenn viele Knoten im Netzwerk verteilt sind.

Um die Coverage-Performance der Algorithmen zu überprüfen, wurde in Abbildung 6.9 und Abbildung 6.10 die durch die Sensorabdeckung gegebene Netzwerkfunktionalität mit den in Kapitel 4.2 gegebenen Kriterien untersucht.



Abbildung 6.8 Performance der XGAF Algorithmen bei unterschiedlicher Anzahl platzierter Knoten, (A) Anteil besetzter Zellen im Verhältnis zur Gesamtanzahl an Zellen, (B) Node Efficiency



Abbildung 6.9 Anteil von mit XGAF Varianten geclusterten Sensornetzwerken mit 100% Netzwerkfunktionalität (Realistic Case Coverage)

Man kann in beiden Abbildungen erkennen, dass für HXGAF deutlich mehr Knoten notwendig sind als für die anderen XGAF Varianten, wenn eine komplette Netzwerkfunktionalität garantiert werden soll. Grund dafür ist Form und Größe der HXGAF Zellen, welche wenig Redundanz durch benachbarte Zellen bietet. Abbildung 6.11 (A) stellt die garantiert abgedeckte Fläche einer XGAF Zelle dar. Diese wird im Folgenden als *Working Area* A_W einer Zelle bezeichnet. In Abbildung 6.11 (B) ist zu erkennen, welcher Bereich einer unbesetzten Zelle dennoch garantiert durch Nachbarzellen abgedeckt wird. Diese Fläche wird im Folgenden als *Assisted Area* A_A einer Zelle bezeichnet. Sollte also eine Zelle unbesetzt sein, so wird nur ein Teil der Zelle nicht garantiert durch besetzte Nachbarzellen abgedeckt.

Abbildung 6.11 (E) und TABELLE 6.3 vergleichen die maximalen Zellgrößen und Working Areas der einzelnen XGAF Zellformen untereinander.



Abbildung 6.10 Anteil von mit XGAF Varianten geclusterten Sensornetzwerken mit 100% Netzwerkfunktionalität (Worst Case Coverage)



Abbildung 6.11 XGAF-Zellen und garantiert abgedeckte Flächen in Abhängigkeit von der Sensorreichweite, (A) SXGAF, (B) TGXAF, (C) HXGAF, (D) PXGAF, (E) Vergleich der Zellgrößen und entstehenden Working Areas maximal dimensionierter Zellen

Es ist zu erkennen, dass PXGAF-Zellen die höchste Working Area aufweisen. Eine unbesetzte Zelle wird jedoch in einer SXGAF Zelle am Wahrscheinlichsten abgedeckt, da es hier, im Gegensatz zu PXGAF zu keiner Überlappung der Working Areas benachbarter Zellen kommt. Die kleinste Assisted Area weist HXGAF auf, welches das Ergebnis in Abbildung 6.9 und Abbildung 6.10 plausibel macht. Obwohl HXGAF durch die Größe seiner Zellen die höchste Wahrscheinlichkeit für eine komplett besetzte Zellstruktur aufweist, hat bei HXGAF eine einzelne unbesetzte Zelle schon eine relativ hohe Wahrscheinlichkeit, ein Loch in der Sensorabdeckung zu generieren. Im Gegensatz dazu können SXGAF und PXGAF, aber auch TXGAF mit einer durchaus hohen Wahrscheinlichkeit eine komplette Netzwerkfunktionalität garantieren, selbst wenn einzelne Zellen im Netzwerk unbesetzt sind.

Sollte also eine möglichst hohe garantierte Sensorabdeckung im Netzwerk gefordert sein, die Anzahl maximal abgeworfener Knoten aber begrenzt sein, ist SXGAF daher der vorzuziehende Clusteringalgorithmus.

PARAMETER	ALGORITHMUS			
	SXGAF	TXGAF	HXGAF	PXGAF
Maximale Zellgröße	$0, 5R_S^2$	$0,433R_{s}^{2}$	$0,649R_S^2$	$0, 5R_S^2$
Working Area	$0,873R_{s}^{2}$	$0,705R_{S}^{2}$	$0,823R_{s}^{2}$	$0,879R_{S}^{2}$
Assisted Area	$0,373R_{s}^{2}$	$0,272R_{s}^{2}$	$0,174R_{s}^{2}$	$0,307R_{s}^{2}$

 TABELLE 6.3 COVERAGE-EIGENSCHAFTEN EINER MAXIMAL DIMENSIONIERTEN

 XGAF Zelle (DIMENSION IN ABHÄNGIGKEIT VON DER SENSORREICHWEITE)

6.2 MASCLE

Obwohl der Ansatz der XGAF-Algorithmen prinzipiell funktioniert und eine solide Redundanzerkennung und Ausnutzung aufweist, bedeutet jede aktive Zelle, dass ein Knoten permanent aktiv ist und Energie verbraucht. Die zugrunde liegenden Zellformen sind zudem so gewählt, dass sie die Working Areas benachbarter Zellen stark überlappen. Häufig, jedoch nicht immer, wäre es möglich, Zellen abzuschalten und so Energie zu sparen.

Das bedeutet, dass zum einen eine gewisse Knotendichte von Nöten ist, um einen Knoten in fast jeder Zelle zu garantieren, zum anderen der Anteil der sich in Betrieb befindlichen Knoten immer noch eine hohe Redundanz bezüglich der Netzwerkfunktionalität aufweist.

Ziel der Mutual Assistance in Cluster Environment а (kurz: MASCLE) -Algorithmen war es daher, zum einen den Anteil aktiver Knoten weiter zu reduzieren und zum anderen zu verhindern, dass eine einzelne unbesetzte/leere Zelle in der Lage ist, eine komplette Netzwerkfunktionalität zu gefährden. Die MASCLE-Algorithmen arbeiten daher mit einer angepassten maximalen Zellgröße, welche der entstehenden Zellstruktur Selbstheilungsfähigkeiten verleiht. Im Folgenden werden die entwickelten MASCLE-Algorithmen, deren Eigenschaften und Heilungsmöglichkeiten vorgestellt.

Obwohl bei der Ermittlung der Zellgröße anhand der Working Range R_W die Sendereichweite berücksichtigt wurde, lag das Hauptaugenmerk der Algorithmen auf einer Optimierung der lückenlosen Abdeckung des zu überwachenden Gebiets.

6.2.1 2 – MASCLE

Um eine garantierte Sensorabdeckung für eine ausgefallene oder leere XGAF-Zelle zu ermöglichen, bietet es sich insbesondere an, SXGAF als Ausgangsbasis zu verwenden, da hier, wie in Abbildung 6.11 (C) und TABELLE 6.3 zu sehen, der bereits garantiert durch Nachbarn abgedeckte Teil mit 74,6% besonders groß ist. Der in [61] im Zuge dieser Arbeit entwickelte *Dual Phase*-MASCLE (kurz: 2-MASCLE) Algorithmus basiert daher auf dem SXGAF Algorithmus, wobei für eine Zelle gilt, dass ihre Assisted Area A_A mindestens so groß ist wie die eigentliche Zellfläche. Ziel ist es dabei, die Fläche einer Zelle mit den Assisted Areas ihrer vier orthogonal benachbarten Zellen abzudecken.

Aufgrund der Symmetrie von SXGAF muss daher jede der vier Nachbarzellen einer 2-MASCLE Zelle zum einen in der Lage sein, mindestens ein Viertel der Zelle mit ihrer Working Area garantiert abzudecken, und zum anderen muss die Working Area A_W einer 2-MASCLE Zelle bis zur Mitte jeder orthogonal benachbarten Zelle reichen. Für die Berechnung der Zellgröße einer Zelle wird



Abbildung 6.12 Eigenschaften einer 2-MASCLE Zelle, (A) Geometrische Ermittlung der maximalen Größe, (B) Resultierende Working Area und partielle Abdeckung einer Nachbarzelle, (C) Kommunikationsreichweite und garantiert erreichbare Nachbarzellen, (D) Einteilung der Zellen in verschiedene Phasen, (E) Nummerierung der Zellen zur Bestimmung der Phase

die in Gleichung (6.3) beschriebene Working Range herangezogen, welche Sensor- und Kommunikationsreichweite berücksichtigt.

Die geometrische Ermittlung der maximalen Zellgröße ist in Abbildung 6.12 (A) dargestellt. Die Berechnung der maximalen Zelllänge und das Verhältnis zu SXGAF ist in Gleichung (6.9) beschrieben.

$$L_{2M} = 0.632 * R_W = 0.894 * L_{SXGAF} \tag{6.9}$$

Somit sinkt die maximale Zelllänge verglichen mit SXGAF um ca. 10,6%, was zu einer Zellfläche von 80% verglichen mit SXGAF führt. Zwar führt diese Verkleinerung der Zellen auf den ersten Blick zu 25% mehr Clustern zur Überwachung derselben Fläche, jedoch decken nun vier diagonal benachbarte 2-MASCLE Zellen nun garantiert die eingeschlossene Zelle mit ihrer Working Area ab, welche in Abbildung 6.12 (B) dargestellt ist. Gleichzeitig wird die Anzahl garantiert erreichter Nachbarzellen auf 12 angehoben.

Da es für eine garantierte Sensorabdeckung des zu überwachenden Gebiets ausreichend ist, wenn die Working Areas aktiver Zellen das Gebiet abdecken, ist es bei 2-MASCLE somit möglich, jede zweite Zelle im regulären Betrieb abzuschalten, wie in Abbildung 6.12 (D) dargestellt. Dabei wird eine Kommunikation zur Senke nicht gefährdet, da immer noch 8 Nachbarn einer Zelle aktiv und erreichbar sind. Um das auszunutzen, findet daher bei 2-MASCLE, ähnlich zu den in Kapitel 2.4.1 vorgestellten k-Cover-Algorithmen, nach einer gewissen Zeit T ein Phasenwechsel statt. Bei jedem Phasenwechsel schalten sich inaktive Cluster in den Betriebszustand und aktive Cluster werden in einen Schlafzustand versetzt. Das bedeutet, dass selbst der Clusterhead dieser Zellen temporär in einen Schlafmodus gesetzt wird. Nach einer gewissen Zeit findet wieder ein Phasenwechsel statt, jeder aktive Cluster wechselt erneut in den Schlafzustand und umgekehrt. Dieser Wechsel setzt sich periodisch fort, sodass der Algorithmus in zwei Phasen arbeitet, welches zur Namensgebung geführt hat. Jeder Cluster ist in der Lage, die zu einem Zeitpunkt t aktive Phase P_{akt} mittels Gleichung (6.12) zu errechnen.

$$P_{akt} = \lfloor t/T \rfloor \mod N \tag{6.10}$$

N gibt dabei die Anzahl der Phasen an, für 2-MASCLE gilt N=2. Da der Vorgang zeitkritisch ist, sollte die Phasenlänge T so gewählt werden, dass eine Zeitdrift kompensiert werden kann.

Um ihre Phasenzugehörigkeit zu bestimmen, muss eine Zelle zuerst ihre Position im virtuellen Grid herausfinden. Da für die Einordnung eines Knotens seine Position und somit Abstand D von der Senke bekannt sein muss, dient die Senke als Koordinatenursprung. Jeder Knoten kann die Zellkoordinaten C[x,y]seiner Zelle dementsprechend mittels Gleichung (6.11) ausrechnen.

$$C[x] = D_x \, div \, L_{2M}; \, C[y] = D_y \, div \, L_{2M} \tag{6.11}$$

Durch Anwendung von Gleichung (6.11) ergibt sich die in Abbildung 6.12 (E) zu erkennende Zellnummerierung, mit der die Phasenzugehörigkeit P_{cell} jedes Cluster innerhalb einer Zelle bestimmt werden kann. Um die eigene Phasenzugehörigkeit P_{cell} zu ermitteln, wird von jedem Cluster Gleichung (6.12) angewendet.

$$P_{cell} = (C[x] + C[y]) \mod 2$$
 (6.12)

Durch Ausführung der Gleichung entsteht die bereits in Abbildung 6.12 (E) zu erkennende Schachbrettstruktur. Es sind also alle Zellen einer Phase diagonal benachbart und es kann eine komplette Abdeckung durch eine einzelne Phase erreicht werden. Der einfache 2-MASCLE Algorithmus ist in Abbildung 6.13 (A) als Flussdiagramm dargestellt.



Abbildung 6.13 Flussdiagramm des 2-MASCLE Algorithmus, (A) 2-MASCLE ohne Heilung, (B) 2-MASCLE mit Selbstheilungsfunktion

Clusterbasierte Selbstheilung

Mit der Fähigkeit, in 2 Phasen zu arbeiten, und dennoch eine garantierte komplette Netzwerkfunktionalität zu bewahren, solange alle Zellen besetzt sind, ist 2-MASCLE in der Lage, nur 62,5% der von SXGAF benötigten Knoten einschalten zu müssen. Somit kann mit dem in Kapitel 4.1.4 beschriebenen vereinfachtem Energiemodell zur Laufzeit 37,5% der von SXGAF benötigten Energie gespart werden.

Jedoch führt ein Ausfall einer 2-MASCLE Zelle mit einer deutlich erhöhten Wahrscheinlichkeit zu einem Loch in der Sensorüberwachung im Sensornetzwerk, da nicht nur die ausgefallen Zelle an sich, sondern auch die außerhalb der Zelle liegende Working Area der ausgefallenen Zelle nicht mehr garantiert durch Nachbarzellen abgedeckt wird.

Die durch regulär aktive Nachbarn garantierte Assisted Area ist in Abbildung 6.14 (A) dargestellt. Da es im Gegensatz zu SXGAF gelingen muss, die



Abbildung 6.14 Heilung in 2-MASCLE, (A) Garantierte Abdeckung einer ausgefallenen Zelle ohne Heilung, (B) Komplette Heilung einer ausgefallenen Zelle durch alle 4 orthogonal benachbarten Zellen, (C) Vergleich der Flächen von SXGAF und 2-MASCLE

gesamte Working Area abzudecken, ist das Verhältnis von Assisted Area zu Working Area mit 48,3% deutlich kleiner als bei SXGAF, bei dem das Verhältnis von Assisted Area zu Zellfläche wie bereits beschrieben 74,6% beträgt. Die Größenverhältnisse sind in Abbildung 6.14 (C) dargestellt.

Durch diese kleinere Assisted Area ist es nun wesentlich unwahrscheinlicher, zufällig das Gebiet einer ausgefallenen Zelle zu heilen. Glücklicherweise ist 2-MASCLE in der Lage, eine garantierte Selbstheilung ausgefallener oder unbesetzter Zellen durchzuführen und ausgefallene Zellen zu kompensieren.

Um eine Selbstheilung durchzuführen, wird der Algorithmus wie in Abbildung 6.13 (B) gezeigt erweitert. Nach einem Phasenwechsel sendet jeder in den Betriebszustand wechselnde Cluster einen Broadcast, den Life Sign, welcher die Clusternummer enthält. Erst wenn ein Cluster, welcher in den Schlafzustand wechseln will, die Life Signs aller 4 orthogonal benachbarten Cluster empfangen hat, ist es ihm gestattet, in den Schlafmodus zu wechseln. Das bedeutet im Umkehrschluss, dass alle 4 orthogonalen Nachbarn einer ausgefallenen Zelle auch in der folgenden Phase aktiv bleiben und die Funktionalität des ausgefallenen Clusters übernehmen. In Abbildung 6.14 (B) ist dargestellt, dass eine ausgefallene Zelle komplett geheilt werden kann, wenn die 4 orthogonal benachbarten Zellen als heilende Zellen fungieren und nicht in einen Schlafmodus versetzt werden. Dabei ist zu erkennen, dass sowohl die ausgefallene Zelle an sich als auch deren Working Area garantiert geheilt wird. Die mit der Selbstheilung verbundene Effizienzsteigerung und das Netzwerkverhalten gegenüber einem mit SXGAF geclustertem Netzwerk wird in Kapitel 6.3 gezeigt.

6.2.2 4 – MASCLE

2-MASCLE erlaubt bereits eine signifikante Verbesserung der Lebenszeit und der Anzahl der für komplette Netzabdeckung benötigten Knoten. Als Weiterentwicklung von 2-MASCLE verfolgt das in [72] im Zuge dieser Arbeit entwickelte *Quadruple Phase*-MASCLE (kurz: 4-MASCLE) das Ziel, nur noch jede vierte Zelle zu jedem Zeitpunkt im Betriebsmodus zu betreiben, und dennoch komplette Netzwerkfunktionalität zu gewährleisten. Es sollten weiterhin quadratische, im Gitter angeordnete Zellen verwendet werden und die Nummerierung der Zellen C[x,y] wird ähnlich zu XGAF auf Basis des Abstands zur Senke *D* gemäß Gleichung (6.13) durchgeführt.

$$C[x] = D_x \, div \, L_{4M}; \, C[y] = D_y \, div \, L_{4M} \tag{6.13}$$

Im Gegensatz zu 2-MASCLE soll der Algorithmus nun in vier Phasen ablaufen, sodass für 4-MASCLE Gleichung (6.10) mit N=4 gilt.

Für die Anordnung der aktiven Zellen bieten sich 2 verschiedene Varianten an, das sogenannte symmetrische 4-MASCLE sowie die asymmetrische 4-MASCLE Variante, die im Folgenden beschrieben werden.

Symmetrisches 4-MASCLE

In der symmetrischen 4-MASCLE Variante befinden sich alle Zellen jeder 2. Reihe und jede 2. Spalte des virtuellen Gitters komplett im Schlafmodus. Um eine komplette Abdeckung zu erreichen, muss dazu die Working Area einer aktiven Zelle mindestens bis zum Mittelpunkt jeder diagonal benachbarten Zelle reichen, wie in Abbildung 6.15 (A) zu erkennen ist. Die resultierende Zelllänge einer symmetrischen 4-MASCLE Zelle L_{S4M} ergibt sich nach Umstellen der entsprechenden trigonometrischen Funktionen in Abhängigkeit von der Working Range R_W nach Gleichung (6.14).

$$L_{S4M} = 0.471 * R_W = 0.667 * L_{SXGAF} \tag{6.14}$$

Mit dieser Zellgröße ist es nun möglich, die aktiven Zellen wie in Abbildung 6.15 (C) dargestellt anzuordnen.

Mit der gegebenen Zelllänge, seiner Position und dem Abstand von der Senke kann sich jeder Knoten nun in die ihm zugehörige Zelle einordnen und mit den anderen der Zelle zugehörigen Knoten einen Cluster bilden. Damit zu jedem Zeitpunkt *t* nur jede vierte Zelle aktiv ist, müssen sich die Zellen entsprechend



Abbildung 6.15 Eigenschaften von 4-MASCLE Zellen, (A) Geometrische Ermittlung der maximalen Größen, (B) Resultierende Working Area und partielle Abdeckung von Nachbarzellen, (C) Einteilung der Zellen in verschiedene Phasen, (D) Kommunikationsreichweite und garantiert erreichbare Nachbarzellen,

ihrer Nummerierung mittels Gleichung (6.15) die ihnen zugehörige Phase ausrechnen.

$$P_{cell} = C[x]mod \ 2 + 2(C[y]mod \ 2) \tag{6.15}$$

Durch Anwendung dieser Gleichung weist sich jede Zelle eine Phase zwischen 0 und 3 zu, die entsprechend im Netzwerk verteilt sind.

Asymmetrisches 4-MASCLE

Die asymmetrische 4-MASCLE Variante unterscheidet sich von der symmetrischen Variante dadurch, dass jede zweite mit aktiven Zellen besetzte Reihe um eine Zelle verschoben ist, wie Abbildung 6.15 (C) zeigt. Die Zellgröße darf also maximal so groß gewählt werden, dass eine inaktive Zelle entweder durch einen orthogonalen Nachbarn und die gegenüberliegenden diagonalen Nachbaren oder durch 2 gegenüberliegende orthogonale Nachbarn abgedeckt wird. Die maximale Dimensionierung einer asymmetrischen 4-MASCLE Zelle ergibt sich daher wie in Abbildung 6.15 (A) dargestellt. Die resultierende Zelllänge einer asymmetrischen 4-MASCLE Zelle L_{A4M} ergibt sich nach Umstellen der entsprechenden trigonometrischen Funktionen nach Gleichung (6.16).

$$L_{A4M} = 0.526 * R_W = 0.74 L_{SXGAF} \tag{6.16}$$

Um die Verschiebung jeder zweiten aktiven Reihe zu berücksichtigen, verändert sich die Berechnung der einer Zelle zugehörigen Phase wie in Gleichung (6.17) beschrieben.

$$P_{cell} = (C[y]div \, 2 + C[x])mod \, 2 + 2 * (C[y]mod \, 2)$$
(6.17)

Genau wie bei der symmetrischen Variante weist sich jede Zelle durch Anwendung dieser Gleichung eine Phase zwischen 0 und 3 zu, die entsprechend im Netzwerk verteilt sind.

Clusterbasierte Selbstheilung

Ähnlich zu 2-MASCLE besitzen beide Varianten von 4-MASCLE die Eigenschaft, jeweils in mehreren Phasen zu arbeiten und dennoch eine garantierte komplette Netzwerkfunktionalität zu bewahren, solange alle aktiven Zellen besetzt sind. So muss nur jede vierte vorhandene Zelle in 4-MASCLE zu jedem Zeitpunkt aktiv sein, was zusammen mit der verkleinerten Zellfläche dazu führt, dass bei der symmetrischen 4-MASCLE Variante nur 56,25% und bei asymmetrischem 4-MASCLE sogar nur 45,65% der von SXGAF benötigten Knoten eingeschaltet werden müssen, sodass mit dem in Kapitel 4.1.4 beschriebenen



Abbildung 6.16 Assistierte Fläche ausgefallener Zellen in (A) symmetrischem 4-MASCLE, (B) asymmetrischem 4-MASCLE; (C) Vergleich der Flächen von 2-MASCLE und beider Varianten von 4-MASCLE

einfachem Energiemodell 44,75% bzw. 54,25% der von SXGAF benötigten Energie gespart werden.

Jedoch führt ein Ausfall einer 4-MASCLE Zelle mit einer noch höheren Wahrscheinlichkeit zu einer Lücke bei der Sensorüberwachung im Sensornetzwerk, da wie bei 2-MASCLE nicht nur die ausgefallen Zelle an sich, sondern auch die außerhalb der Zelle liegende Working Area der ausgefallenen Zelle nicht mehr garantiert durch Nachbarzellen abgedeckt wird. Obwohl die Assisted Area, wie in Abbildung 6.16 (C) zu sehen, für beide Varianten größer als bei 2-MASCLE ist, deckt sie die eigentliche Kernzelle wie in Abbildung 6.16 (A) und (B) dargestellt kaum noch oder gar nicht mehr ab, sodass mit einer hohen Wahrscheinlichkeit eine ausgefallene Zelle zu einer Einschränkung der Netzwerkfunktionalität führt. Genau wie 2-MASCLE ist 4-MASCLE jedoch in der Lage, eine ausgefallene Zelle durch die zusätzliche Aktivierung benachbarter Zellen zu heilen. Um die Working Area garantiert abzudecken, ist es bei 4-MASCLE allerdings nicht notwendig, sämtliche im Schlafzustand befindliche Nachbarzellen zu aktivieren, wie es bei 2-MASCLE der Fall ist. Stattdessen ist es möglich, dass nur benachbarte Zellen aktiviert werden, welche wirklich benötigt werden. Die Anordnung dieser benachbarten aktiven Zellen wird im Folgenden als Assistance Arrangement bezeichnet und die garantierte Heilung mittels solcher im Folgenden beschrieben.

Clusterbasierte Selbstheilung mittels Assistance Arrangements

Größter Vorteil der Heilung mittels Assistance Arrangements stellt die Energie dar, die gespart werden kann. Werden anstatt alle 8 Nachbarn einer ausgefallenen 4-MASCLE Zelle beispielsweise nur 3 Nachbarzellen für eine Heilung benötigt, muss auch nur der entsprechende Anteil, d.h. 37,5% der Energie für die Heilung dieser Zelle aufgewendet werden. Die Heilung wird also energieeffizienter.



Abbildung 6.17 Flussdiagramm der Selbstheilung in MASCLE mittels Assistance Arrangements

Um eine Selbstheilung mittels Assistance Arrangements zu erreichen, ist es nötig den in Abbildung 6.13 (B) dargestellten Heilungsalgorithmus zu erweitern, wie es in Abbildung 6.17 gezeigt wird. Sollte zu Beginn eines Phasenwechsels ein Cluster einer Zelle durch das Ausbleiben des Life Sign Broadcast feststellen, dass nach einem Phasenwechsel eine nun eigentlich aktive Nachbarzelle Z unbesetzt oder ausgefallen ist, bleibt er nicht automatisch aktiv, sondern sendet einen *Possible Assistance* Broadcast an alle anderen Nachbarn der ausgefallenen Zelle. Da jeder Nachbar einer ausgefallenen Zelle gemäß der auf der Working Range basierenden maximalen Zellgröße bei symmetrischem 4-MASCLE in der Lage ist, alle Nachbarn der Zelle Z zu empfangen, kann jeder Knoten ermitteln, welche anderen, möglicherweise heilenden Zellen noch leben und welche nicht. Für asymmetrisches 4-MASCLE können die Broadcasts der diagonalen Nachbarn einer ausgefallenen Zelle Z durch Wiederholung der orthogonalen Nachbarn ebenso jedem Nachbarn zur Verfügung gestellt werden, wie in [72] beschrieben.

Nach einer gewissen Zeit $T_a \ll T$ kann davon ausgegangen werden, dass jede lebende Nachbarzelle der ausgefallenen Zelle Z über die Verfügbarkeit der anderen Nachbarzellen von Z Bescheid weiß.

Anhand des Wissens, welche benachbarten Zellen einer ausgefallenen Zelle Z noch am Leben sind, wählt darauf hin jeder Cluster für sich die energieeffizienteste mögliche Assistance Arrangement und prüft, ob er für deren Anwendung im Betriebsmodus befindlich sein muss.


Abbildung 6.18 Energieeffiziente Heilung in symmetrischem 4-MASCLE mittels Assistance Arrangements, (A) mit 2 Zellen, (B,C) mit 3 Zellen, (D) mit 4 Zellen; (E) Von links nach rechts sinkend: Assistance Arrangements sortiert nach Energieeffizienz

Sollten mehrere benachbarte, eigentlich aktive Zellen ausgefallen sein, darf ein Cluster nur dann in den Schlafzustand wechseln, wenn er für keine der Assistance Arrangements benötigt wird.

Um herauszufinden, welche Assistance Arrangements eine garantierte Heilung einer ausgefallenen Zelle ermöglichen, wird in dieser Arbeit der Begriff der gültigen Assistance Arrangement eingeführt. Die gültige Assistance Arrangement ist wie folgt definiert:

Definition gültige Assistance Arrangement:

Eine Assistance Arrangement ist genau dann gültig, wenn die zur Heilung einer leeren Zelle Z aktivierten Zellen in der Lage sind, die Working Area von Z komplett mit ihren eigenen Working Areas und den Working Areas von aktiven, möglicherweise selbst geheilten Nachbarzellen abzudecken.

In [72] wurde zum schnellen Auffinden von möglichen Assistance Arrangements von dieser Definition abgewichen und die effektive Working Area einer symmetrischen 4-MASCLE Zelle auf ein Quadrat und für die asymmetrische Version auf ein unregelmäßiges Sechseck reduziert. Dies führt in [72] genau dann zu einem gültigen Assistance Arrangement, sobald $R_{W,eff}$ durch benachbarte Zellen abgedeckt wird. Für symmetrisches 4-MASCLE ergibt sich damit jedoch kein Unterschied und es ergeben sich Assistance Arrangements, wie sie in Abbildung 6.18 (A-D) gezeigt sind. Durch Rotieren in 90° Schritten um die





Abbildung 6.19 Energieeffiziente Heilung in asymmetrischem 4-MASCLE mittels Assistance Arrangements, (A-D) mit 4 Zellen; (E) Von links nach rechts sinkend: Assistance Arrangements sortiert nach Energieeffizienz

ausgefallene Zelle erhält man sämtliche möglichen gültigen Assistance Arrangements, wie in Abbildung 6.18 (E) dargestellt.

Im Falle des asymmetrischen 4-MASCLE können durch Verwendung der obigen Definition im Vergleich zu [72] weitere energieeffiziente gültige Assistance Arrangements gefunden werden, welche eine noch energieeffizientere Heilung einer ausgefallenen Zelle ermöglichen.

Die gültigen energieeffizientesten Assistance Arrangements für asymmetrisches 4-MASCLE sind in Abbildung 6.19 dargestellt. Dabei ist zu erkennen, dass die in Abbildung 6.19 (A,B) dargestellten Assistance Arrangements offensichtlich nicht um die ausgefallene Zelle rotierbar sind. Der Grund dafür ist in Abbildung 6.20 dargestellt. Liegen zwei benachbarte und ausgefallene Zellen nebeneinander und die umliegenden Knoten wählen für beide eine Assistance Arrangement, bei denen keine der heilenden Zellen mit beiden ausgefallenen Zellen benachbart ist, kann es trotz Heilung zu einer Fläche kommen, die nicht garantiert abgedeckt ist.



Abbildung 6.20 Unzureichende Heilung in asymmetrischem 4-MASCLE von 2 benachbarten Zellen durch Heilung der linken Zelle mit ungültiger Assistance Arrangement

Es gelten daher nur dann Assistance Arrangements als gültig, die zumindest eine Zelle rechts der ausgefallenen Zelle zur Heilung aktivieren.

Damit die Heilung per gültiger Assistance Arrangement funktioniert, müssen alle benachbarten Zellen einer ausgefallenen Zelle Z dasselbe Assistance Arrangement auswählen. Daher werden die Assistance Arrangements gemäß ihrer Energieeffizienz wie in Abbildung 6.18 (E) bzw. Abbildung 6.19 (F) dargestellt von links nach rechts mit sinkender Energieeffizienz sortiert und auf jedem Knoten fest gespeichert. Für den Fall, dass mögliche Assistance Arrangements dieselbe Anzahl heilender Zellen, also die gleiche Energieeffizienz besitzen, wurden die Assistance Arrangements willkürlich sortiert.

Sobald jedem umliegenden Cluster bekannt ist, welche umliegenden Cluster noch verfügbar sind, wird das sich am weitesten links befindliche mögliche Assistance Arrangement im Zuge des Algorithmus von allen Nachbarn ausgewählt. Sollte kein gültiges Assistance Arrangement mehr möglich sein, bleiben einfach alle verbliebenen Nachbarzellen der ausgefallenen Zelle Z aktiv, um so zu versuchen die ausgefallene Zelle mittels eines ungültigen Assistance Arrangements zu heilen.

Die Effizienzsteigerung der 4-MASCLE Varianten und das Netzwerkverhalten gegenüber einem mit SXGAF oder 2-MASCLE geclustertem Sensornetzeerk wird in Kapitel 6.3 gezeigt.

6.2.3 Hex – MASCLE

Ebenso wie XGAF sind die MASCLE-Algorithmen nicht auf quadratisch geformte Zellen beschränkt, sondern für jede mosaikbildende Zellform möglich. Da neben Quadraten regelmäßige Hexagone aufgrund ihrer relativ großen Zellfläche vielversprechend sind, wurde in [73] daher der MASCLE-Algorithmus auf Sechsecke angewendet und als *Hexagon based*-MASCLE (kurz: Hex-MASCLE) bezeichnet. Im Zuge der Arbeit wurden zwei Varianten von Hex-MASCLE analysiert.



Abbildung 6.21 Anordnung aktiver Zellen in Hex-MASCLE, (A) 3-phasiges Hex-MASCLE (B) 4-phasiges Hex-MASCLE, (C) Nummerierung der Zellen zur Bestimmung der Phase

*Hex-MASCLE*³

Die erste Version für eine auf Hexagonen basierende MASCLE-Variante hat das Ziel, drei Phasen im Netzwerk zu etablieren. Dazu wurde eine Verteilung der aktiven Zellen wie in Abbildung 6.21 (A) gewählt. Eine aktive Zelle ist also nur von inaktiven Zellen umgeben, während jede inaktive Zelle drei aktive Nachbarn besitzt. Mit einer wie in Abbildung 6.21 (C) gezeigten Zellnummerierung kann jeder Cluster die aktive Phase seiner Zelle mittels Gleichung (6.18) ausrechnen.

$$P_{cell} = (C[x] + (C[y] \mod 2) \mod 3$$
(6.18)



Um zu garantieren, dass in der gewählten Anordnung von aktiven und inakti-

Abbildung 6.22 Eigenschaften von Hex-MASCLE Zellen auf Basis der *Working Range*, (A) Ermittlung der maximalen Zelldimension, (B) Erzielte Working Area, (C) Garantiert erreichte Nachbarn

ven Zellen eine ausreichende Konnektivität und Sensorabdeckung gewährleistet werden kann, muss die Working Area mindestens ein Drittel jeder Nachbarzelle abdecken und die Working Range bis zum Mittelpunkt jeder Nachbarzelle reichen, wie in Abbildung 6.22 (A) dargestellt. Daraus ergibt sich die Form und Größe der Working Area, wie sie in Abbildung 6.22 (B) zu sehen ist und eine garantierte Konnektivität, wie sie Abbildung 6.22 (C) zeigt. Die aus dieser Bedingung resultierende maximale Zelllänge L_{HM3} ergibt sich nach Abbildung 6.22 (A) und ist in Gleichung (6.19) gezeigt.

$$L_{HM3} = 0.378 * R_W = 0.76 * L_{HXGAF}$$
(6.19)

*Hex-MASCLE*⁴

Ziel der 2. Variante von Hex-MASCLE ist es, zu jedem Zeitpunkt *t* nur jede vierte Zelle im regulären Netzbetrieb im aktiven Modus zu halten. Es ergeben sich effektive Anordnungen der aktiven Zellen ähnlich zu 4-MASCLE, bei der jede zweite Reihe komplett abgeschaltet wird. In jeder noch verbleibenden Reihe wird jede zweite Zelle aktiviert, sodass sich durch das Verschieben jeder zweite aktiven Reihe eine symmetrische und eine asymmetrische Variante von Hex-MASCLE⁴ ergeben. Aufgrund der Tatsache, dass bei 4-MASCLE die asymmetrische Variante eine bessere Performance erzielt, wird hier nur die asymmetrische Variante von Hex-MASCLE⁴ untersucht, die im Folgenden einfach als Hex-MASCLE⁴ bezeichnet wird. Wie bereits erläutert, ergibt sich eine Anordnung der aktiven Zellen ähnlich dem asymmetrischen 4-MASCLE, wie Abbildung 6.21 (B) zeigt. Um zu dieser Anordnung zu gelangen, müssen die Cluster innerhalb der Zellen ihre zugehörige Phase bestimmen. Mit der in Abbildung 6.21 (C) gewählten Zellnummerierung ergibt sich die Phase einer Zelle nach Gleichung (6.20).

$$P_{cell} = \left(C[x] + \frac{C[y]}{2}\right) \mod 2 + C[Y] \mod 2$$
(6.20)

Damit in der somit ausgewählten Anordnung aktiver und inaktiver Zellen die Konnektivität zwischen aktiven Zellen und die Sensorabdeckung durch aktive Zellen in Hex-MASCLE⁴ gewährleistet werden kann, muss die Working Area einer aktiven Zelle mindestens die Hälfte jeder Nachbarzelle abdecken, wie Abbildung 6.22 (A) darstellt. Genau wie in Hex-MASCLE³ ergibt sich in Hex-MASCLE⁴ die Form und Größe der Working Area wie in Abbildung 6.22 (B) zu sehen und eine garantierte Konnektivität wie Abbildung 6.22 (C) zeigt. Die aus dieser Bedingung resultierende maximale Zelllänge L_{HM3} ergibt sich nach Abbildung 6.22 (A) in Gleichung (6.21).

$$L_{HM3} = 0.333 * R_W = 0.67 * L_{HXGAF}$$
(6.21)

Clusterbasierte Selbstheilung

Genau wie die anderen MASCLE-Varianten besitzen beide Varianten von Hex-MASCLE die Eigenschaft, jeweils in mehreren Phasen zu arbeiten und dennoch eine garantierte komplette Netzwerkfunktionalität zu bewahren, solange alle als aktiv geltende Zellen besetzt sind. Die Aktivierung jeder dritten Zelle bei Hex-MASCLE³ und jeder vierten Zelle bei Hex-MASCLE⁴ führt dazu, dass nur 45,0% bzw. 43,2% der von SXGAF benötigten Knoten eingeschaltet werden müssen, was zu einer entsprechenden Energieersparnis führt.

Genau wie bei einer 4-MASCLE Zelle führt bei Hex-MASCLE ein Ausfall einer Zelle mit einer hohen Wahrscheinlichkeit zu einer Lücke bei der Sensorüberwachung im Sensornetzwerk. Der Grund dafür ist in Abbildung 6.23 (A) und Abbildung 6.23 (B) dargestellt. Dort erkennt man deutlich, dass die eigentliche Kernzelle im Falle des Ausfalls einer Zelle bei beiden Hex-MASCLE Varianten gar nicht mehr abgedeckt wird. Zudem ist, wie in Abbildung 6.23 (C) zu sehen ist, die Assisted Area bei Hex-MASCLE³ im Vergleich zu den Assisted Areas der anderen MASCLE-Varianten extrem klein. Es ist in Hex-MASCLE also unbedingt notwendig, eine Überprüfung der aktiven Zellen und eine anschließende Heilung mittels Assistance Arrangements durchzuführen.

Clusterbasierte Selbstheilung mittels Assistance Arrangements

Ähnlich wie 4-MASCLE kann in Hex-MASCLE eine ausgefallene oder unbesetzte Zelle mittels einer Auswahl von Nachbarzellen geheilt werden. Im Gegensatz zu 4-MASCLE besitzen beide Versionen von Hex-MASCLE dieselben gültigen Assistance Arrangements, wie sie in Abbildung 6.24 dargestellt



Abbildung 6.23 Assistierte Fläche ausgefallener Zellen in (A) Hex-MASCLE³, (B) Hex-MASCLE⁴; (C) Vergleich der Flächen aller MASCLE-Varianten



Abbildung 6.24 Energieeffiziente garantierte Heilung in Hex-MASCLE mittels Assistance Arrangements, (A-B) mit 3 Zellen; (C-E) mit 4 Zellen; Von links nach rechts sinkend: Assistance Arrangements sortiert nach Energieeffizienz

werden.

Damit die Heilung per gültiger Assistance Arrangement funktioniert, müssen alle benachbarten Zellen einer ausgefallenen Zelle Z dasselbe Assistance Arrangement auswählen. Daher werden die Assistance Arrangements gemäß ihrer Energieeffizienz wie in Abbildung 6.24 dargestellt von links nach rechts mit sinkender Energieeffizienz sortiert und auf jedem Knoten fest gespeichert. Für den Fall, dass mögliche Assistance Arrangements dieselbe Anzahl heilender Zellen, also die gleiche Energieeffizienz besitzen, wurden die Assistance Arrangements genau wie bei 4-MASCLE willkürlich sortiert.

Auch die eigentliche Heilung funktioniert wie bei 4-MASCLE. Nach einem Phasenwechsel und einem Ausbleiben der Life Sign Nachricht einer Zelle Z sendet jeder aktuelle Clusterhead einer Nachbarzelle der Zelle Z einen Broadcast mit einer Possible Assistance Nachricht, wie bereits in Abbildung 6.17 gezeigt wird. Nach dem Empfang jeder anderen Nachbarzelle der Zelle Z ist jedem umliegenden Cluster bekannt, welche umliegenden Cluster der Zelle Z noch verfügbar sind. Jeder Nachbarcluster der Zelle Z wählt daraufhin das am weitesten links befindliche mögliche Assistance Arrangement der in Abbildung 6.24 dargestellten Assistance Arrangements aus.

Sollte kein gültiges Assistance Arrangement mehr möglich sein, bleiben auch hier alle verbliebenen Nachbarzellen der ausgefallenen Zelle Z aktiv, um so zu

versuchen die ausgefallene Zelle mittels eines ungültigen Assistance Arrangements zu heilen.

Die Effizienzsteigerung der Hex-MASCLE Varianten und das Netzwerkverhalten gegenüber einem mit SXGAF oder 4-MASCLE geclustertem Netzwerk wird im nächsten Kapitel gezeigt.

6.3 Vergleich ausgewählter Algorithmen

Um die Performance der entwickelten Algorithmen zu untersuchen, werden nun die entwickelten MASCLE-Algorithmen zum einen mit ihren Basisalgorithmen SXGAF und HXGAF, zum anderen mit dem in Kapitel 2.4.1 beschriebenen CPNS Algorithmus verglichen. Dabei werden die in Kapitel 4.2 erläuterten Vergleichskriterien Realistic Case Coverage, Node Efficiency und Lebenszeit sowohl unter den in Kapitel 4.1.6 beschriebenen idealen als auch realitätsnahen Bedingungen untersucht.

Die Simulationsumgebung stellt sich wie in TABELLE 6.4 beschrieben dar. Da die MASCLE-Algorithmen zum Aufbau einer Torus-Welt eine bestimmte Simulationsfläche benötigen, wurde die Fläche für jeden Algorithmus einzeln angepasst und die Anzahl platzierter Knoten entsprechend angepasst.

Im ersten Schritt wurde in Abbildung 6.25 die Realistic Case Coverage unter idealen Bedingungen untersucht, die bei einer bestimmten Anzahl an platzierten Knoten erreicht wird. Man erkennt, dass der CPNS Algorithmus die wenigsten Knoten benötigt, um immer 100% Netzwerkfunktionalität zu gewährleisten. Grund dafür ist, dass CPNS nur dann einen Knoten abschaltet,

PARAMETER	WERT
Simulierte Fläche	$1000 \ m*1000 \ m^{*1}$
Maximal benötigte Sendereichweite	120 m
Angenommene Sensorreichweite	56 m
Simulierte Netzwerke	250
Kanalmodell	Unit disk graph
Knotenposition	Exakt/Flacher Sandstrand/Trockenes Unterholz
Platzierte Knoten für die Ermittlung der Lebenszeit	10000*1
Lebensdauer eines aktiven Knotens T_{AN}	<i>ca. 91 h</i> ^{*2}

 TABELLE 6.4 SIMULATIONSUMGEBUNG FÜR DEN VERGLEICH DER EIGENSCHAFTEN VON

 verschiedenen positionsbasierten Clusteringalgorithmen

*1: Simulationsfläche und platzierte Knoten für die XGAF und MASCLE Algorithmen entsprechend individuell angepasst, um eine korrekte Torus-Welt zu erreichen

*2: Entspricht einem MICA2-Knoten im permanenten Datenempfangsmodus mit einer Ausgangsenergie von 5200mAh



Abbildung 6.25 Realistic Case Coverage positionsbasierter Clusteringverfahren für ideale Bedingungen über verschiedene Anzahlen platzierter Knoten

wenn er garantiert unwichtig für die Netzwerkfunktionaliät ist, während sich SXGAF und HXGAF darauf verlassen, dass in benachbarten Zellen Knoten aktiv sind.

Sämtliche MASCLE-Algorithmen benötigen zwar mehr Knoten als CPNS, jedoch deutlich weniger als SXGAF oder HXGAF. Dies liegt an der zellbasierten Heilung. Stellt ein MASCLE-Algorithmus fest, dass eine Zelle, die eigentlich aktiv sein sollte, inaktiv ist, so wird das möglicherweise entstehende Loch durch das Aktivieren von Knoten benachbarter Zellen abgefangen. Es wird bei den MASCLE-Algorithmen jedoch nicht geprüft, ob weiter entfernte Nachbarn einer ausgefallenen Zelle ebenfalls für eine Heilung sorgen könnten.

Im nächsten Schritt wurde für verschiedene MASCLE und Vergleichsalgorithmen dieselbe Untersuchung für die realitätsnahen Szenarien Flacher Sandstrand in Abbildung 6.26 sowie Trockenes Unterholz in Abbildung 6.27 durchgeführt.



Es ist zu erkennen, dass die Ungenauigkeit bei der Lokalisierung im Szenario

Abbildung 6.26 Realistic Case Coverage positionsbasierter Clusteringverfahren für das Szenario Flacher Sandstrand über verschiedene Anzahlen platzierter Knoten



Abbildung 6.27 Realistic Case Coverage positionsbasierter Clusteringverfahren für das Szenario Trockenes Unterholz über verschiedene Anzahlen platzierter Knoten

Flacher Sandstrand nur wenig Auswirkung auf die Anzahl benötigter Knoten für eine komplette Netzwerkfunktionalität hat. Anders verhält es sich bei dem Szenario Trockenes Unterholz.

So muss man für den CPNS Algorithmus ca. 10% mehr Knoten platzieren, um die Ungenauigkeit bei der Lokalisierung abzufangen.

Gravierender ist der Einfluss auf die geographischen Clusteringalgorithmen. Sind HXGAF und insbesondere SXGAF noch relativ unbeeinflusst von der ungenauen Lokalisierung, kann beispielsweise bei Hex-MASCLE³ eine komplette Netzwerkfunktionalität nur noch bei rund 85% der Netzwerke erzielt werden. Steigt die Anzahl der platzierten Knoten, scheint der Anteil der Netze mit kompletter Netzwerkfunktionalität sogar wieder zu sinken. Grund für dieses Phänomen lässt sich aus der in Abbildung 6.28 dargestellten Node Efficiency ableiten.

Steigt die Anzahl der platzierten Knoten, reduziert sich bei den XGAF-



Abbildung 6.28 Node Efficiency der positionsbasierten Clusteringalgorithmen über verschiedene Anzahlen platzierter Knoten



Abbildung 6.29 Wahrscheinlichkeit kompletter Netzwerkfunktionalität positionsbasierter Clusteringalgorithmen bei idealen Bedingungen über der Laufzeit des Sensornetzwerks

Algorithmen und CPNS die *Node Efficiency*, bis sie einen minimalen Wert erreicht. Bei den MASCLE-Algorithmen dagegen steigt die *Node Efficiency* ab einer bestimmten Anzahl platzierter Knoten solange wieder, bis sie einen Wert erreicht, der demselben Algorithmus ohne angewendete Selbstheilungsfunktion entspricht.

Ursache dafür ist die Tatsache, dass mit mehr platzierten Knoten auch mehr Zellen besetzt sind, und somit weniger Heilungsvorgänge notwendig sind und durchgeführt werden. Es werden dementsprechend mit ansteigender Zahl platzierter Knoten weniger Knoten im Netzwerk aktiviert. Dies führt mit einer ungenauen Lokalisierung wie im Szenario Trockenes Unterholz natürlich zu einer erhöhten Wahrscheinlichkeit für Löcher im Sensornetzwerk.

Zugleich zeigt die *Node Efficiency* jedoch, dass die Hex-MASCLE Algorithmen im Vergleich mit den anderen Algorithmen die höchstmöglichen Werte erzielen, also am wenigsten Knoten für das Erreichen einer kompletten Netzwerkfunktionalität aktivieren. Dies macht sie zwar anfälliger für Fehler bei der Lokalisierung, kann jedoch auch die Lebenszeit drastisch erhöhen, wie im Folgenden gezeigt wird.

In Abbildung 6.29 ist die durchschnittliche Lebenszeit eines Sensornetzwerks dargestellt, welches sich mit einem entsprechenden Algorithmus clustert. Nach jeder Zeit T_{AN} sterben dabei alle aktiven Knoten, da ihre Energie verbraucht ist, und neue Knoten der jeweiligen Cluster werden aktiviert. Es ist zu erkennen, dass SXGAF und HXGAF mit $T_{life}=10,6T_{AN}$ bzw. $T_{life}=12,8T_{AN}$ mit Abstand die geringste Lebenszeit vorweisen. Schon nach wenigen Zeitschritten tauchen dementsprechend in jedem untersuchten Netzwerk Löcher auf. CPNS, 2-MASCLE und symmetrisches 4-MASCLE erreichen mit $T_{life}=18,6T_{AN}$, $T_{life}=17,2T_{AN}$ bzw. $T_{life}=16,5T_{AN}$ ungefähr dieselbe Lebenszeit, wohingegen die



Abbildung 6.30 Anteil aktiver Knoten zur Gesamtzahl platzierter Knoten positionsbasierter Clusteringalgorithmen über der Laufzeit des Sensornetzwerks

Hex-MASCLE Algorithmen mit $T_{life}=21,8T_{AN}$ für Hex-Mascle³ und $T_{life}=21,3T_{AN}$ für Hex-Mascle⁴ mit Abstand am längsten Netzwerke ohne Löcher in der Sensorabdeckung erhalten können. Für den Fall einer exakten Lokalisierung sind dementsprechend die komplexeren, geographische Cluster bildenden MASCLE-Algorithmen dem zeitlich clusternden CPNS Algorithmen überlegen. In Abbildung 6.29 wird zudem gezeigt, wie sich die Hex-MASCLE Algorithmen verhalten, wenn man die clusterbasierte Selbstheilung abschaltet. Es ist zu erkennen, dass die Algorithmen dann Lebenszeiten aufweisen, die mit $T_{life}=10,1T_{AN}$ für Hex-Mascle³ und $T_{life}=9,0T_{AN}$ für Hex-Mascle⁴ noch unter der Lebenszeit von SXGAF liegen. Grund für die enorme Lebenszeitverlängerung der Algorithmen mit Assistance ist dementsprechend die clusterbasiere Selbstheilung, welche sich nicht nur auf die Lebenszeit, sondern auch auf das Verhalten der aktiven Knoten im Netzwerk auswirkt.

Um dies zu verdeutlichen, ist in Abbildung 6.30 der Anteil der aktiven Knoten von der Gesamtanzahl platzierter Knoten über der Laufzeit des Sensornetzwerks dargestellt. Wie es zu erwarten ist, verhält sich für die XGAF Algorithmen der Anteil der aktiven Knoten so, dass ein bestimmter Anteil von Knoten, der der Anzahl von Clustern im Netzwerk entspricht, aktiv bleibt. Nach mehreren Zyklen ist bereits die Energie vieler Knoten verbraucht und die ersten Zellen beginnen auszufallen. Da es keine Art Überprüfung oder gar Heilung ausgefallener Zellen gibt, werden daher immer weniger Knoten aktiviert. CPNS verhält sich ähnlich. Es wird immer eine bestimmte Menge an im Netzwerk verteilten Knoten aktiviert, um möglichst komplette Netzwerkfunktionalität zu gewährleisten. Erst wenn diese Menge nicht mehr verfügbar ist, kann es zu Löchern in der Sensorabdeckung kommen.



Abbildung 6.31 Wahrscheinlichkeit kompletter Netzwerkfunktionalität positionsbasierter Clusteringalgorithmen im Szenario Flacher Sandstrand über der Laufzeit des Sensornetzwerks

Anders verhalten sich die MASCLE-Algorithmen. Auch hier startet das Netzwerk mit einem Anteil an aktiven Knoten, der der Anzahl an zu aktivierenden Zellen entspricht. Fällt jedoch eine Zelle aus, die eigentlich zu aktivieren war, übernehmen je nach Algorithmus zwei oder mehr benachbarte Zellen die Aufgabe der ausgefallenen Zelle, es kommt also zu einem Anstieg des Anteils aktiver Knoten, welches zu der in Abbildung 6.29 dargestellten Lebenszeitverlängerung führt. Aufgrund der Tatsache, dass nur eine bestimmte Anzahl an Knoten vorhanden ist, erreicht die Anzahl aktivierter Knoten irgendwann ein Maximum, und danach fällt der Anteil aktiver Knoten rapide ab. Zu diesem Zeitpunkt kann jedoch keiner der MASCLE-Algorithmen noch komplette Netzwerkfunktionalität garantieren.

Verzichtet man bei den MASCLE-Algorithmen auf die clusterbasierte Selbstheilung, verhalten sie sich ähnlich den XGAF-Algorithmen, wie ebenfalls in Abbildung 6.30 dargestellt ist. Da aufgrund der geringeren Anzahl aktiver Knoten eine ausgefallene, eigentlich zu aktivierende Zelle mit einer deutlich höheren Wahrscheinlichkeit zu einem Loch in der Sensorabdeckung führt, resultiert dieses Verhalten in der in Abbildung 6.29 dargestellten, verkürzten Lebenszeit.

Um die Lebenszeit der positionsbasierten Clusteringalgorithmen auch unter realitätsnahen Bedingungen zu untersuchen, wurde die Lebenszeit der Algorithmen SXGAF, HXGAF, 2-MASCLE, Hex-MASCLE sowie CPNS unter Lokalisierungsbedingungen untersucht, wie sie in den Szenarien Flacher Sandstrand und Trockenes Unterholz vorliegen. Die Ergebnisse dieser Untersuchungen sind in Abbildung 6.31 und Abbildung 6.32 dargestellt.



Abbildung 6.32 Wahrscheinlichkeit kompletter Netzwerkfunktionalität positionsbasierter Clusteringalgorithmen im Szenario Trockenes Unterholz über die Laufzeit des Sensornetzwerks

In Abbildung 6.31 ist zu erkennen, dass die Ungenauigkeit, die bei einer Lokalisierung im Szenario Flacher Sandstrand entsteht, kaum Einfluss auf die Lebenszeit der entsprechenden Algorithmen besitzt. Lediglich Hex-MASCLE³ erreicht nur noch in maximal 95% der Fälle komplette Netzwerkfunktionalität, was zu einer reduzierten Lebenszeit von $T_{life}=21,0T_{AN}$ führt. Anders sieht es für das Szenario Trockenes Unterholz aus. Während sich die Lebenszeit der Algorithmen SXGAF, HXGAF, 2-MASCLE sowie CPNS immer noch mindestens 99% der Lebenszeit unter idealen Bedingungen beträgt, können die Hex-MASCLE Algorithmen nicht mehr in jedem Fall komplette Netzwerkfunktionalität gewährleisten. Insbesondere Hex-MASCLE³ erreicht nur noch in ca. 80% der Fälle komplette Netzwerkfunktionalität, was mit einer Lebenszeit von $T_{life} = 18,0T_{AN}$ zu einem Verlust der Lebenszeit von 17,5% verglichen mit exakter Lokalisierung führt. Gründe für die extreme Verschlechterung von Hex-MASCLE³ bei einer ungenauen Lokalisierung finden sich in der schon in Abbildung 6.23 äußerst geringen Assisted Area. Regulär aktive Zellen decken nur einen Bruchteil der Working Area ihrer Nachbarn garantiert ab, sodass eine ungünstige Lokalisierung häufig zu einem Loch in der Sensorabdeckung führt. Hex-MASCLE⁴ wird mit einer Lebenszeitreduzierung von 7% zwar auch von der schlechten Lokalisierung beeinflusst, liegt aber mit einer verbleibenden Lebenszeit von $T_{life} = 19,8T_{AN}$ immer noch 6,5% vor CPNS.

6.4 Zwischenfazit

Das vorliegende Kapitel beschreibt detailliert verschiedene Varianten der entwickelten lokalisierungsbasierten Clusteringalgorithmen XGAF und MASCLE und vergleicht ausgewählte Algorithmen in verschiedenen Szenarien mit ausgewählten Konkurrenzalgorithmen der Literatur (CPNS sowie SXGAF als Anwendung von GAF auf Sensornetzwerke)

Zusammenfassend lässt sich erkennen, dass im Fall einer exakten zur Verfügung stehenden Lokalisierung eines der Hex-MASCLE Verfahren verwendet werden sollte, um möglichst hohe Lebenszeit zu erreichen.

Für eine Lokalisierung mit im Vergleich zur Sendereichweite hohem Fehler von $\sigma_{Position} > 0,06R_T$, wie er im Szenario Trockenes Unterholz auftritt, sollte eher auf 2-MASCLE oder CPNS zurückgegriffen werden. Zwar bietet für eine fehlerhafte Lokalisierung Hex-MASCLE immer noch eine geringfügig höhere Lebenszeit, es kann jedoch nicht mehr in jedem Fall komplette Netzwerkfunktionalität gewährleistet werden.

Sollte dementsprechend ein Clusteringalgorithmus für ein Sensornetzwerk mit ungenauer Lokalisierung gewählt werden, muss beachtet werden, dass es sich bei 2-MASCLE im Gegensatz zu CPNS um ein geographisches Clusteringverfahren handelt. Damit kommt 2-MASCLE mit zusätzlichen Vorteilen einher. So können innerhalb eines Clusters Knoten ausgetauscht oder die ermittelten Clusterstärken für Routingverfahren ausgenutzt werden, wie es für ein Fallbeispiel in [38] vorgestellt wird.

7 Lokalisierungsfreies Clustering

Für drahtlose Sensornetzwerke, deren Komponenten in unzugängliche, nicht durch GPS erreichbare Gebiete ausgebracht werden, besteht häufig nicht die Möglichkeit, Sensorknoten anhand von Referenzknoten zu lokalisieren. Aber auch in Indoor-Szenarien oder Sensornetzwerken mit beweglichen Knoten kann eine Lokalisierung möglicherweise nur eingeschränkt oder gar nicht möglich sein.

In drahtlosen Sensornetzwerken, in denen Knoten ihre eigene Position nicht oder nur eingeschränkt kennen und nicht abschätzen können, ist Clustering dennoch ein notwendiges und energieeffizientes Verfahren, mit welchem die Lebenszeit des Netzwerks vervielfacht werden kann, ohne die Sensornetzwerkfunktionalität einzuschränken. Das vorliegende Kapitel dient dazu, die im Zuge dieser Arbeit entwickelten Clusteringalgorithmen vorzustellen und zu analysieren, welche nicht auf einer vorher ermittelten Position der Knoten basieren, und somit unabhängig von der Genauigkeit eines vorhergehenden Lokalisierungsverfahrens sind.

Während in dem in Kapitel 7.1 vorgestellten speziellen Verfahren regelmäßig angeordnete Beaconknoten und deren Broadcasts dazu dienen, das zu überwachende Gebiet in SXGAF ähnliche Zonen aufzuteilen, wird in Kapitel 7.2 und 7.3 eine angepasste Sendereichweite dazu verwendet, um die Clusterbegrenzen zu bestimmen. Kapitel 7.4 vergleicht die entwickelten reichweitenbasierten Verfahren untereinander sowie mit anderen lokalisierungsfreien und positionsbasierten Clustering-Verfahren.

7.1 Beacon Based Clustering

In den Arbeiten [60, 63, 66] gehen die Autoren jeweils von regelmäßig angeordneten Beacons aus, wobei 4 Beacons jeweils die Eckpunkte eines Quadrats darstellen. Wird in einem solchen Szenario der CL-Lokalisierungsalgorithmus ausgeführt, stellt man fest, dass es innerhalb eines solchen Quadrates nur wenige verschiedene Punkte gibt, auf die sich ein unbekannter Knoten lokalisieren kann Dabei lokalisieren sich Knoten eines bestimmten Areals auf dieselbe Position, wie Abbildung 7.1 (A) und Abbildung 7.1 (B) zeigen. Die entstehenden Areale werden dabei als *Bean Cell, Edge Cell, Center Cell* sowie *Corner Cell* bezeichnet, deren Form und Größe in Abhängigkeit der Sendereichweite der Beacons R_{TB} erheblich schwankt. Beacon Based Clustering (kurz: BBC) [74] ermöglicht es nun, diese Gebiete optimal als XGAF-Zellen zu verwenden.



Abbildung 7.1 Zellbildung und Lokalisierung mittels reichweitenbegrenzter Nachrichten regelmäßig angeordneter Beacons; (A) $R_{TB}=0.8d_B$; (B) $R_{TB}=0.88d_B$; (C) Vereinigung von Bean Cells und Center Cell zur Star Cell und Ähnlichkeit mit XGAF bei entsprechendem Beaconabstand d_B ; (D) Gewählte Zell- und Beaconnummerierung

Um dies zu erreichen, wurden die entstehenden Areale hinsichtlich Form und Größe analysiert. Die Ergebnisse dieser Analyse sind im Folgenden aufgeführt.

7.1.1 Aufteilung der Areale

Im Gegensatz zu regulären XGAF Zellen besitzen die durch Beaconnachrichten ermittelten Areale unterschiedliche Form und Größe. Wie bereits erwähnt, sind sowohl Form als auch Größe der Areale dabei von der eingestellten Sendereichweite der Beacons abhängig. Die Größen der einzelnen Zellen in Abhängigkeit von R_{TB} sind in Abbildung 7.2 dargestellt. Da es für ein effizientes XGAF Clustering nötig ist, dass alle Zellen eine ähnliche Größe besitzen, bietet es sich an, die vornehmlich kleineren Areale (Bean Cell und Center Cell) zur einem größeren Areal, der sogenannten Star Cell zusammenzufassen, wie bereits Abbildung 7.1 (C) zeigt.



Abbildung 7.2 Größe der entstehenden Zellen im Verhältnis zur Fläche, die von 4 Beacons aufgespannt wird, bei verschiedenen Sendereichweiten der Beacons.

7.1.2 Beaconabstand

Damit sich die regelmäßige XGAF-Struktur auf der durch Beacons aufgespannte Zellstruktur abbilden lässt, ist es unerlässlich, dass der Beaconabstand ein Vielfaches der XGAF-Zellgröße ist. Wird der Beaconabstand d_B zu benachbarten Beacons gemäß Gleichung (7.1) gewählt, entspricht jede enstehende Zelle exakt einer XGAF Zelle, wie Abbildung 7.1 (C) zeigt.

$$d_B = 2 * L_{max,SXGAF} = R_W * \sqrt{2} \tag{7.1}$$

Mit dieser gewählten Beacondistanz entspricht jede entstehende Zelle ungefähr den Dimensionen einer XGAF Zelle, wobei sich im Mittelpunkt jeder vierten XGAF-Zelle ein Beacon befindet. Um jeden Knoten zu seiner richtigen Zelle zuzuordnen, sendet jeder Beacon eine Beaconnachricht aus, welche die in Abbildung 7.1 (C) gewählte Beaconnummerierung enthält. diesen Mit Informationen kann jeder Knoten anhand der empfangenen N Beaconnachrichten die Nummerierung der ihm zugeordneten Zelle ausrechnen. Dies geschieht anhand Gleichung (7.2).

$$C[x, y] = \left[\frac{\sum_{i=1}^{N} B[x, y]}{N} + 0.5\right]$$
(7.2)

7.1.3 Minimale Sendereichweite der Sensorknoten

Eine der grundlegenden Voraussetzungen in XGAF ist es, das ein beliebiger Sensorknoten einer Zelle mit einem beliebigen Knoten einer Nachbarzelle kommunizieren kann. Um dies zu garantieren, dürfen die am Weitesten voneinander entfernten Punkte zweier benachbarter Zellen nicht weiter als die Sendereichweite entfernt liegen. Je nach Funkreichweite der Beacons besitzen die entstehenden Zellen jedoch unterschiedliche Form und Größe, sodass die benötigte Sendereichweite nicht immer konstant ist. Die benötigte Sendereichweite im Verhältnis zum Abstand zweier benachbarter Beacons ist in Abbildung 7.3 dargestellt. In der Grafik ist zu erkennen, dass sobald eine garantierte Kommunikation diagonal benachbarter Zellen erforderlich ist, also eine Kommunikation von Edge Cell zu Edge Cell und Corner Cell zu Star Cell erfolgen soll, die benötigte Sendereichweite $R_{T,Diag}$ nicht von der Sendereichweite der Bacons abhängt, sondern Gleichung (7.3) folgt:

$$R_{T,Diag} = \sqrt{2} * d_B = 2 * R_W = R_{T,SXGAF,Diag}$$
(7.3)

Die minimale zu garantierende Sendereichweite muss für eine garantierte Kommunikation mit allen 8 Nachbarn also ebenso groß sein wie die Sendereichweite von SXGAF mit 8 garantierten Nachbarn $R_{T,SXGAF,Diag}$, obwohl bei BBC unterschiedliche Zellformen vorliegen.

Wählt man eine Strategie, bei der nur orthogonal benachbarte Zellen miteinander kommunizieren sollen, genügt es, eine Edge Cell zu Corner Cell sowie Edge Cell zu Star Cell Kommunikation zu garantieren. In diesem Fall hängt die minimal mögliche garantiere Kommunikationsreichweite direkt von der Sendereichweite der Beacons ab. Um eine minimale benötigte Sendereichweite $R_{T,Ortho}$ zu erhalten, sollte daher die Sendereichweite der Beacons R_{TB} wie in Gleichung (7.4) so eingestellt werden, dass die benötigte Sendereichweite Edge Cell zu Corner Cell sowie Edge Cell zu Star Cell identisch sind.

$$F \ddot{u}r R_{TB} = 0.77 d_B gilt:$$

$$R_{T,Ortho} = 1.15 * d_B = 1.63 * R_W = 1.03 R_{T,SXGAF,Ortho}$$
(7.4)

Wie man anhand Gleichung (7.4) erkennt, führt hier die unregelmäßige Zellform dazu, dass eine unbedeutend höhere Sendereichweite benötigt wird als bei flächenoptimiertem SXGAF.

7.1.4 Optimale Sendereichweite der Beacons

Um eine möglichst geringe Anzahl an Sensorknoten platzieren zu müssen, und dennoch eine komplette Sensorabdeckung des zu überwachenden Gebiets zu gewährleisten, ist es sinnvoll, eine möglichst optimale Sendereichweite der Beacons einzustellen. Dabei gibt es zwei untersuchte Optimierungsansätze.



Abbildung 7.3 Maximale Ausdehnung einer Zelle bei unterschiedlichen Sendereichweiten der Beacons

Zum einen wurde die Strategie untersucht, möglichst jede entstehende Zelle auf dieselbe Größe zu bringen, um so bei der Ausbringung der Knoten zu erreichen, dass möglichst jede Zelle mit derselben Wahrscheinlichkeit besetzt wird. Die Untersuchung der Flächen bei unterschiedlicher Sendereichweite der Beacons ist bereits in Abbildung 7.2 dargestellt. Man erkennt, dass bei $R_{TB}=0,805d_B$ Star Cell und Corner Cell dieselbe Größe haben, diese Reichweite also optimal erscheint.

Zum anderen gibt es die Strategie, eine möglichst geringe maximale Ausdehnung jeder entstehenden Zelle zu erhalten. Somit ist es möglich, keine unnötig große Sensorreichweite verwenden zu müssen, um die Abdeckung einer Zelle mit einem beliebigen aktiven Knoten innerhalb dieser Zelle garantieren zu können. Die maximale Ausdehnung $R_{Cell,Max}$ innerhalb jeder Zelle ist für verschiedene Beaconreichweiten in Abbildung 7.3 dargestellt.

Man erkennt, dass für $R_{TB}=0.800d_B$ Star Cell und Edge Cell eine maximale Ausdehnung von $R_{Cell,Max}=0.75d_B$ besitzen, sodass diese Sendereichweite der Beacons optimal erscheint. SXGAF Zellen besitzen dank ihrer Regularität aber

PARAMETER	WERT
Simulierte Fläche	1000 m*1000 m
Evaluierte Fläche	1000 m*1000 m (Donutwelt)
Sendereichweite	14,4 m
Sensorreichweite (Für Coverage)	70,7 m
Simulierte Netzwerke	1000
Beaconabstand d_B	100 m
Platzierte Beacons	100
Kanalausbreitung	Unit disk graph
Knotenposition	Exakt

TABELLE 7.1 SIMULATIONSUMGEBUNG FÜR DEN VERGLEICH DER EIGENSCHAFTEN VON BBC und SXGAF Zellen unter idealen Bedingungen



Abbildung 7.4 Anteil besetzter BBC- und SXGAF-Zellen bei unterschiedlicher Anzahl platzierter Sensorknoten

auch hier einen Vorteil, da ihre maximale Ausdehnung in demselben Szenario nur $R_{SXGAF,Max}=0.707d_B$ beträgt.

Um die Performance mit SXGAF vergleichen zu können, wurde die in TABEL-LE 7.1 beschriebene Simulationsumgebung geschaffen. Im ersten Schritt wurde dabei untersucht, wie sich die eingestellte Sendereichweite der Beacons und damit die unterschiedliche Größe der einzelnen Zellen auf die Anzahl besetzter Zellen und damit aktiver Knoten im Netzwerk auswirkt. Das Ergebnis ist in Abbildung 7.4 dargestellt. Man sieht, dass für $0.8d_B \le R_{TB} \le 0.82d_B$ der größte Anteil vorhandener Zellen besetzt ist, da bei dieser Sendereichweite der Beacons keine Zellsorte besonders klein ist.

Weiterhin wurde untersucht, inwiefern sich das Verändern der Sendereichweite der Beacons auf die gesamte Sensorabdeckung auswirkt. Das Ergebnis ist in Abbildung 7.5 dargestellt. Auch hier erkennt man eine signifikante Korrelation. Für eine möglichst wahrscheinliche komplette Funktionalität eines Sensornetzes bietet sich daher eine optimale Reichweite der Beacons von $0.8d_B \le R_{TB,Opt} \le 0.83d_B$ an. Vergleicht man detailliert den Anteil komplett abgedeckter Sensornetze und damit die Netzwerkfunktionalität für $R_{TB,Opt} = 0.8d_B$ für



Abbildung 7.5 Anteil komplett abgedeckter Sensornetzwerke mit BBC- oder SXGAF -Clustering für unterschiedlicher Anzahl platzierter Sensorknoten



Abbildung 7.6 Anteil der Netzwerke mit 100% Netzwerkfunktionalität von BBC und SXGAF bei verschiedenen Anzahlen platzierter Knoten.

verschiedene Anzahlen platzierter Sensorknoten, wie in Abbildung 7.6 dargestellt, so erkennt man, dass für diesen speziellen Fall mit BBC tatsächlich die Performance von XGAF erreichbar ist.

Mit BBC erhält man also eine auf der Idee von XGAF basierte Clusteringvariante, bei der eine exakte Lokalisierung unerheblich wird. Jedoch müssen die Beacons einerseits regulär angeordnet sein, anderseits auch auf die Working Range R_W des Sensornetzwerks eingestellt sein. Ein flexibler, später angepasster und dennoch effizienter Netzbetrieb ist somit unter Umständen nicht möglich.

7.2 Free-GAF

Mit BBC ist es nun möglich, eine von einer exakten Lokalisierung unabhängige Zuordnung zu Zellen vorzunehmen, deren Größe, Form und Anordnung in etwa denen von SXGAF entspricht. Jedoch benötigt BBC regelmäßig und im richtigen Abstand angeordnete Beacons und exakte Sendereichweiten, sodass ein Clustering eines beliebigen Sensornetzwerks mittels BBC nicht ohne Weiteres möglich ist. In [75] wurde daher im Zuge dieser Arbeit der Algorithmus *Localization Free – Geographic Adaptive Fidelity* (kurz: Free-GAF) entwickelt, mit welchem ein XGAF-ähnliches Clustering erfolgen kann, ohne auf vorab platzierte Beacons oder eine andere Art der Lokalisierung angewiesen zu sein. Die Idee von Free-GAF ist es dabei, ähnlich wie bei BBC die Sendereichweite der Sensorlmeten auszuputzen um die Gronzen der Chuster fosterulegen.

der Sensorknoten auszunutzen, um die Grenzen der Cluster festzulegen. Die spätere Form eines Clusters ist dabei für Free-GAF nicht entscheidend. Lediglich die maximale Ausdehnung eines entstehenden Clusters, die die Working Range R_W nicht überschreiten darf, ist von Bedeutung.

Um Free-GAF Cluster zu erzeugen, werden auf jedem Knoten die Zustände *Unclustered Node, Possible Clusterhead, Clusterhead* und *Clustermember* definiert. Nach der Platzierung der Sensorknoten starten alle Knoten als Unclustered Node und sind im Empfangsmodus. Der eigentliche Clustering-Algorithmus beginnt dabei auf einem Basisknoten, wie in Abbildung 7.7 (A) dargestellt ist. Der Basisknoten ist als *erster Clusterhead* definiert, auf welchem dann der folgende dreistufige Algorithmus abläuft:

7.2.1 Informationsphase

In dieser Phase hat ein Clusterhead die Aufgabe, alle benachbarten Knoten mit einer maximalen Entfernung von R_W über seine Existenz zu informieren, sodass die benachbarten Knoten erfahren, dass ein Cluster entsteht. Dies geschieht mit einem Broadcast einer sogenannten *Possible Clusterhead*-Nachricht mit einer angepassten Sendereichweite von $R_{PC,FG} = R_W$ und einer eindeutigen Cluster-ID als Inhalt, wie in Abbildung 7.7 (B) dargestellt ist.

Alle Empfänger dieser Nachricht, welche sich im Zustand Unclustered Node befinden, wechseln ihren Zustand zu dem Zustand Possible Clusterhead. Sie werden also möglicherweise die nachfolgenden Clusterheads und können hier bereits den sendenden Clusterhead als Nachbarcluster speichern. Zusätzlich setzen die empfangenden Knoten einen internen Timer (Clusterhead-Timer T_{CH}) auf die Bezugszeit T_B . Diese Bezugszeit repräsentiert, nach welcher Zeit ein Possible Clusterhead im frühesten Fall zu einem Clusterhead wechseln dürfte, und kann beispielsweise 1 Sekunde betragen.

Weiterhin speichert ein Clusterhead eine Variable *Waiting Cycle* (kurz: C_W) mit dem Wert $C_W=1$ ab. Diese Variable wird im Laufe des Clusteringprozess mehrfach verändert und stellt dar, nach wie vielen Zyklen des Clusterhead-Timers der Knoten von dem Zustand Possible Clusterhead automatisch in den Zustand Clusterhead wechselt.

7.2.2 Clusteringphase

In dieser Phase zwingt ein Clusterhead seine geographisch nächsten Knoten, seinem Cluster beizutreten. Um das zu erreichen, sendet ein Clusterhead direkt nach der Possible Clusterhead Nachricht einen zweiten Broadcast, die sogenannte *Join Cluster* Nachricht mit der Reichweite $R_{JC,FG}=R_W/2$. Alle Knoten, die diese Nachricht empfangen, und selber nicht bereits Clustermember oder Clusterhead sind, wechseln in den Zustand Clustermember und ordnen sich dem sendenden Clusterhead zu. Dies ist in Abbildung 7.7 (C) dargestellt.

7.2.3 Bestätigungsphase

In dieser Phase bestätigen alle gerade in den Zustand Clustermember gewechselten Knoten mit einem *Cluster Affirmation* Broadcast, dass sie dem Cluster beitreten. Da der Clusterhead sich maximal in einer Entfernung $R_W/2$ befindet, genügt es, wenn der Broadcast die Reichweite $R_{CA,FG} = R_W/2$ besitzt. Der Bestätigungs-Broadcast dient dem Algorithmus in zweierlei Hinsicht.

Zum einen erhält der Clusterhead Informationen über die Anzahl Knoten in seinem Cluster, und kann damit die Stärke seines Clusters bestimmen, welches für ein späteres Routing von Bedeutung sein kann [38].

Zum anderen kann der Broadcast genutzt werden, um die Auswahl weiterer Clusterheads zu beeinflussen. Dafür sieht der Algorithmus vor, dass jeder Possible Clusterhead, welcher einen Cluster Affirmation Broadcast empfängt, seinen Waiting Cycle um einen Wert t_{add} erhöht. Hierbei ist beabsichtigt, dass Possible Clusterhead Knoten, welche nahe dem Cluster liegen, nach Beendigung der Bestätigungsphase einen höheren Wert in ihrem Waiting Cycle vorliegen haben als weiter entfernte Possible Clusterhead Knoten. Die Bestätigungsphase für den ersten Cluster ist in Abbildung 7.7 (D) dargestellt.

7.2.4 Anwendung im Sensornetzwerk

Nach diesen drei Phasen existiert also ein erster Cluster und ihn umgebende Knoten im Zustand Possible Clusterhead mit unterschiedlichen Werten C_w. Damit der Algorithmus sich durch das gesamte Sensornetzwerk arbeitet, geschieht Folgendes: Jeder Clusterhead-Timer reduziert seinen Wert C_W nach Ablauf des Clusterhead-Timers um ein Decrement Interval tred. Für Free-GAF hat sich experimentell ein Decrement Interval von $t_{red}=1$ als günstig erwiesen. Der Clusterhead-Timer wird anschließend wieder auf die Bezugszeit T_B gesetzt. Sobald der Waiting-Cycle eines Possible Clusterhead Knotens auf 0 reduziert wurde, wechselt er selbst in den Zustand Clusterhead und arbeitet die drei Phasen nacheinander ab, wie in Abbildung 7.7 (E-G) dargestellt ist. Bei genügend hoher Knotendichte ist somit dafür dass sich gesorgt, der Clusteringalgorithmus durch das gesamte Netzwerk arbeitet. Für eine hohe Geschwindigkeit bis zur Beendigung des Algorithmus ist dabei eine möglichst geringe Bezugszeit T_B zu wählen.

Sobald sich der Algorithmus durch das Netzwerk gearbeitet hat, liegt ein geclustertes Netzwerk vor, bei dem kein einziger Cluster eine Ausdehnung größer R_W besitzt, wie Abbildung 7.7 (H) zeigt. Für den laufenden Betrieb geht man nach dem erfolgten Clustering genau wie bei XGAF davon aus, dass ein einzelner beliebiger Knoten pro entstandenem Cluster ausreicht, um die komplette Netzwerkfunktionalität aufrechtzuerhalten, während alle anderen Sensorknoten in den Schlafmodus wechseln.



Abbildung 7.7 Clustering eines Sensornetzwerkes mittels Free-GAF (Initialwert $C_W=1$, $t_{Add}=1$, $t_{red}=1$), (A) Ausgangssituation, (B) Broadcast der Possible Clusterhead Nachricht, (C) Broadcast der Join Cluster Nachricht, (D) Broadcasts der Cluster Affirmation Nachrichten, (E-G) Zweiter Clusterhead wiederholt die Prozedur, (H) Fertig geclustertes Netzwerk

Welcher Clusterknoten die Aufgabe übernimmt und wann der Wechsel des aktiven Knotens erfolgt, entscheidet auch hier jeder Cluster autonom, ohne Bezug auf seine Nachbarn zu nehmen.

7.3 Free-CLASH

Nachdem mit Free-GAF ein erster lokalisierungsfreier Clusteringalgorithmus in dieser Arbeit entwickelt wurde, wurde während der Simulationen festgestellt, dass der Algorithmus zwei entscheidende Schwachpunkte besitzt.

Zum einen existiert in Free-GAF die Möglichkeit, dass Flächen im zu überwachenden Gebiet entstehen, die keinem Cluster zugeordnet sind und somit möglicherweise nicht von einer Sensorabdeckung profitieren. Zum anderen gibt es bei Free-GAF keinerlei Kontrolle über die Clusterform und kein Bestreben nach Regularität. Durch dieses fehlende Bestreben werden bei Free-GAF deutlich mehr Cluster als beispielsweise bei SXGAF erzeugt.

Um die beiden Nachteile zu beseitigen, wurde im Zuge dieser Arbeit der Clusteringalgorithmus Free-CLASH (*engl. Localization-Free Clustering with Approximation to Symmetric Hexagons*) [76] entwickelt. Basierend auf Free-GAF beseitigt der Algorithmus die Nachteile mittels der folgenden vier Modifikationen:

7.3.1 Anpassen der Reichweite des Possible Clusterhead Broadcast

Eine der Möglichkeiten, nach einem erfolgten Free-GAF Löcher in der Sensorabdeckung zu erhalten, entsteht, sobald drei miteinander benachbarte Clusterheads während des Clustering-Prozesses maximale Distanz zueinander haben, wie Abbildung 7.8 (A-B) zeigt, und es zu einer ungünstigen Auswahl an aktiven Knoten innerhalb der Cluster kommt, wie beispielhaft Abbildung 7.8 (C) zeigt. Um dieses Verhalten zu verhindern, sieht Free-CLASH vor, den maximalen Abstand von drei miteinander benachbarten Clusterheads $R_{PC,FC}$ wie in Gleichung (7.5) und Abbildung 7.8 (D-E) gezeigt zu reduzieren, indem die Reichweite des Possible Clusterhead Broadcasts verringert wird.

$$R_{PC,FC} = \sqrt{3}/2 * R_W \tag{7.5}$$

Durch die somit angepasste Reichweite ist es nicht mehr möglich, dass Löcher in der Sensorabdeckung entstehen, wie Abbildung 7.8 (F) für die ungünstige Kombination aus benachbarten aktiven Knoten zeigt.



Abbildung 7.8 Entstehung eines Lochs in der Sensorabdeckung zwischen drei benachbarten Clustern, (A) Ausgangssituation, (B) Entstehung von drei benachbarten Clustern mit maximaler Distanz, (C) ungünstige Auswahl aktiver Knoten in jedem Cluster, (D-F) Clusterentwicklung und Sensorabdeckung von 3 benachbarten Clustern mit angepasster Possible Clusterhead-Broadcast Reichweite

7.3.2 Selektive Auswahl neuer Clusterheads

Ein weiterer Nachteil von Free-GAF ist es, dass ausgewählte Clusterheads bei Free-GAF nur bedingt Bezug auf bisher existierende Clusterheads nehmen. So wird ein Clusterhead zwar eher gewählt, wenn er möglichst wenige *Join Cluster* Nachrichten erhält, also möglichst weit vom Ursprungscluster entfernt ist, jedoch fehlt das Bestreben, mit möglichst vielen anderen Clusterheads verbunden zu sein. Dadurch kann das in Abbildung 7.9 (A-B) dargestellte Verhalten entstehen, welches zu Löchern im Sensornetzwerk führen kann, wie Abbildung 7.9 (C) veranschaulicht.

In Free-CLASH wurde daher die Lösung entwickelt, dass ein Clusterhead nur dann entstehen darf, nachdem er mindestens zwei Possible Clusterhead-Nachrichten empfangen hat, wie in Abbildung 7.9 (D) dargestellt ist. Sollte ein Knoten im Zustand Unclustered Node also nur einen Possible Clusterhead-Broadcast empfangen, wird der Knoten zuerst in den hier als Possible Clusterhead Candidate bezeichneten Zustand versetzt. Erst durch den Erhalt einer zweiten Possible Clusterhead-Nachricht erlangt der Knoten den Zustand Possible Clusterhead. Mittels dieser selektiven Auswahl neuer Clusterheads wird das Entstehen eines stärker miteinander verbundenen Clusterstruktur erreicht. die. kombiniert mit der verringerten Possible Clusterhead-Nachrichtenreichweite, keine Löcher in der Sensorabdeckung enthält, wie Abbildung 7.9 (D-F) veranschaulicht.

Da für die Entstehung eines neuen Clusterheads in Free-CLASH nun jedoch zwei bereits vorhandene Clusterheads von Nöten sind, kann die Strategie entweder erst ab dem zweiten zusätzlich entstehenden Clusterhead angewendet



Abbildung 7.9 Nicht-selektive versus selektive Clusterheadwahl, (A) Ausgangssituation, (B) Entstehung von zwei weiteren Clusterheads, (C) Loch in der Sensorabdeckung durch ungünstige Wahl aktiver Knoten, (D) Selektion der Possible Clusterheads, (E-F) Entstehung von eng vermaschten Clustern ohne Möglichkeit für Löcher in der Sensorabdeckung

werden oder man beginnt den Algorithmus bereits mit zwei vorhandenen Clusterheads.

7.3.3 Beschleunigung der Clusterheadwahl für eine gleichmäßige Clusterausbreitung

Die selektive Auswahl eines Clusterheads sorgt dafür, dass jeder entstehende Clusterhead durch mindestens zwei bisher vorhandene Cluster erzeugt wird, und sich somit eine gewisse Regelmäßigkeit einstellt. Trotzdem kann es während der Entstehung neuer Cluster durch unterschiedliche Knotendichten einerseits dazu kommen, dass ein Possible Clusterhead dank mehrerer bereits vorhandener Nachbarcluster viele Join Cluster-Broadcast empfängt und somit eine sehr lange Wartezeit T_{CH} bekommt, bevor er neuer Clusterhead werden darf. Andererseits können ungünstige Knotenverteilungen dafür sorgen, dass Knoten, die eigentlich relativ günstig für die Wahl zum Clusterhead liegen, sehr spät ihre zweite Possible Clusterhead-Nachricht empfangen.

In beiden Fällen kann es passieren, dass ein anderer Knoten zuerst Clusterhead wird, und den eigentlich günstiger liegenden Knoten dazu zwingt, seinem Cluster beizutreten. Dieses Beispiel ist in Abbildung 7.10 (A-C) zu sehen.

Als Lösung wird in Free-CLASH die Abhängigkeit des Wertes t_{red} von der Anzahl an empfangenen Possible Clusterhead Nachrichten N_{PC} vorgesehen. In den durchgeführten Simulationen hat sich eine quadratische Abhängigkeit wie



Abbildung 7.10 Ausbreitung der Cluster bei Free-GAF/Free-CLASH mit $C_W=1$, $t_{add}=1$, (A) Ausgangssituation, (B-C) Enstehung eines Lochs in der Sensorabdeckung durch einen zu hohen T_{CH} -Wert, (D-E) Beschleunigung von Possible Clusterheads mit mehereren Nachbarn, (F) Clusterausbreitung ohne Loch in der Sensorabdeckung

in Gleichung (7.6) dargestellt als günstig erwiesen.

$$t_{red} = (N_{PC}^2 - 1)/3 \tag{7.6}$$

Man erkennt, dass sich für $N_{PC}=1$ mit $t_{red}=0$ der Wert eines Clusterhead-Timers gar nicht reduziert. Sobald also nur eine Possible Clusterhead-Nachricht empfangen wurde, kann ein Knoten kein Clusterhead werden, was der Idee der Possible Clusterhead Candidates entspricht. Zwei empfangene Possible Clusterhead Nachrichten sorgen dafür, dass ein Clusterhead mit $t_{red}=0,33$ frühestens nach drei Perioden entstehen kann. Drei empfangene Possible Clusterhead-Nachrichten beschleunigen die Entstehung bereits um Faktor drei, während vier empfangene Clusterheads die Entstehung um Faktor acht beschleunigen.

Die Beschleunigung der Entstehung von Clusterheads, welche bereits über mehrere Nachbarn verfügen, verhindert nun zumeist die Entstehung von Löchern durch eine ungleichmäßige Ausbreitung von Clustern, wie Abbildung 7.10 (D-F) zeigt. Als Nachteil ermöglicht die Beschleunigung von Clusterheads mit mehreren Nachbarn jedoch die Herausbildung von *Cross Couplings*, deren Bedeutung, Entstehung und Vermeidung im nächsten Abschnitt beschrieben wird.

7.3.4 Vermeiden von Cross-Couplings

Die drei bisher vorgestellten Modifikationen an Free-GAF sorgen dafür, dass die entstehende Clusterstruktur regelmäßiger wird und sich gleichmäßig ausbreitet, jedoch auch dichter wird. Das ist zwar zum einen notwendig, um potenpotentielle Löcher bei der Sensorabdeckung im Netzwerk zu vermeiden, zum anderen sorgt eine erhöhte Clusteranzahl jedoch auch für mehr aktive Knoten im Netzwerk und führt damit zu einer verringerten Lebensdauer. Eine Möglichkeit, die Clusterdichte zu verringern, ohne die Sensorabdeckung zu gefährden, besteht darin, hier als *Cross-Coupling* bezeichnete Clusterstrukturen aufzudecken und zu vermeiden. Als Cross-Coupling werden Strukturen bezeichnet, bei denen 4 Clusterheads nach der Entstehung ihrer Cluster maximal eine Distanz von $R_{PC,FC}$ besitzen, also jeder der 4 Cluster miteinander benachbart ist. Das Entstehen einer solchen Struktur ist in Abbildung 7.11 (A-B) dargestellt.

Cross-Couplings lassen sich effizient vermeiden, indem in jeder Possible Clusterhead-Broadcast die IDs der bereits vorhandenen Nachbarn enthalten sind. Ein Knoten, der mehrere Possible Clusterhead-Nachrichten empfängt, kann daraufhin analysieren, ob er nach dem Wechsel zum Clusterhead drei Nachbarn hätte, die ebenfalls alle drei miteinander benachbart sind.



Abbildung 7.11 Entstehung und Vermeidung von Cross-Couplings, (A) Beschleunigte Auswahl eines nahen Possible Clusterheads durch drei benachbarte Cluster, (B) Entstehung eines Cross Couplings, (C) Verzögerung von Possible Clusterheads dessen Nachbarn alles miteinander benachbart sind, (D) Resultierende Clusterstruktur ohne Cross-Coupling

Sollte das der Fall sein, kann ein Possible Clusterhead seinen Wert t_{red} weiter verändern, um die Zustandsänderung zum Clusterhead zu verzögern, anstatt zu beschleunigen. Als praktikabel hat sich in den Simulationen die Anpassung wie in Gleichung (7.7) beschrieben erwiesen.

$$t_{red} = \begin{cases} (N_{PC}^2 - 1)/3; & Cross \ Coupling = Falsch\\ 0,25; & Cross \ Coupling = Wahr \end{cases}$$
(7.7)

Der somit verzögerte Ablauf des Clusterhead-Timers sorgt dafür, dass sich, bei ausreichend vorhandenen Knoten, eine HXGAF ähnliche Struktur entwickelt, ein Cluster also final zumeist sechs direkte Nachbarn hat, wie in Abbildung 7.11 (C-D) dargestellt ist.

7.3.5 Anwendung im Sensornetzwerk

Trotz der Modifikationen von Free-GAF verändert sich die Grundstruktur des Algorithmus nicht, sondern läuft wie im in Abbildung 7.12 dargestellten Flussdiagramm ab. Der grundlegende Unterschied ist also durch die Ermittlung neuer Clusterheads gegeben, wie die Flussdiagramme in Abbildung 7.13 (A-B) darstellen. Als Resultat erhält man für Free-CLASH eine andere Struktur des Netzwerks. Dies ist anhand eines Beispiels in Abbildung 7.14 zu erkennen.

Abbildung 7.14 (A) stellt dar, welche Knoten sich welchem Cluster für das gegebene Beispielszenario zuordnen. Es ist zu erkennen, dass wie erwartet Mitglieder eines Clusters unmittelbar benachbart sind und sich Cluster weder bei Free-CLASH noch bei Free-GAF überlappen.



Abbildung 7.12 Flussdiagramm der lokalisierungsfreien Clusteringalgorithmen Free-GAF und Free-CLASH

Abbildung 7.14 (B) zeigt die Voronoidiagramme der jeweiligen Clusterheads. Während die Clusterheads in Free-GAF teilweise deutlich ungeordneter erscheinen, bildet sich für Free-CLASH eine eher geordnete Hexagonstruktur heraus.

Abbildung 7.14 (C) zeigt das Voronoidiagramm der Clusterschwerpunkte. Auch hier ist bei Free-CLASH im Vergleich zu Free-GAF eine Struktur mit erhöhter Regularität zu erkennen. Dieses Streben nach einer geordneten Struktur führt schließlich zu einer verbesserten Netzwerkfunktionalität und erhöhten Lebensdauer, wie das nächste Kapitel zeigt.



Abbildung 7.13 Flussdiagramm der Regeln zur Auswahl eines geeigneten Clusterheads, (A) Free-GAF, (B) Free-CLASH, wie sie in Abbildung 7.13 einzusetzen sind



Abbildung 7.14 Anwendung der lokalisierungsfreien Clusteringalgorithmen auf ein Beispielnetzwerk; (A) Zuordnung der Knoten zu ihren Clustern, (B) Voronoi-Diagramm der Clusterheads, (C) Voronoidiagramm der errechneten Clusterschwerpunkte

7.4 Vergleich ausgewählter Algorithmen

Um die Performance der entwickelten Algorithmen zu untersuchen, werden nun die entwickelten Algorithmen Free-GAF und Free-CLASH mit dem in Kapitel 2.4.1 vorgestellten k-Cover Algorithmus verglichen. Weiterhin wurden die reichweitenbasierten Verfahren Free-CLASH und Free-GAF mit den positionsbasierten Verfahren SXGAF und HXGAF sowie ausgewählten MASCLE-Algorithmen verglichen.

Um die Algorithmen zu vergleichen, wurden die in Kapitel 4.2 erläuterten Vergleichskriterien Realistic Case Coverage, Node Efficiency und Lebenszeit sowohl unter den in Kapitel 4.1.6 beschriebenen idealen als auch realitätsnahen Bedingungen untersucht.

Die Simulationsumgebung stellt sich wie in TABELLE 7.2 beschrieben dar und entspricht derselben Simulationsumgebung wie Kapitel 6.3. Da sich die Cluster der reichweitenbasierten Algorithmen von einer zentralen Position aus bilden, ist eine Torus-Welt nicht in der Lage, durch Ecken und Ränder entstehende Effekte zu vermeiden. Bei einer Torus-Welt treffen Clusteringanfragen irgendwann auf bereits geclusterte Koten, was effektiv einem Rand entspricht.

Um Randeffekte zu vermeiden, wurde daher die Variante gewählt, bei der die evaluierte Fläche kleiner als die simulierte Fläche ist. Die simulierte Fläche ist in dieser Simulationsumgebung in jeder Richtung um den Betrag der Working Range größer als die evaluierte Fläche und als zentrale Instanz wurde bei jedem simulierten Sensornetzwerk zufällig ein Sensorknoten innerhalb der evaluierten Fläche ausgewählt.

Ähnlich wie bei den positionsbasierten Algorithmen wurden im ersten Schritt wurde in Abbildung 7.15 die Realistic Case Coverage unter idealen Bedingun-

PARAMETER	WERT
Simulierte Fläche	1112 m*1112 m
Evaluierte Fläche	1000 m*1000 m
Maximal benötigte Sendereichweite	120 m
Sensorreichweite	56 m
Simulierte Netzwerke	250
Kanalmodell	Unit Disk Graph / Log-Normal Shadow Fading
Szenarien	Ideale Bedingungen/Flacher Sand- strand/Trockenes Unterholz
Platzierte Knoten für die Ermittlung der Lebenszeit	10000/ km²
Lebensdauer eines aktiven Knotens T_{AN}	ca. 91 h^{*1}

TABELLE 7.2 SIMULATIONSUMGEBUNG FÜR DEN VERGLEICH DER EIGENSCHAFTEN VON Verschiedenen reichweitenbasierten Clusteringalgorithmen

*1: Entspricht einem MICA2-Knoten im permanenten Datenempfangsmodus mit einer Ausgangsenergie von 5200mAh



Abbildung 7.15 Realistic Case Coverage untersuchter Clusteringverfahren für ideale Bedingungen über verschiedene Anzahlen platzierter Knoten

gen untersucht, die bei einer bestimmten Anzahl an platzierten Knoten erreicht wird.

Es ist zu erkennen, dass für den 1-Cover Algorithmus am wenigsten Knoten notwendig sind, um 100% Netzwerkfunktionalität zu erreichen. Da für 1-Cover jeder mögliche Knoten aktiviert wird, ist das nicht weiter überraschend. Weiterhin ist in dem Diagramm zu erkennen, dass Free-CLASH eine ähnliche Performance wie SXGAF und Free-GAF eine etwas bessere Performance als HXGAF erzielt, die reichweitenbasierten Algorithmen also durchaus mit den positionsbasierten Algorithmen konkurrieren können. An die Performance der MASCLE-Algorithmen können sie jedoch nicht heranreichen.

Im nächsten Schritt wurde dieselbe Untersuchung für die Szenarien flacher Sandstrand und trockenes Unterholz getätigt. Die Ergebnisse sind in Abbildung 7.16 und Abbildung 7.17 dargestellt.



Abbildung 7.16 Realistic Case Coverage untersuchter Clusteringverfahren für das Szenario Flacher Sandstrand über verschiedene Anzahlen platzierter Knoten



Abbildung 7.17 Realistic Case Coverage untersuchter Clusteringverfahren für das Szenario Trockenes Unterholz über verschiedene Anzahlen platzierter Knoten

Es ist zu erkennen, dass sich die Performance der lokalisierungsfreien Algorithmen in beiden Szenarien, vergleichbar mit den XGAF Algorithmen, nur wenig ändert. Die entwickelten lokalisierungsfreien Algorithmen sind also relativ robust gegenüber unterschiedlichen Umgebungen und den damit verbundenen Schwankungen in der Sendereichweite.

Ähnlich sieht es für die in Abbildung 7.18 gezeigte Node Efficiency aus. Hier ist zu erkennen, dass Free-GAF und Free-CLASH mit steigender Anzahl platzierter Knoten ähnliche Werte erzielen, also ungefähr die gleiche Anzahl an Knoten aktivieren, um vollständige Netzwerkfunktionalität zu erzielen. Weiterhin befindet sich die Node Efficiency der beiden lokalisierungsfreien und reichweitenbasierten Algorithmen zwischen den Werten von SXGAF und HXGAF, somit ist auch hier die Performance mit den positionsbasierten XGAF-Algorithmen vergleichbar.

Zum Vergleich ist in Abbildung 7.18 zusätzlich die Node Efficiency der lokali-



Abbildung 7.18 Node Efficiency der lokalisierungsfreien Clusteringalgorithmen sowie weiterer Vergleichsalgorithmen über verschiedene Anzahlen an platzierten Knoten



Abbildung 7.19 Wahrscheinlichkeit kompletter Netzwerkfunktionalität verschiedener Clusteringalgorithmen bei idealen Bedingungen über der Laufzeit des Sensornetzwerks

sierungsfreien Cover-Algorithmen dargestellt. Da diese mit steigender Anzahl platzierter Knoten automatisch die Anzahl aktiver Knoten erhöhen, sinkt die Node Efficiency stetig mit sinkender Anzahl platzierter Knoten. Um nicht zu viele Knoten im Sensornetzwerk zu aktivieren, sollte die geeignete Anzahl an Covern daher von der Anzahl platzierter Knoten abhängen.

Das Problem der geeigneten Anzahl von Covern spiegelt sich auch bei der Lebenszeit wider. Abbildung 7.19 stellt die Lebenszeit der lokalisierungsfreien Algorithmen und einiger Vergleichsalgorithmen dar. Wie in der Abbildung zu erkennen, gilt bei den gewählten Annahmen, dass nach k Zeiteinheiten die Energie sämtlicher Knoten eines gewählten k-Covers verbraucht sind. Selbst bei einer sorgfältigen Wahl der Anzahl an Covern wird jedoch nur eine Lebenszeit von $T_{life}=7,8T_{AN}$ erreicht.

Free-GAF wiederum besitzt mit $T_{life}=8,6T_{AN}$ zwar eine höhere Lebenszeit als die k-Cover, aber eine deutlich geringere Lebenszeit als Free-CLASH, welches eine Lebenszeit von $T_{life}=10,2T_{AN}$ aufweist. Ursache dafür sind erneut in der von Free-CLASH angestrebten Gleichmäßigkeit der Clustergrößen zu erkennen. Dieses Verhalten spiegelt sich auch in Abbildung 7.20 wider. Hier ist der Anteil aktiver Knoten über der Laufzeit dargestellt. Man sieht, dass Free-GAF aufgrund einer höheren Anzahl von Clustern mit einem deutlich höheren Anteil aktiver Knoten startet, dieser Anteil jedoch zu Beginn deutlich schneller sinkt als der Anteil aktiver Knoten von Free-CLASH. Ursache dafür ist das Streben nach Gleichmäßigkeit von Free-CLASH, durch welches zum einen weniger Knoten aktiviert sind, zum anderen dafür gesorgt wird, dass die Cluster gleichmäßiger besetzt werden. Dadurch kommt es im Gegensatz zu Free-GAF weniger häufig zu vorzeitigen Ausfällen einzelner, für die Funktionalität des


Abbildung 7.20 Anteil aktiver Knoten zur Gesamtzahl platzierter Knoten verschiedener untersuchter Clusteringalgorithmen über der Laufzeit des Sensornetzwerks

Netzwerks jedoch entscheidender Cluster. Daher erzielt Free-CLASH die höchste Lebenszeit der untersuchten lokalisierungsfreien Algorithmen und erreicht sogar 96% der Lebenszeit von SXGAF, sodass Free-CLASH dementsprechend mit einem positionsbasierten Algorithmus konkurrieren kann. Um die Lebenszeit der lokalisierungsfreien Clusteringalgorithmen auch unter realitätsnahen Bedingungen zu untersuchen, wurde die Lebenszeit der Algorithmen unter Schwankungen in der Sendereichweite untersucht, wie sie in den Szenarien Flacher Sandstrand und Trockenes Unterholz vorliegen. Die Ergebnisse dieser Untersuchungen sind in Abbildung 7.21 und Abbildung 7.22 gezeigt. Man erkennt, dass das Übertragungsverhalten im Flachen Sandstrand bereits Einfluss auf die Lebenszeit von Free-GAF hat und zu einer Reduzierung der Lebenszeit um 14% auf $T_{life}=7,4T_{AN}$ führt. Der Einfluss auf Free-



Abbildung 7.21 Wahrscheinlichkeit kompletter Netzwerkfunktionalität verschiedener Clusteringalgorithmen im Szenario Flacher Sandstrand über die Laufzeit des Sensornetzwerks



Abbildung 7.22 Wahrscheinlichkeit kompletter Netzwerkfunktionalität verschiedener Clusteringalgorithmen im Szenario Trockenes Unterholz über der Laufzeit des Sensornetzwerks

CLASH ist ähnlich wie bei den positionsbasierten XGAF Algorithmen im Szenario Flacher Sandstrand vernachlässigbar gering. Ursache dafür ist wiederum in dem Bestreben nach Gleichförmigkeit zu finden. So wird selbst ein durch ungünstige Übertragungsbedingungen verzerrter Cluster häufig immer noch in der Lage sein, seine Aufgabe zu erfüllen.

Anders sieht es für das Szenario Trockenes Unterholz aus. Aufgrund der schwankenden Sendereichweiten sind beide Algorithmen nicht mehr in der Lage, korrekt Knoten in ihre Cluster aufzunehmen. Somit entsteht zum einen eine größere Varianz in der Anzahl an Clustermitgliedern, zum anderen werden auch nachfolgende Clusterheads zufälliger ermittelt, was ebenso zu einer größeren Unregelmäßigkeit der Cluster führt. Durch dieses Verhalten sinkt die Lebenszeit von Free-CLASH verglichen mit idealen Bedingungen um 15,7% auf $T_{life}=8,6T_{AN}$ und für Free-GAF sogar um 25,6% auf $T_{life}=6,4T_{AN}$. Dieses Absinken ist stärker als das der positionsbasierten Algorithmen und führt für Free-GAF sogar deutlich unter die Trendlinie der k-Cover Algorithmen, deren Netzwerkfunktionalität aufgrund der zufälligen Wahl der aktiven Knoten unberührt von den Übertragungsbedingungen bleibt.

7.5 Zwischenfazit

In dem vorliegenden Kapitel wurden die entwickelten lokalisierungsfreien Clustering-Verfahren untersucht. Beginnend mit BBC, welches aufgrund seiner benötigten regelmäßig angeordneten Referenzknoten einen Sonderfall darstellt, wurden weiterhin die entwickelten Algorithmen Free-GAF und Free-CLASH vorgestellt, deren Performance sich mit denen der positionsbasierten Verfahren sowie den k-Cover Algorithmen vergleichen lässt. Zusammenfassend lässt sich erkennen, dass die lokalisierungsfreien Verfahren zwar eine geringere Lebenszeit erzielen, aber für relativ günstige Übertragungsbedingungen, wie sie im Szenario Flacher Sandstrand vorherrschen, mit den XGAF-Algorithmen konkurrieren können. Für Szenarien, in denen eine Lokalisierung nicht notwendig oder beispielsweise aufgrund fehlender oder fehlerhafter Positionsinformationen möglich stellen die ist. reichweitenbasierten und lokalisierungsfreien Clusteralgorithmen daher eine echte Alternative zu den positionsbasierten Algorithmen dar. Im Vergleich zu den k-Cover Algorithmen besitzen die Algorithmen Free-GAF und Free-CLASH mehrere entscheidende Vorteile. Zum einen wird mit der Anzahl an Covern auch die maximale Lebenszeit jedes Knotens bestimmt, sodass bei jedem k-Cover Algorithmus unter den getroffenen idealen Annahmen nach k Zeiteinheiten jeder Knoten seine Energie verbraucht hat. Im Gegensatz dazu ist in Free-GAF bzw. Free-CLASH auch mit dem Verlust der vollständigen Netzwerkfunktionalität noch eine große Anzahl in Cluster organisierter Knoten übrig, welche weiterhin verwendet werden können. Es kommt also nicht zu einem abrupten Ausfall des Netzwerks. Der zweite große Vorteil ist, dass es in Free-GAF und insbesondere Free-CLASH möglich ist, den aktiven Knoten innerhalb eines Clusters ohne Einschränkung der Netzwerkfunktionalität austauschen zu können und somit spontan auf Knotenausfälle reagieren zu können. Auch kann, ähnlich wie bei den XGAF- oder MASCLE-Algorithmen, die Clusterstärke bestimmt und beispielsweise für Routing-Verfahren ausgenutzt werden.

8 Zusammenfassung und Ausblick

Die Entwicklung drahtlos kommunizierender Sensornetzwerke ist eine Schlüsseltechnologie des 21. Jahrhunderts. Ein massiver Einsatz von Sensorknoten kann in vielen Szenarien Umweltdaten liefern, auf welche mit herkömmlichen Methoden kein Zugriff stattfinden kann. Eine der maßgeblichen Einschränkungen für die massive Verwendung ist die derzeitige Größe der Sensorknoten, welche hauptsächlich von der Energieversorgung abhängig ist. Um diese derzeit größte Komponente eines Sensorknotens auf eine von vielen Szenarien benötigte unauffällige Größe reduzieren zu können, und gleichzeitig eine ausreichende Lebenszeit des Netzwerks erreichen zu können, sind energiesparende Algorithmen unerlässlich.

Das Konzept des Clusterings bietet in diesem Zusammenhang das Potential, aus einer Menge zufällig verteilter Knoten nur einen Bruchteil zu aktivieren und somit bei den abgeschalteten Knoten enorme Energie zu sparen. Herausforderung ist es hierbei, die geeigneten Knoten auszuwählen, welche in einem aktiven Zustand bleiben.

Der Kernpunkt dieser Arbeit bestand daher darin, neue energieeffiziente Clusteringverfahren zu entwickeln. Da sich Clusteringverfahren, welche auf der Position der Sensorknoten im Sensornetzwerk beruhen, als am aussichtsreichsten für eine Steigerung der Lebenszeit zeigten, wurde zusätzlich auf die Lokalisierung von Sensorknoten im Sensornetzwerk eingegangen und ein Algorithmus entwickelt, welcher die Vorteile von heuristischen und exakten Verfahren kombiniert, Ausreißer eliminiert und somit eine Erhöhung der Lokalisierungsgenauigkeit für jedes untersuchte Szenario von mindestens 15% erzielt.

Als zentrales Ergebnis dieser Arbeit wurden daraufhin mehrere energieeffiziente geographische Clusteringverfahren entwickelt. Dabei gelang es, mit Free-CLASH ein reichweitenbasiertes, lokalisierungsfreies Verfahren zu entwickeln, dessen Lebenszeit an die der positionsbasierten Verfahren zu 96% heranreicht und im Vergleich zu bisherigen lokalisierungsfreien k-Cover Verfahren eine Lebenszeitsteigerung von 30,7% erzielt.

Bei den positionsbasierten Verfahren gelang es mit den MASCLE-Verfahren, eine Selbstheilung auf Clusterbasis in das Sensornetzwerk einzubetten. Dabei gelang es, mit dem entwickelten Hex-MASCLE³ Algorithmus innerhalb des untersuchten Szenarios eine Steigerung der Lebenszeit zum zeitlichen Clusteringverfahren CPNS von mehr als 17,2% herzustellen. Zum geographischen Ursprungsalgorithmus SXGAF konnte sogar eine Lebenszeitsteigerung von mehr als 105% erreicht, die Lebenszeit des Netzwerks also verdoppelt werden.

Neben diesen unter idealen Bedingungen ermittelten Ergebnissen wurde für ausgewählte Clusteringalgorithmus das Verhalten unter realitätsnahen Bedingungen untersucht. Dazu wurde der Netzwerksimulator PROWLER so modifiziert, dass er das Log-Normal Shadow Fading Modell verwenden kann, und die Algorithmen für Kanalmodelle von zwei für Sensornetzwerke relevante Szenarien untersucht. Dabei wurde gezeigt, dass die Algorithmen, welche eine hohe Lebenszeit erzielen, für gewöhnlich anfälliger für Lokalisierungsfehler sind und Löcher im Netzwerk entstehen können. Obwohl für das fehlerbehaftetere Szenario "Trockenes Unterholz" Hex-MASCLE nicht mehr für alle Knotenanordnungen 100%ige Netzwerkfunktionalität gewährleisten kann, ist 2-MASCLE in der Lage, noch problemlos mit dem CPNS-Algorithmus konkurrieren zu können und dem Sensornetzwerk die Vorteile eines geographischen Clusterings bieten zu können.

8.1 Ausblick

Die in der Arbeit entwickelten Algorithmen wurden aufgrund der begrenzt verfügbaren Hardware bislang nur in Simulationen getestet. Sollte ein Anwendungsszenario und die Menge verfügbarer Knoten feststehen, so könnte ein Test auf realer Hardware die Gelegenheit bieten, die ermittelten Ergebnisse zu bestätigen oder neue Erkenntnisse zu gewinnen.

Desweiteren wurden die Algorithmen bisher nur auf zweidimensional angeordnete und gleichverteilte Knoten angewendet. Eine Anpassung der entwickelten Algorithmen für die dritte Dimension oder auf andere mögliche Knotenverteilungen könnte den Einsatz der entwickelten Verfahren in der Realität vielfältiger gestalten.

In zukünftigen, auf der vorliegenden Arbeit aufbauenden Arbeiten sollte untersucht werden, wie sich die Vorteile der austauschbaren Knoten innerhalb eines Clusters maximal ausnutzen lassen. Basis könnte ein selbstorganisierter Rollentausch sein, welcher in [53] beschrieben wird.

Bei den in der vorliegenden Arbeit beschriebenen Verfahren handelt es sich um Cluster unterschiedlicher Stärke. Hier sollte untersucht werden, wie man die im Sensornetzwerk entstehenden Aufgaben so verteilen kann, dass Cluster mit geringerem Restenergieniveau selbstorganisiert weniger Aufgaben erhalten, so dass frühzeitige Ausfälle von ganzen Clustern vermieden werden. Einen ersten Ansatz bietet dabei der in [38] beschriebene adaptive Routingbaum.

Für eine Weiterentwicklung der reichweitenbasierten, lokalisierungsfreien Algorithmen gibt es zwei wesentliche Ansätze. Zum einen könnte der Gedanke der MASCLE-Algorithmen ebenfalls auf diese Verfahren angewendet werden. Die Aktivierung weniger Cluster würde auch hier bedeutende Energieersparnisse einbringen. Zum anderen sollte man untersuchen, in wie weit sich eine nachträgliche Anpassung der Clustergrenzen und Umverteilung der Clustermitglieder lohnen könnte. So könnten schwache Cluster vermieden oder regelmäßigere Strukturen hergestellt werden. Auch eine selbstständige Reaktion auf dynamische Veränderungen innerhalb des Netzwerks, beispielsweise durch eine unerwartete Positionsänderung oder das Ausbringen neuer Knoten zur Laufzeit ist ein wesentlicher Forschungsschwerpunkt, welcher die Anwendbarkeit der entwickelten Clusteringalgorithmen weiter erhöht.

Literaturverzeichnis

- [1] "BMP085m Digital pressure sensor", Available: <u>www.bosch-sensortec.com</u>, 2008
- [2] A. Merkle and A. Terzis, "Digitale Funkkommunikation mit Bluetooth". Poing, Franzis' Verlag GmbH, 2002
- [3] "Single Chip Very Low Power RF Transceiver with 8051-Compatible Microcontroller", Available: <u>http://www.keil.com/dd/docs/datashts/chipcon/cc1010_ds.pdf</u>, 2002
- [4] F. Dressler, "Self-Organization in Sensor and Actor Networks". Chichester, John Wiley & Sons, 2007
- [5] Zo, Abrams, G. Ashish, and P. Serge, "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks", in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, Berkeley, California, USA, 2004
- [6] "BMA150, Digital, triaxial acceleration sensor", Available: www.bosch-sensortec.com, 2008
- [7] "MPR-MIB Users Manual", Available: <u>http://www.xbow.com</u>, 2004
- [8] "Datasheet SHT21, Humidity and Temperature Sensor", Available: www.sensirion.com, 2010
- [9] "Tmote Sky: Ultra low power IEEE 802.15.4 compilant wireless sensor module", Available: <u>http://www.snm.ethz.ch/pub/uploads/Projects/tmote_sky_datasheet.pdf</u>, 2006
- [10] K. Sohrabi, B. Manriquez, and G. J. Pottie, "Near ground wideband channel measurement in 800-1000 MHz", in *Vehicular Technology Conference, IEEE 49th*, pp. 571-574, 1999

- [11] "SensorChipTM CO2, IR gas sensor 'on a chip'", Available: www.ams.de, 2009
- [12] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next Century Challenges: Mobile Networking for 'Smart Dust", in *Fifth Annual ACM/IEEE Int. Conference on Mobile Computing and Networking*, pp. 271-278, 1999
- [13] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring", in *Proceedings of* the 1st ACM international workshop on wireless sensor networks and applications, Atlanta, USA, pp. 88 - 97, 2002
- [14] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring Volcanic Eruptions with a Wireless Sensor Network", in *Proceedings of the Second European Workshop on Wireless Sensor Networks* Istanbul, Türkei, 2005
- [15] H. Stroppe, "Physik für Studenten der Natur- und Technikwissenschaften". Fachbuchverlag Leipzig, 1999
- [16] A. S. Tanenbaum, "Computer Networks, Fourth Edition". Precentice Hall, 2002
- [17] K. David, N. Calvin, S. G. Robert, L. Jason, Y. Yougu, and E. Chip, "Experimental evaluation of wireless simulation assumptions", in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, Venice, Italy, 2004
- [18] T. S. Rappaport and S. Sandhu, "Radio-wave propagation for emerging wireless personal-communication systems", *Antennas and Propagation Magazine, IEEE*, vol. 36, pp. 14-24, 1994
- [19] T. S. Rappaport, "Wireless communications: principles and practice". Prentice Hall PTR, 2002
- [20] S. Martinez, Alejandro, P. Molina Garcia, M. Jose, L. Egea, Esteban, A. Vales, Javier, L. Juan, Leandro, H. Garcia, and Joan, "An accurate radio channel model for wireless sensor networks simulation : Communica-

tion and Signal Processing in Sensor Networks", vol. 7. Seoul, Korea, Korean Institute of Communication Sciences, 2005

- [21] L. Starke, "Grundlagen der Funk- und Kommunikationstechnik". Heidelberg, Hüthig Verlag GmbH, 1996
- [22] "Electronic Nose ARTINOS® Multi-Scan", Available: http://www.sysca-ag.de, 2009
- [23] L. Bencini, F. Chiti, G. Collodi, D. D. Palma, R. Fantacci, A. Manes, and G. Manes, "Agricultural monitoring based on Wireless Sensor Network technology: real long life deployments for physiology and pathogens control", in *Third International Conference on Sensor Technologies and Applications*, Athen, Griechenland, pp. 372-378, 2009
- [24] E. Basha and D. Rus, "Design of Early Warning Flood Detection Systems for Developing Countries", in *Proceedings of the Conference on Information and Communication Technologies and Development*, Bangalore, Indien, 2007
- [25] J. Schulz, F. Reichenbach, J. Blumenthal, and D. Timmermann, "Low cost system for detecting leakages along artificial dikes with wireless sensor networks", in *European Conference on Computer Systems, Proceedings of the workshop on Real-world wireless sensor networks*, Glasgow, Scotland, pp. 66-70, 2008
- [26] M. Hefeeda, "Forest Fire Modeling and Early Detection using Wireless Sensor Networks", Ad Hoc & Sensor Wireless Networks, vol. 7, pp. 169-224, 2009
- [27] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, and A. Wood, "Wireless Sensor Networks for In-Home Healthcare: Potential and Challenges", in *High Confidence Medical Device Software and Systems Workshop*, Philadelphia, USA, 2005
- [28] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A

line in the sand: a wireless sensor network for target detection, classification, and tracking", *Computer Networks*, vol. 46, pp. 605-634, 2004

- [29] "IEEE Standard for Information technology--Telecommunications and information exchange between systems--Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments", *IEEE Std* 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009), pp. 1-51,
- [30] S. C. Ergen and P. Varaiya, "PEDAMACS: power efficient and delay aware medium access protocol for sensor networks", *Mobile Computing*, *IEEE Transactions on*, vol. 5, pp. 920-930, 2006
- [31] Y. Wei, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks", in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, pp. 1567-1576 vol.3, 2002
- [32] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks", in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, p. 224, 2004
- [33] M. Ringwald and K. Romer, "BitMAC: a deterministic, collision-free, and robust MAC protocol for sensor networks", in *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, pp. 57-69, 2005
- [34] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey", *Communications Magazine, IEEE*, vol. 44, pp. 115-121, 2006
- [35] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing", in *Mobile Computing Systems and Applications, 1999. Pro*ceedings. WMCSA '99. Second IEEE Workshop on, pp. 90-100, 1999

- [36] D. Hui and R. Han, "A node-centric load balancing algorithm for wireless sensor networks", in *Global Telecommunications Conference*, 2003. *GLOBECOM '03. IEEE*, pp. 548-552 Vol.1, 2003
- [37] X.-X. Liu, Y. Cao, and X.-W. Chen, "Mouse colony optimization and simulated annealing algorithm for energy balance routing in wireless sensor networks", *The International Journal of Systems & Cybernetics*, vol. 38, pp. 406-416, 2009
- [38] J. Salzmann, S. Kubisch, F. Reichenbach, and D. Timmermann, "Energy and Coverage Aware Routing Algorithm in Self Organized Sensor Networks", in *Fourth International Conference on Networked Sensing Systems, INSS '07*, Braunschweig, Deutschland, pp. 77-80, 2007
- [39] Y. Zhang and Q. Huang, "A Learning-based Adaptive Routing Tree for Wireless Sensor Networks", *JCM*, vol. 1, pp. 12-21, 2006
- [40] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks", *Ad Hoc Networks*, vol. 3, pp. 325-349, 2005
- [41] A. Ameer Ahmed and Y. Mohamed, "A survey on clustering algorithms for wireless sensor networks", *Comput. Commun.*, vol. 30, pp. 2826-2841, 2007
- [42] T. Di and D. G. Nicolas, "A coverage-preserving node scheduling scheme for large wireless sensor networks", in *Proceedings of the 1st* ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, USA, 2002
- [43] X. Ya, H. John, and E. Deborah, "Geography-informed energy conservation for Ad Hoc routing", in *Proceedings of the 7th annual international conference on Mobile computing and networking*, Rom, Italien, 2001
- [44] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energyefficient communication protocol for wireless microsensor networks", in System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, p. 10 pp. vol.2, 2000

- [45] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach", in *INFOCOM 2004*. *Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, p. 640, 2004
- [46] C. L. a. K. W. a. V. King, "Randomized Coverage-Preserving Scheduling Schemes for Wireless Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, 2005
- [47] F. Cattivelli and A. H. Sayed, "Self-organization in bird flight formations using diffusion adaptation", in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*, pp. 49-52, 2009
- [48] H. Karl, A. Wolisz, A. Willig, H. Chan, and A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation", vol. 2920. Springer Berlin / Heidelberg, 2004
- [49] E. Deborah, G. Ramesh, H. John, and K. Satish, "Next century challenges: scalable coordination in sensor networks", in *Proceedings of the* 5th annual ACM/IEEE international conference on Mobile computing and networking, Seattle, Washington, USA, 1999
- [50] C. Müller-Schloer, H. Schmeck, T. Ungerer, J. Salzmann, R. Behnke, and D. Timmermann, "OC Principles in Wireless Sensor Networks", in Organic Computing — A Paradigm Shift for Complex Systems. vol. 1, Springer Basel, pp. 503-516, 2011
- [51] M. Ding, D. Chen, K. Xing, and X. Chen, "Localized fault-tolerant event boundary detection in sensor networks", in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, Miami, Florida, USA, pp. 902-913, 2005
- [52] R. Behnke, J. Salzmann, S. Simanowski, and D. Timmermann, "Efficient Localized Detection of Erroneous Nodes (ELDEN)", in *IEEE 10th International Conference on Computer and Information Technology* (CIT), pp. 2884-2890, 2010

- [53] F. Reichenbach, A. Bobek, P. Hagen, and D. Timmermann, "Increasing Lifetime of Wireless Sensor Networks with Energy-Aware Role-Changing", in *Self-Managed Networks, Systems, and Services*. vol. 3996, Springer Berlin / Heidelberg, pp. 157-170-170, 2006
- [54] "The Network Simulator ns-2", Available: http://www.isi.edu/nsnam/ns/, 2011
- [55] "J-Sim", Available: http://www.j-sim.zcu.cz/, 2011
- [56] "Opnet", Available: <u>http://www.opnet.com</u>, 2011
- [57] "GloMoSim Global Mobile Information Symstems Simulation Library", Available: <u>http://pcl.cs.ucla.edu/projects/glomosim/</u>, 2011
- [58] "Prowler: Probalistic Wireless Network Simulator", Available: http://www.isis.vanderbilt.edu/Projects/nest/prowler/, 2011
- [59] "RMASE: Rouing Modeling Application Simulation Environement", Available: <u>http://www2.parc.com/isl/groups/era/nest/Rmase/</u>, 2011
- [60] R. Behnke, J. Salzmann, R. Grossmann, D. Lieckfeldt, D. Timmermann, and K. Thurow, "Strategies to overcome border area effects of coarse grained localization", in *Positioning, Navigation and Communication*, 2009. WPNC 2009. 6th Workshop on, pp. 95-102, 2009
- [61] J. Salzmann, R. Behnke, D. Lieckfeldt, and D. Timmermann, "2-Mascle - A Coverage Aware Clustering Algorithm with Self Healing Abilities", in 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne, Australien, 2007
- [62] G. Xu, "GPS: Theory, Algorithms and Applications". Springer, 2007
- [63] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices", *Personal Communications*, *IEEE*, vol. 7, pp. 28-34, 2000
- [64] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted Centroid Localization in Zigbee-based Sensor Networks", in

Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on, pp. 1-6, 2007

- [65] S. Andreas, H. Chih-Chieh, and B. S. Mani, "Dynamic fine-grained localization in Ad-Hoc networks of sensors", in *Proceedings of the 7th annual international conference on Mobile computing and networking*, Rom, Italien, 2001
- [66] R. Behnke, J. Salzmann, and D. Timmermann, "SBCL improved centroid estimation", in Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on, pp. 1-6, 2009
- [67] R. Behnke and D. Timmermann, "AWCL: Adaptive Weighted Centroid Localization as an efficient improvement of coarse grained localization", in *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pp. 243-250, 2008
- [68] J. Salzmann, R. Behnke, and D. Timmermann, "HyPAERLoc: Plausible Hybrid Localization for Wireless Sensor Networks ", in 5th International Conference on Sensor Technologies and Applications (SENSORCOMM'11), Nice, Frankreich, 2011
- [69] R. Behnke, J. Salzmann, D. Lieckfeldt, and D. Timmermann, "sDLS -Distributed Least Squares localization for large Wireless Sensor Networks", in Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference on, pp. 1-6, 2009
- [70] J. Salzmann, R. Behnke, and D. Timmermann, "Tessellating Cell Shapes for Geographical Clustering", in *Proceedings of the 10th IEEE International Conference on Computer and Information Technology* (CIT 2010), Bradford, England, pp. 2891-2896, 2010
- [71] J. Salzmann, R. Behnke, and D. Timmermann, "Analyse regelmäßiger Clusterformen in Sensornetzwerken", in 2. Symposium Maritime Elektrotechnik, Elektronik und Informationstechnik, Rostock, Deutschland, 2007
- [72] J. Salzmann, R. Behnke, M. Gag, and D. Timmermann, "4-MASCLE -Improved Coverage Aware Clustering with Self Healing Abilities", in

Ubiquitous, Autonomic and Trusted Computing, 2009. UIC-ATC '09. Symposia and Workshops on, Brisbane, Australien, pp. 537-543, 2009

- [73] J. Salzmann, R. Behnke, and D. Timmermann, "Hex-MASCLE hexagon based clustering with self healing abilities", in *Wireless Communications and Networking Conference (WCNC), IEEE*, Cancun, Mexiko, pp. 528-533, 2011
- [74] J. Salzmann, R. Behnke, and D. Timmermann, "Redundancy Aware Clustering via Centroid Localization Technique", in *Proceedings of the Architecture of Computing Systems ARCS*, Hannover, Deutschland, pp. 359-365, 2010
- [75] J. Salzmann, R. Behnke, and D. Timmermann, "A Self-Organized Localization-Free Clustering Approach for Redundancy Exploitation in Large Wireless Sensor Networks", in *GI Jahrestagung, Workshop: Adaptive und organische Systeme*, München, Deutschland, pp. 747-754, 2008
- [76] J. Salzmann, R. Behnke, J. You, and D. Timmermann, "Free-CLASH -Improved Localization-Free Clustering in Large Wireless Sensor Networks", in *International Conference on Ultra Modern Telecommunications*, St. Petersburg, Russland, pp. 1-6 2009

Anhang

Thesen

Drahtlose Sensornetzwerke

- 1. Sensornetzwerke bestehen aus einer Vielzahl zufällig verteilter Sensorknoten und einer zentralen Instanz. Aufgrund der zufälligen Verteilung der Sensorknoten müssen möglicherweise zusätzliche, redundante Knoten platziert werden, um jeden Punkt eines zu überwachenden Gebiets abzudecken.
- 2. Ein Sensorknoten vereint Sensoreinheit, Funkeinheit, Microcontroller und Batterie auf wenigen Kubikzentimetern. Er erfasst Daten mit seinen Sensoren und leitet diese mit Hilfe anderer Netzwerkteilnehmer zur zentralen Instanz.
- 3. Limitierender Faktor für Einsatzdauer, Einsatzbereich sowie Größe der Sensorknoten ist die Batterie. Ein energieeffizienter Betrieb des drahtlosen Sensornetzwerks ist daher für ein effizientes drahtloses Sensornetzwerk unerlässlich.

Selbstorganisation und Selbstheilung

- 4. Aufgrund der schieren Anzahl möglicher Knoten im Sensornetzwerk sind herkömmliche, zentralisierte Organisationsstrukturen, die jeden Knoten individuell steuern, ineffizient. Eine Steuerung mittels "Localized Algorithms", welche nicht das Verhalten einzelner Knoten kontrolliert, sondern individuell umzusetzende Regeln ins Netzwerk gibt, stellt eine energieeffiziente Alternative dar.
- 5. Werden Knoten an zufälligen Positionen im Netzwerk platziert, führt die entstehende geographische Redundanz dazu, dass Knoten abgeschaltet werden können, um Energie zu sparen. Mit der als Selbstheilung bezeichneten Methode können diese Knoten später selbstständig entscheiden in den Netzwerkbetrieb zurück zu wechseln, um bislang ausgefallene Knoten zu ersetzen.

Lokalisierung

- 6. Um eine Lokalisierung auch in unzugänglichen Gebieten zu ermöglichen, können Referenzknoten verwendet werden, die ihre Position kennen. Mittels Broadcast teilen diese Referenzknoten den umliegenden Knoten ihre Position mit. Diese können durch die Auswertung der Signalstärke zudem die Entfernung zu den Referenzknoten bestimmen.
- 7. Durch die ermittelte Entfernung zu Referenzknoten in Reichweite kann ein Knoten seine Position bestimmen. Je nach maximal erreichbarer Genauigkeit werden dabei heuristische und exakte Lokalisierungsverfahren unterschieden.

- 8. Die untersuchten heuristischen Lokalisierungsverfahren haben den Vorteil, nur eine relativ geringe maximale Abweichung von der eigentlichen Position zu errechnen. Dafür ist jedoch selbst unter idealen Funkbedingungen keine genaue Lokalisierung möglich.
- 9. Die untersuchten exakten Verfahren zur Lokalisierung haben das Potential höchste Genauigkeiten zu erreichen. Unter realitätsnahen Bedingungen und einer ungünstigen Anordnung der Referenzknoten in Kommunikationsreichweite kann es vorkommen, dass ein Knoten sich auf eine extrem ungenaue Position lokalisiert.
- 10. Das neu vorgestellte Lokalisierungsverfahren HyPAERLoc kombiniert die Lokalisierung eines exakten Verfahrens mit der Lokalisierung eines heuristischen Verfahrens. Durch diese Kombination wird zum einen eine deutlich genauere Lokalisierung ermöglicht, zum anderen werden Ausreißer minimiert, welche typisch für die untersuchten exakten Verfahren sind.

Clustering

- 11. Das Einteilen eines Sensornetzwerks in Gruppen wird als Clustering bezeichnet. Mit dem geographischen Clustering werden Knoten einer bestimmten Region zu einem Cluster gruppiert. Bei dem zeitlichem Clustering besitzt dagegen jeder Cluster möglichst Knoten aus dem gesamten zu überwachenden Gebiet.
- 12. Sind genügend Knoten im Netzwerk vorhanden und wird die maximale Ausdehnung der geographischen Cluster anhand der Sensor- und Sendereichweite gewählt. So ist es möglich, alle Knoten mit Ausnahme des Clusterheads temporär zu deaktivieren, was zu einer bedeutenden Energieersparnis führt.

Positionsbasiertes Clustering

- 13. Ist jedem Knoten seine Position bekannt und sind Aussagen über Sensor- und Sendereichweite verfügbar, werden mit dem XGAF (Extended Geographical Adaptive Fidelity) Verfahren mit Hilfe eines virtuellen Gitters Form und Größe der Cluster vorab festgelegt. Um garantierte Konnektivität und Sensorabdeckung zu gewährleisten, darf die maximale Ausdehnung eines Clusters nicht größer als die halbe Sendereichweite und nicht größer als die Sensorreichweite sein.
- 14. Bei den MASCLE (Mutual Assistance in a Cluster Environment) Verfahren wird die maximale Ausdehnung der Cluster verringert. Da die so entstehenden Cluster eine größere Fläche abdecken und weiter kommunizieren können als XGAF Cluster, ist es möglich, in regelmäßigen Abständen ganze Cluster temporär zu deaktivieren, was zu einer weiteren Energieersparnis führt.
- 15. Die geringere Clusterausdehnung der MASCLE Verfahren bietet zusätzlich das Potential, ausgefallene oder nicht besetzte Cluster im Netzwerk durch bestimmte Anordnungen von Nachbarclustern zu er-

setzen. Diese zellbasierte Selbstheilung sorgt für eine erheblich höhere Lebenszeit als die von XGAF und anderen Clusteringverfahren.

Lokalisierungsfreies Clustering

- 16. Werden regelmäßig angeordnete Knoten als Referenzknoten für eine Lokalisierung verwendet, so kann für das XGAF-ähnliche Clustering auf eine genaue Lokalisierung verzichtet werden. Stattdessen werden Knoten mit dem BBC (Beacon Based Clustering) Verfahren anhand der empfangenen Beacons in Cluster eingeordnet. Um die maximal erlaubte Ausdehnung der so entstehenden Zellen einzuhalten, sind bestimmte Beaconabstände und Sendereichweiten notwendig.
- 17. Steht keine Lokalisierung zur Verfügung, können XGAF-ähnliche Cluster durch gezielt angepasste Sendereichweiten von ausgewählten Clusterheads erreicht werden. Mittels des Free-GAF (Localization Free – Geographic Adaptive Fidelity) Algorithmus wird ein Sensornetzwerk von einem zentralen Knoten aus sequentiell geclustert.
- 18. Mit dem Free-CLASH (Localization Free Clustering with Approximation to Symmetric Hexagons) Clusteringalgorithmus wird die Auswahl von Clusterheads und zugehöriger Knoten verfeinert. Durch die gezieltere Auswahl wird eine homogenere Clusterstruktur geschaffen und eine Lebenszeit ähnlich den positionsbasierten XGAF-Algorithmen erreicht.

Liste der wissenschaftlichen Veröffentlichungen

Begutachtete Konferenzbeiträge

- 4.) Behnke, Ralf; Salzmann, Jakob; Gorski, Philipp; Timmermann, Dirk, HDLS: Improved Localization via Algorithm Fusion; Proceedings of 5th International Conference on Sensor Technologies and Applications (SENSORCOMM'11), ISBN: 978-1-61208-010-9, Nice, France, August 2011
- 5.) Behnke, Ralf; Born, Alexander; Salzmann, Jakob; Timmermann, Dirk; Bill, Ralf; Combining Scalability and Resource Awareness in Wireless Sensor Network Localization; 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011), pp. 537-542, ISBN: 978-1-4244-9920-5, Shanghai, China, April 2011
- 6.) Behnke, Ralf; Salzmann, Jakob; Timmermann, Dirk; sDLSne Improved Scalable Distributed Least Squares Localization with minimized Communication; 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2010), pp. 1693-1698, ISBN: 978-0-4244-8015-9, Istanbul, Turkey, September 2010
- Behnke, Ralf; Salzmann, Jakob; Simanowksi, Stephan; Timmermann, Dirk; Efficient Localized Detection of Erroneous Nodes (ELDEN); 10th IEEE International Conference on Computer and Information Technology (CIT 2010), pp. 2884-2890, ISBN: 978-0-7695-4108-2, Bradford, UK, June 2010
- Behnke, Ralf; Salzmann, Jakob; Timmermann, Dirk; Improvements on Scalable Distributed Least Squares Localization for Large Wireless Sensor Networks; 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC'2010), pp. 273-277, ISBN: 978-1-4244-6857-7, Modena, Italy, May 2010
- 9.) Behnke, Ralf; Salzmann, Jakob; Timmermann, Dirk; Minimizing Cost of Scalable Distributed Least Squares Localization; 2nd International Workshop on Wireless Sensor Networks: theory and practice; in conjunction with NTMS 2009, ISBN: 978-1-4244-6273-5, Kairo, Egypt, December 2009

- Behnke, Ralf; Salzmann, Jakob; Timmermann, Dirk; sDLSnd -Scalable Distributed Least Squares Localization with reduced Complexity; 2nd International Workshop on Service computing, Contextaware, Location-aware and Positioning techniques; in conjunction with NTMS 2009, ISBN: 978-1-4244-6273-5, Kairo, Egypt, December 2009
- Behnke, Ralf; Salzmann, Jakob; Timmermann, Dirk; SBCL -Improved Centroid Estimation; International Workshop on Sensing and Acting in Ubiquitous Environments (SEACUBE'09), ISBN: 978-1-4244-3941-6, St. Petersburg, Russland, Oktober 2009
- 12.) Behnke, Ralf; Salzmann, Jakob; Lieckfeldt, Dominik; Timmermann, Dirk; sDLS - Distributed Least Squares Localization for Large Wireless Sensor Networks; International Workshop on Sensing and Acting in Ubiquitous Environments (SEACUBE'09), ISBN: 978-1-4244-3941-6, St. Petersburg, Russia, October 2009
- Behnke, Ralf; Salzmann, Jakob; Grossmann, Ralf; Lieckfeldt, Dominik; Timmermann, Dirk; Thurow, Kerstin; Strategies to overcome Border Area Effects of Coarse Grained Localization; Workshop on Positioning, Navigation and Communication 2009 (WPNC'09), pp. 95-102, ISBN: 978-1-4244-3292-9, Hannover, Germany, March 2009
- 14.) Behnke, Ralf; Salzmann, Jakob; Lieckfeldt, Dominik; Golatowski, Frank; Timmermann, Dirk; Thurow, Kerstin; Exploiting Malicious Node Detection for Lifetime Extension of Sensor Networks; 6th Annual IEEE Consumer Communications and Networking Conference, ISBN: 978-1-4244-2309-5, Las Vegas, Nevada, USA, January 2009
- 15.) Cornelius, Claas; Sill, Frank; Sämrow, Hagen; Salzmann, Jakob; Timmermann, Dirk; da Silva, Diógenes; Encountering gate oxide breakdown with shadow transistors to increase reliability; 21st Symposium on Integrated Circuits and Systems Design (SBCCI), S. 111-116, ISBN: 978-1-60558-231-3, Gramado, Brasilia, September 2008
- 16.) Lieckfeldt, Dominik; You, Jiaxi; Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; Characterizing the energy efficiency of localization algorithms in wireless sensor networks; International Conference on Wireless Communications and Mobile Computing, IWCMC '09, pp. 839-843, ISBN: 978-1-60558-569-7, Leipzig, Germany, June 2009
- 17.) Lieckfeldt, Dominik; You, Jiaxi; Behnke, Ralf; Salzmann, Jakob; Timmermann, Dirk; Assessing the Energy Efficiency of

Localization in Wireless Sensor Networks; 6th Annual IEEE Consumer Communications and Networking Conference, ISBN: 978-1-4244-2309-5, Las Vegas, USA, January 2009

- 18.) Reichenbach, Frank; Born, Alexander; Salzmann, Jakob; Timmermann, Dirk; Bill, Ralf; DLS: A Resource-Aware Localization Algorithm with High Precision in Large Wireless Sensor Networks; In Proceedings of the 4th Workshop on Positioning, Navigation and Communication (WPNC'07), pp. 247-254, ISBN: 1-4244-0870-9, Hannover, Deutschland, March 2007
- 19.) Salzmann, Jakob; Behnke, Ralf; Gorski, Philipp; Timmermann, Dirk; HyPAERLoc: Plausible Hybrid Localization for Wireless Sensor Networks; Proceedings of 5th International Conference on Sensor Technologies and Applications (SENSORCOMM'11) (in press), ISBN: 978-1-61208-010-9, Nice, France, August 2011
- 20.) Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; Hex-MASCLE - Hexagon based Clustering with Self Healing Abilities; IEEE Wireless Communications and Networking Conference, pp. 1339-1344, ISBN: 978-1-61284-253-0, Cancun, Mexico, March 2011
- Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; Tessellating Cell Shapes for Geographical Clustering; 10th IEEE International Conference on Computer and Information Technology (CIT 2010), pp. 2891-2896, ISBN: 978-0-7695-4108-2, Bradford, United Kingdom, June 2010
- 22.) Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; Redundancy Aware Clustering via Centroid Localization Technique; Architecture of Computing Systems ARCS, pp. 359-365, ISBN: 978-3-8007-3222-7, Hannover, Germany, February 2010
- 23.) Salzmann, Jakob; Behnke, Ralf; You, Jiaxi; Timmermann, Dirk; Free-CLASH – Improved Localization Free Clustering in Large Wireless Sensor Networks; The International Workshop on Scalable Ad Hoc and Sensor Networks, ISBN: 978-1-4244-3941-6, St. Petersburg, Russia, October 2009
- 24.) Salzmann, Jakob; Behnke, Ralf; Gag, Martin; Timmermann, Dirk; 4-MASCLE - Improved Coverage Aware Clustering with Self Healing Abilities; The International Symposium on Multidisciplinary Autonomous Networks and Systems (MANS 2009), pp. 537-543, ISBN: 978-0-7695-3737-5/09, Brisbane, Australia, July 2009

- 25.) Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; A Self-Organized Localization-Free Clustering Approach for Redundancy Exploitation in Large Wireless Sensor Networks; GI Jahrestagung, Workshop: Adaptive und organische Systeme, pp. 747-754, ISBN: 978-3-88579-228-4, München, Germany, September 2008
- 26.) Salzmann, Jakob; Behnke, Ralf; Lieckfeldt, Dominik; Timmermann, Dirk; 2-Mascle - A Coverage Aware Clustering Algorithm with Self Healing Abilities; 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISBN: 1-4244-1502-0, Melbourne, Australia, December 2007
- Salzmann, Jakob; Sill, Frank; Timmermann, Dirk; Algorithm for Fast Statistical Timing Analysis; International Symposium on System-on-Chip, pp. 90-93, ISBN: 1-4244-1367-2, Tampere, Finnland, November 2007
- Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; Analyse regelmäßiger Clusterformen in Sensornetzwerken; 12. Symposium Maritime Elektrotechnik, Elektronik und Informationstechnik, S. 375 -380, Rostock, Deutschland, Oktober 2007
- 29.) Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; Geographical Clustering with Coarse-Grained Localization; 5th International Forum "Life Science Automation", ISBN: 978-3-938042-12-0, Landsdowne, Virginia, USA, October 2007
- 30.) Salzmann, Jakob; Kubisch, Stephan; Reichenbach, Frank; Timmermann, Dirk; Energy and Coverage Aware Routing Algorithm in Self Organized Sensor Networks; Proceedings of Fourth International Conference on Networked Sensing Systems, pp. 77-80, ISBN: 1-4244-1231-5, Braunschweig, Germany, June 2007
- 31.) Sämrow, Hagen; Cornelius, Claas; Gorski, Philipp; Salzmann, Jakob; Timmermann, Dirk; Effiziente Simulation von Gateoxiddefekten auf Gatterebene mit Transistorlevel-Genauigkeit; 15th Workshop Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (MBMV), pp. 157-168, ISBN: 978-3-8300-6201-1, Kaiserslautern, Germany, March 2012
- 32.) Sämrow, Hagen; Cornelius, Claas; Gorski, Philipp; Salzmann, Jakob; Tockhorn, Andreas; Timmermann, Dirk; Functional Enhancements of TMR for Power Efficient and Error Resilient ASIC Designs; 14th IEEE Symposium on Design and Diagnostics of Electronic Cir-

cuits and Systems, pp. 183-188, ISBN: 978-1-4244-9754-6, Cottbus, Germany, April 2011

- 33.) Sämrow, Hagen; Cornelius, Claas; Salzmann, Jakob; Tockhorn, Andreas; Timmermann, Dirk; Utilizing Parallelism of TMR to Enhance Power Efficiency of Reliable ASIC Designs; 6th International Conference on Computer Engineering and Systems, ICCES'10, pp. 251-256, ISBN: 978-1-4244-7041-9, Kairo, Egypt, December 2010
- 34.) You, Jiaxi; Lieckfeldt, Dominik; Salzmann, Jakob; Timmermann, Dirk; GAF&Co: Connectivity Aware Topology Management for Sensor Networks; 20th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC2009), ISBN: 978-1-4244-5123-4, Tokyo, Japan, September 2009
- 35.) You, Jiaxi; Lieckfeldt, Dominik; Han, Qi; Salzmann, Jakob; Timmermann, Dirk; Look-ahead Geographic Routing for Sensor Networks; 5th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS 2009), ISBN: 978-1-4244-3304-9, Galveston, Texas, USA, March 2009

Buchbeiträge

36.) Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; OC Principles in Wireless Sensor Networks; C. Müller-Schloer et. al. (Editors): Organic Computing — A Paradigm Shift for Complex Systems, pp. 503-516, ISBN: 978-3-0348-0129-4, Germany, April 2011

Journalbeiträge

- 37.) You, Jiaxi; Han, Qi; Lieckfeldt, Dominik; Salzmann, Jakob; Timmermann, Dirk; Virtual position based geographic routing for wireless sensor networks; Computer Communications, Elsevier Journal on, pp. 1255-1265, Kidlington, United Kingdom, July 2010
- Behnke, Ralf; Salzmann, Jakob; Gorski, Philipp; Timmermann, Dirk; Evaluation of Hybrid Distributed Least Squares for Improved Localization via Algorithm Fusion in Wireless Sensor Networks; Sensors & Transducers Journal (ISSN: 1726-5479), Vol.14-2, Special Issue, March 2012, pp.179-195, March 2012

Fachvorträge

 Salzmann, Jakob; Timmermann, Dirk; Energy Aware Self Organized Communication in Complex Networks; SPP 1183 "Organic Computing" 3. Kolloqium, Stuttgart, Germany, September 2006

- 40.) Salzmann, Jakob; Timmermann, Dirk; Organic Principles in Complex Networks; SPP 1183 "Organic Computing" 5. Kolloqium, Lübeck, Germany, September 2007
- 41.) Salzmann, Jakob; Timmermann, Dirk; Organic Principles in Sensor Networks; 6th colloquium of the SPP 1183 Organic Computing, Paderborn, Germany, March 2008
- 42.) Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; Organic Algorithms for Complex Networks; 7th colloquium of the SPP 1183 Organic Computing, Zürich, Swiss, September 2008
- 43.) Salzmann, Jakob; Behnke, Ralf; Timmermann, Dirk; Organic Sensor Networks; SPP 1183 "Organic Computing" 8. Kolloquium, Bochum, Germany, Februar 2009