

Effiziente Berechnung von Signal-zu-Rausch-Verhältnissen in simulierten Sensornetzwerken

Max Pagel

7. Januar 2010

Universität zu Lübeck
Institut für Telematik

Masterarbeit

Effiziente Berechnung von
Signal-zu-Rausch-Verhältnissen in simulierten
Sensornetzwerken

von
cand. inform. Max Pagel

Aufgabenstellung und Betreuung:

Prof. Dr. S. Fischer, Dr. A. Kröller, Dr. D. Pfisterer

Lübeck, den 7. Januar 2010

Erklärung

Ich versichere, die vorliegende Arbeit selbstständig und nur unter Benutzung der angegebenen Hilfsmittel angefertigt zu haben.

Lübeck, den 7. Januar 2010

Kurzfassung

Bei Forschungen im Bereich von Sensornetzwerken werden neue Verfahren in der Regel erst mit Hilfe von Simulatoren evaluiert, bevor sie in praktischen Tests überprüft werden. Einer dieser Simulatoren ist der Simulator SHAWN. In dieser Arbeit wird gezeigt, dass die bisherige Simulation des CSMA/CA Algorithmus zur Verhinderung von Kollisionen in SHAWN aufgrund zu hoher Abstraktion von der Wirklichkeit unrealistische Ergebnisse liefert. Um dies zu beheben werden mehrere Verfahren zur effizienten und genauen Simulation von Signalstärken und Signal-zu-Rausch-Verhältnissen entwickelt, die sich in Simulationsgenauigkeit und Laufzeit unterscheiden. Diese Verfahren wurden beispielhaft in SHAWN implementiert und evaluiert. Es konnte gezeigt werden, dass eines dieser Verfahren eine hohe Genauigkeit bei guten Laufzeiteigenschaften bietet.

Aufgabenstellung

Drahtlose Sensornetzwerke (engl. Wireless Sensor Networks, WSNs) sind heterogene Systeme, die aus kleinen, mit stark begrenzten Ressourcen ausgestatteten Sensorknoten sowie Gateways und Backend-Systemen bestehen. In die Umwelt eingebettet erfassen Sensorknoten physikalische Parameter (wie z.B. Temperatur, Helligkeit oder Bewegung) und leiten diese an Gateways weiter. Gateways senden die erfassten Daten über herkömmliche Netzwerke zu Backend-Systemen, wo sie schließlich weiterverarbeitet und visualisiert werden. Die Realisierung von Anwendungen für drahtlose Sensornetzwerke ist komplex, da hierbei Herausforderungen verteilter Anwendungen als auch eingebetteter Systeme zu lösen sind. Erschwert wird dies durch die erwähnte Heterogenität, unvorhersehbare Umwelteinflüsse sowie die Größe der Netze. Um diese Herausforderungen zu meistern verwenden Entwickler typischerweise Simulationen, um Anwendungen in einer kontrollierbaren Umgebung zu testen und zu optimieren.

Zur Simulation von Sensornetzwerken existieren verschiedenste Simulationswerkzeuge mit unterschiedlichen Eigenschaften. Ns-2 beispielsweise ist auf die möglichst realitätsnahe Simulation von Protokollstacks bis zum MAC Layer und die Modellierung von Funkübertragungen spezialisiert. Ein alternativer Ansatz wird von Shawn verfolgt. Eine der grundlegenden Ideen ist es, den Effekt von Phänomenen zu simulieren und nicht die Phänomene selbst. Anstatt z.B. ein komplettes MAC-Protokoll (inkl. Der Funkausbreitung) zu simulieren, modelliert Shawn seine Effekte auf höhere Ebenen (z.B. Verlust, Verfälschung und Verzögerung von Nachrichten). Simulationen werden dadurch aufgrund der im Detail bekannten Modellierung wiederholbarer und aussagekräftiger. Oft können diese Modelle sehr effizient implementiert werden, wodurch die Simulationszeit erheblich reduziert wird und bis dato kaum erreichbare Knotenzahlen handhabbar werden.

In dieser Arbeit soll diese Shawn zugrundeliegende Idee auf die Implementierung von Medienzugriffsprotokollen übertragen werden. Klassischerweise verwenden Simulatoren eine sehr aufwändige Modellierung der Signalausbreitung um das Signal-zu-Rausch-Verhältniss zu bestimmen und um damit wiederum die Kollisionen von Nachrichten auf dem Übertragungsmedium zu erkennen. Eine solche rechenzeitintensive Modellierung widerspricht jedoch den Grundsätzen von Shawn, da sie implizit auch die Anzahl simulierbarer Knoten reduziert. Aktuell existiert daher eine vereinfachte CSMA/CA Implementierung in Shawn, die jedoch aufgrund der von ihr vorgenommenen Abstraktionen kein gutes Abbild der Wirklichkeit darstellt. Es sollen daher effizientere Ansätze zur Simulation von CSMA/CA in Shawn untersucht werden. Die entwickelten Ansätze sollen mit der bestehenden Implementierung verglichen werden und es soll aufgezeigt werden, inwiefern die entwickelten Ansätze eine Verbesserung der Genauigkeit erreichen und welche Auswirkungen dies auf die Simulationsgeschwindigkeit hat.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	3
2.1	Drahtlose Ad-hoc Netzwerke	3
2.2	Signalübertragung in drahtlosen Netzwerken	4
2.2.1	Kollisionen bei der drahtlosen Übertragung von Nachrichten	5
2.3	Der IEEE 802.15.4-Standard für Low-Rate Wireless Personal Area Networks	7
2.4	Unslotted CSMA/CA nach IEEE 802.15.4	9
2.4.1	Funktionsweise des CSMA/CA-Algorithmus	9
2.5	Simulation drahtloser Ad-hoc Netzwerke	12
2.5.1	SHAWN	12
2.5.2	Architektur von SHAWN	13
3	Simulationsansätze	17
3.1	Übertragungsablauf während der Simulation	17
3.1.1	CCA, Schritt 1	17
3.1.2	Kollisionsbehandlung, Schritt 2	18
3.1.3	Ermitteln neuer Empfänger, Schritt 3	18
3.1.4	Beenden des Sendevorgangs, Schritt 4	18
3.2	Das bisherige CSMA/CA-Modell in SHAWN	18
3.2.1	Übertragungsablauf im bisherigen CSMA/CA-Modell	19
3.3	Konzepte zur Verbesserung des bisherigen CSMA/CA-Modells	21
3.3.1	Übertragungsablauf mit Rauschberechnung	22
3.4	Architektur der neuen Transmissionsmodelle	26
3.4.1	NoSNR Ansatz	26
3.4.2	SimpleSNR Ansatz	27
3.4.3	Wahl des Rauschradius unter Berücksichtigung der aktuellen Simulationssituation	28
3.4.4	ExtendedSNR Ansatz	31
4	Implementierungsdetails	35
4.1	Anforderungen an die zugrunde liegende Datenstruktur	35
4.2	k-d-Bäume als zugrundeliegende Datenstruktur	36
4.2.1	Aufbau eines k-d-Baumes	36
4.2.2	Dynamische Eigenschaften eines K-d-Baums	41
4.2.3	Orthogonale Bereichsanfragen an einen k-d-Baum	44
4.2.4	Approximation des Rauschradius	44
4.3	Strategien zur schrittweisen Erweiterung des Rauschradius	46

Inhaltsverzeichnis

5	Evaluation	49
5.1	Anzahl der Knoten in der Simulationswelt	49
5.2	Evaluations-Szenarien	50
5.2.1	Hello-World	50
5.2.2	Flood	51
5.3	Evaluation der Signalsimulation	51
5.3.1	Simulation der Verteilung des Gesamtrauschens	52
5.3.2	Simulation der Signal-zu-Rausch-Verhältnisse	56
5.4	Vergleich der Transmissionsmodelle untereinander	60
5.4.1	Gütekriterien	60
5.4.2	Geeignete Größe des Rauschradius	60
5.4.3	Vergleich der Kollisionswahrscheinlichkeit bei verschiedenen Simulationsszenarien	65
5.4.4	Laufzeitauswertungen	67
6	Zusammenfassung und Ausblick	71
	Abbildungsverzeichnis	73
	Literaturverzeichnis	77

1 Einleitung

Im Zuge der letzten Jahre wurden vermehrt Forschungen im Bereich von Ad-hoc Sensornetzwerken durchgeführt. Bei Ad-Hoc Sensornetzwerken handelt es sich um drahtlose Netzwerke zwischen Geräten, die mit Sensoren ausgestattet sind, z.B. zum Messen von Temperaturen. Diese Geräte, auch Sensorknoten genannt, können so ohne vorhandene Infrastruktur benutzt werden um größere Gebiete zu überwachen. Dabei werden Daten, die von Sensorknoten gesendet werden, die außerhalb der Funkreichweite einer Basisstation liegen, von Knoten zu Knoten weitergereicht, bis sie die Basisstation erreichen.

Um diese Anforderungen effizient und flexibel zu erfüllen, werden fortgeschrittene Routingtechniken und Protokolle benötigt, die den Datenverkehr innerhalb des Netzwerks regeln. Um diese zu entwickeln, wird in der Regel auf Netzwerksimulatoren zurückgegriffen, wie z.B. den Netzwerk Simulator SHAWN. Der Simulator SHAWN versucht eine algorithmische Sicht auf die Simulation zu ermöglichen, indem er verschiedene Aspekte der Nachrichtenübertragung in der Realität durch einfachere, meist statistische Modelle abstrahiert. Auf diese Weise werden Simulationen mit sehr hohen Knotenzahlen möglich, die bei einem Versuch der vollständigen Simulation der realen Nachrichtenübertragung aufgrund der damit verbundenen Komplexität und extremen Laufzeiten nur schwer möglich wären.

In der vorliegenden Arbeit wird nachgewiesen, dass SHAWN bisher bei der Simulation des CSMA/CA-Algorithmus zur Kollisionsvermeidung während der physikalischen Übertragung einer Nachricht zwischen zwei Knoten, einen zu hohen Abstraktionsgrad verwendet. Bisher wurden Eigenschaften wie Signalstärken und Signal-zu-Rausch-Verhältnisse bei der Nachrichtenübertragung nicht berücksichtigt. Es wird nachgewiesen, dass dies die Simulationsergebnisse stark verfälscht. Um diese Defizite zu beheben, werden im Rahmen dieser Arbeit mehrere Verfahren zur laufzeiteffizienten Simulation von Signal-zu-Rausch-Verhältnissen entwickelt. Diese Verfahren werden zur Evaluation exemplarisch in SHAWN implementiert. Es wird gezeigt, dass mit Hilfe eines dieser Verfahren eine hohe Simulationsgenauigkeit bei guten Laufzeiteigenschaften erreicht werden kann. Das beschriebene Verfahren bedient sich dabei geschickter Abschätzungen der Ausbreitung von Funksignalen, um effizient und genau Signal-zu-Rausch-Verhältnisse zu berechnen.

1 Einleitung

2 Grundlagen

Im folgenden Kapitel wird eine grundlegende Einführung in die drahtlose Übertragung von Nachrichten gegeben mit speziellem Augenmerk auf die Anwendung bei drahtlosen Sensornetzwerken im Rahmen des IEEE 802.15.4 Standards. Im weiteren Verlauf wird die Simulationsumgebung SHAWN vorgestellt, mit der die Simulation drahtloser Netzwerke durchgeführt werden kann.

2.1 Drahtlose Ad-hoc Netzwerke

Bei sogenannten drahtlosen Ad-hoc Netzwerken handelt es sich um Funknetzwerke, die aus zwei oder mehr Empfängern, auch Knoten genannt, ein vermaschtes Netz ausbilden. Das besondere an Ad-hoc Netzwerken ist, dass sie im Gegensatz zu herkömmlichen Funknetzwerken, wie z.B. Private WLANs, keine zentralen Kontrollstellen wie Access Points oder Router benötigen. Stattdessen werden Nachrichten von Knoten zu Knoten weitergereicht, bis sie ihr Ziel erreicht haben. Jeder Knoten fungiert in diesem Zusammenhang als eigenständiger Router.

Aufgrund dieser dezentralen Struktur bieten Ad-hoc Netzwerke viele Vorteile, wie selbständige Konfiguration und Optimierung, sowie, aufgrund der Vermaschtheit, hohe Toleranz gegenüber dem Ausfall einzelner Knoten. Um diese Vorteile nutzen zu können, müssen jedoch relativ komplexe Routingalgorithmen angewandt werden, die den Nachrichtenfluss kontrollieren.

In den letzten Jahren wurde in diesem Zusammenhang vermehrt Forschung im Bereich so genannter Sensornetzwerke betrieben. Bei Sensornetzwerken handelt es sich um Ad-hoc Netzwerke, die aus mit Sensoren ausgestatteten Kleingeräten bestehen, die gleichzeitig einen Funktransmitter zur drahtlosen Übertragung von Daten zur Verfügung stellen. Durch Benutzung solcher Sensorknoten lassen sich z.B. sehr einfach große Gebiete überwachen, ohne dass vorher Infrastrukturmaßnahmen vorgenommen werden müssen. In diesem Zusammenhang sei z.B. auf das FleGSens Projekt hingewiesen ¹.

Speziell bei Sensornetzwerken, die in der Regel darauf ausgelegt sind große Bereiche zuverlässig über einen langen Zeitraum zu überwachen, ist es wichtig, dass beim Design der Routingalgorithmen und Protokolle insbesondere Aspekte wie Stromverbrauch und Ausfallsicherheit berücksichtigt werden. Hierbei ist es besonders wichtig, den Funkverkehr auf das notwendigste zu beschränken, da die drahtlose Übertragung von Nachrichten einer der Hauptfaktoren beim Stromverbrauch der Sensorknoten ist. Aus diesem Grund sollten auch Kollisionen von Nachrichten bei der Übertragung ver-

¹ [1]FleGSens - secure area monitoring using wireless sensor networks

2 Grundlagen

mieden werden, da sie eine wiederholte Übertragung der Daten notwendig machen.

2.2 Signalübertragung in drahtlosen Netzwerken

Der folgende Abschnitt soll eine kurze Einführung in die für diese Arbeit relevanten Funktionsweisen der drahtlosen Nachrichtenübertragung liefern. Wer sich tiefergehender für dieses Thema interessiert sei z.B. auf Martin Werner², oder auf Qizheng Gu verwiesen³.

Die drahtlose Übertragung von Nachrichten innerhalb eines Funknetzwerkes geschieht durch die Modulation eines Trägersignals auf der Senderseite, welches auf der Empfängerseite demoduliert wird. Als Trägersignal wird in diesem Fall eine periodisch sich ändernde physikalische Größe bezeichnet, im allgemeinen eine Funkwelle. Addiert man nun zu diesem Trägersignal auf geeignete Weise ein Nutzsignal, z.B. ein Datensignal, so spricht man von Modulation. Das Nutzsignal verändert so die Trägerwelle, was von einem Empfänger wahrgenommen wird. Der Empfänger kann nun das empfangene modulierte Signal demodulieren und so das ursprüngliche Nutzsignal wieder erhalten. Auf diese Weise wurden die Nutzdaten vom Sender an den Empfänger übertragen.

Ob die Demodulation erfolgreich durchgeführt werden kann, hängt maßgeblich von der Signalstärke des Signals ab und vom Signal-zu-Rausch-Verhältnis. Bei der drahtlosen Kommunikation kann es so, wie bei der Kommunikation im wirklichen Leben dazu kommen, dass Nachrichten nicht verstanden werden. Dies kann daran liegen, dass die Kommunikationspartner zu weit voneinander entfernt sind um das Signal zu verstehen, die Signalstärke oder die Empfängerempfindlichkeit also nicht ausreichend sind. Dies hängt damit zusammen, dass die Ausbreitung von Funksignalen einem exponentiellen Abfall unterliegt. Dies wird in der Regel anhand der in Definition 2.2.1 eingeführten Gleichung modelliert.

Ein anderes Phänomen, das selbst bei eigentlich ausreichender Signalstärke dazu führen kann, dass Nachrichten nicht verstanden werden, ist ein zu hohes Umgebungsrauschen. Grob vereinfacht bedeutet dies, dass, viele Nachrichten nicht mehr verstanden werden können, wenn zu viele Kommunikationspartner gleichzeitig reden. Dies liegt darin begründet, dass ein Empfänger die Überlagerung aller Signale, die um ihn herum gesendet werden, gleichzeitig wahrnimmt. Dadurch kann es zu einer starken Veränderung des modulierten Signals der Nachricht kommen, die der Empfänger gerade demoduliert. Diese Veränderung kann so stark sein, dass eine erfolgreiche Demodulation nicht mehr möglich ist. Man spricht in einem solchen Fall von einer Kollision, da die Nachricht auf dem Empfänger mit einer oder mehreren anderen Nachrichten "kollidiert" ist.

In diesem Zusammenhang wird oft vom *Signal-zu-Rausch-Verhältnis* bzw. vom "Signal to Noise Ratio" (SNR) gesprochen. Das Signal-zu-Rausch-Verhältnis beschreibt wie stark sich ein Signal vom Umgebungsrauschen abhebt, welches durch andere Sender verursacht wird. Das Signal-zu-Rausch-Verhältnis wird generell in Dezibel (*dB*)

² [2] Nachrichtentechnik: Eine Einführung für alle Studiengänge

³ [3] RF system design of transceivers for wireless communications

2.2 Signalübertragung in drahtlosen Netzwerken

angegeben. Die theoretische Berechnung des Signal-zu-Rausch-Verhältniss wird ebenfalls in Definition 2.2.1 eingeführt.

Definition 2.2.1 (Signalberechnung) $p(x)$ bezeichne die Sendestärke eines Senders im Punkt x .

Die maximale Sendeleistung aller Knoten sei mit p_{max} bezeichnet.

Funktion 2.1 beschreibt die Signalstärke eines Signals im Punkt y , welches von Punkt x aus mit Stärke $p(x)$ gesendet wird.

$$p(x, y) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}, \text{ mit } p(x, y) = p(x) \|x - y\|^{-\alpha} \quad (2.1)$$

In der Regel gilt $\alpha = 2$.

Funktion 2.2 beschreibt das Signal zu Rausch Verhältnis eines Signals in dB, ausgehend von einem Sender s_j , im Punkt y zum Zeitpunkt t . Die s_i mit $i \in \mathbb{N}$ bezeichnen alle anderen Sender ohne s_j von denen der Empfänger im Punkt y Signale wahrnehmen kann.

$$SNR_t(s_j, y) : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}, \text{ mit } SNR_t(s_j, y) = 10 \log_{10} \frac{p(s_j, y)}{\sum_{s_i \in S_t, i \neq j} p(s_i, y)} \quad (2.2)$$

2.2.1 Kollisionen bei der drahtlosen Übertragung von Nachrichten

Im Gegensatz zu kabelgestützten Netzwerken können Kollisionen bei drahtlosen Netzen von den Sendern nicht erkannt werden. Lediglich Empfänger haben evtl. die Möglichkeit eine Kollision zu erkennen. Dies hängt damit zusammen, dass die Empfangseinheit eines Senders beim Senden ausgeschaltet wird, da sie während des Sendens ohnehin ausschließlich das eigene Signal empfangen würde. Dieses Signal wäre so stark, dass sämtliche anderen Signale, die zum gleichen Zeitpunkt übertragen werden, überlagert würden. Der Sender ist dementsprechend während des Sendens einer Nachricht "taub" im Hinblick auf Übertragungen aus dem restlichen Netzwerk, es ist ihm also nicht möglich aufgrund des Netzwerkverkehrs darauf zu schließen ob eine Kollision stattgefunden hat.

Selbst gesetzt den Fall, dass der Sender in der Lage wäre gleichzeitig zu senden und zu empfangen, so wären Vorhersagen von Kollisionen auf Empfängern schwer möglich, da sich das Netzwerk an der Position des Empfängers in einem völlig anderen Zustand befinden kann. Aus diesem Grund ist es höchstens Empfängern möglich Kollisionen zu erkennen, z.B. durch die Übertragung von Prüfsummen oder speziellen Prüfbits, mit denen sich die korrekte Demodulation des Signals überprüfen lässt. Dies setzt natürlich voraus, dass die Prüfsummen korrekt übertragen werden. Es ist denkbar, dass aufgrund von zu hohem Umgebungsrauschen eine Nachricht von Anfang an nicht verstanden werden kann. Ein Empfänger würde in diesem Fall nicht einmal merken, dass eine Nachricht bei ihm kollidiert ist.

Aufgrund der oben beschriebenen Eigenschaften drahtloser Netzwerke gehen bei einer Kollision, vom Sender unbemerkt, alle gesendeten Daten verloren und es liegt in

2 Grundlagen

der Verantwortung des Empfängers, so er die Kollision erkennt, die Daten erneut anzufordern. In der Regel wird dieses Verhalten indirekt durch das Senden eines "Acknowledgement" (ACK) Signals des Empfängers erledigt, das gesendet wird, wenn eine Übertragung erfolgreich war. Empfängt ein Sender innerhalb eines bestimmten Zeitraumes nach dem Abschluss einer Übertragung kein ACK Signal, so muss er davon ausgehen, dass die Übertragung fehlgeschlagen ist und er die Daten erneut übertragen muss.

Kollisionen führen dazu, dass die Übertragung von Nachrichten stark verlangsamt wird und zusätzlich aufgrund der wiederholten Übertragung die Ressourcen des Netzwerks strapaziert werden. Sie sollten deswegen nach Möglichkeit vermieden werden. Hierzu gibt es verschiedene Algorithmen, wie z.B. den "Carrier Sense Multiple Access with Collision Avoidance", kurz CSMA/CA-Algorithmus. Der CSMA/CA-Algorithmus regelt, wann ein Knoten idealerweise sendet, um so wenig Kollisionen wie möglich zu erzeugen.

Grob gesprochen ermittelt der Algorithmus den richtigen Zeitpunkt zum Senden, indem er zunächst ein sogenanntes "Clear Channel Assessment" (CCA) durchführt. Während des CCAs lauscht der Sender, dies wird "Carrier Sense" genannt, ob nicht gerade zu viele Übertragungen das Netzwerk belasten. Ist dies der Fall, so ist das Risiko, durch eine zusätzliche Übertragung Kollisionen zu erzeugen, hoch. Mit diesem Verhalten wird dem Umstand Rechnung getragen, dass in einem Funknetzwerk mehrere Sender gleichzeitig senden dürfen, also sogenannter "Multiple Access" möglich ist. Sind zu viele Sender aktiv, so wartet der Sender eine zufällige Zeitspanne bevor er erneut lauscht. Andernfalls beginnt er mit dem Senden. Dieser Ablauf wird als "Collision Avoidance" bezeichnet, da versucht wird Kollisionen von vorn herein zu vermeiden.

Der CSMA/CA Algorithmus ist in der Lage viele Kollisionen zu verhindern, es gibt allerdings Situationen die trotz der Anwendung des CSMA/CA Algorithmus zu Kollisionen führen können, wie sog. 'hidden-station' Kollisionen oder Kollisionen durch gleichzeitiges Senden. Diese werden im Folgenden vorgestellt.

2.2.1.1 Hidden-Station-Kollisionen

Abbildung 2.1 illustriert eine Kollision die durch eine sogenannte "Hidden-Station" Situation ermöglicht wird. Unter einer "Hidden-Station" versteht man einen Sender k_h , der außerhalb des Wahrnehmungsbereichs des Knotens k_s liegt, der gerade senden will. Der Empfänger k_e der Nachricht von k_s liegt zwischen den beiden Knoten k_h und k_s . Wenn k_s nun lauscht, ob der Kanal belegt ist, so erhält er das Ergebnis, dass der Kanal frei ist. Dies liegt daran, dass k_h zu weit entfernt ist um von k_s wahrgenommen zu werden. k_s sendet also seine Nachricht. Diese Nachricht kann von k_e jedoch nicht empfangen werden, da k_e ähnlich weit entfernt ist wie k_h , die Nachricht von k_s weist also kein genügend großes Signal-zu-Rausch-Verhältnis auf, um verstanden zu werden. Es kommt zu einer Kollision.

2.3 Der IEEE 802.15.4-Standard für Low-Rate Wireless Personal Area Networks

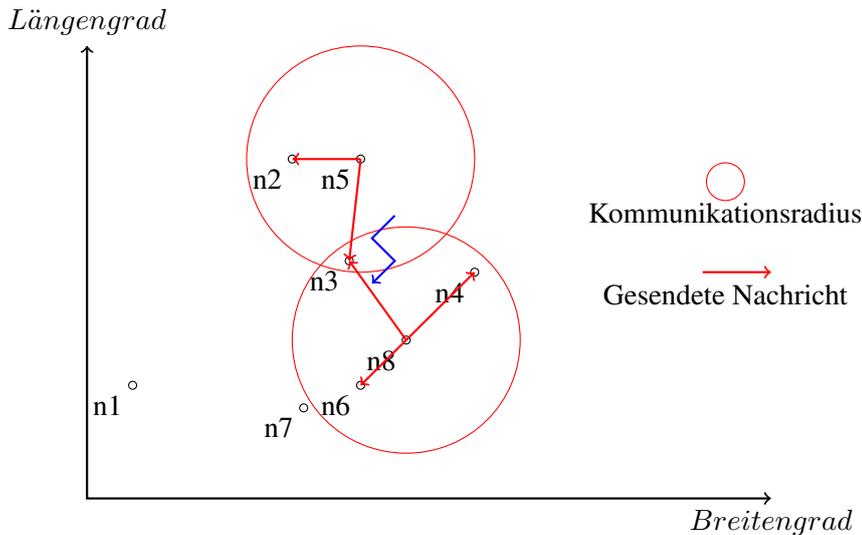


Abbildung 2.1: Hidden Station Kollision. n8 erhält vom CCA das Ergebnis "Kanal ist frei", da er außerhalb des Kommunikationsradius von n5 liegt. n8 sendet also seine Nachricht. Da n5 allerdings gerade ebenfalls eine Nachricht sendet und n3 im Kommunikationsradius von n5 liegt, geht die Nachricht von n8 bei n3 im Rauschen des Sendens von n5 unter. Die Nachrichten sind also auf dem Knoten n3 kollidiert.

2.2.1.2 Kollisionen durch gleichzeitiges Senden

Im Rahmen des CSMA/CA-Algorithmus kann es vorkommen, dass zwei Knoten exakt gleichzeitig mit dem Senden beginnen. Liegen diese Knoten zu nahe beieinander, so ist es sehr wahrscheinlich, dass ihre Nachrichten auf einem Großteil der Empfänger kollidieren. Eine solche Situation ist in Abbildung 2.2 dargestellt. Glücklicherweise kommen Kollisionen dieser Art nur sehr selten vor, so dass sie bei den meisten Simulationen vernachlässigt werden. Sie wären ohnehin nur äußerst schwer zu simulieren, da ein Computer in der Regel seriell arbeitet. Kollisionen durch paralleles Senden zu simulieren ist somit nicht ohne weiteres möglich.

2.3 Der IEEE 802.15.4-Standard für Low-Rate Wireless Personal Area Networks

Im Bereich der Sensornetze hat sich unter anderem der WLAN Standard IEEE 802.15.4 durchgesetzt, der Protokoll und Zusammenschaltung von Geräten via Funkkommunikation in einem "personal area network" (PAN) ⁴ spezifiziert. Im Gegensatz zu wireless local area networks (WLAN) werden WPANs zur Übertragung von Daten über relativ geringe Entfernungen benutzt, wobei keine oder so gut wie keine Infrastruktur vorhanden sein muss, damit das Netzwerk arbeiten kann. Auf diese Weise

⁴ [4, S. iii] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

2 Grundlagen

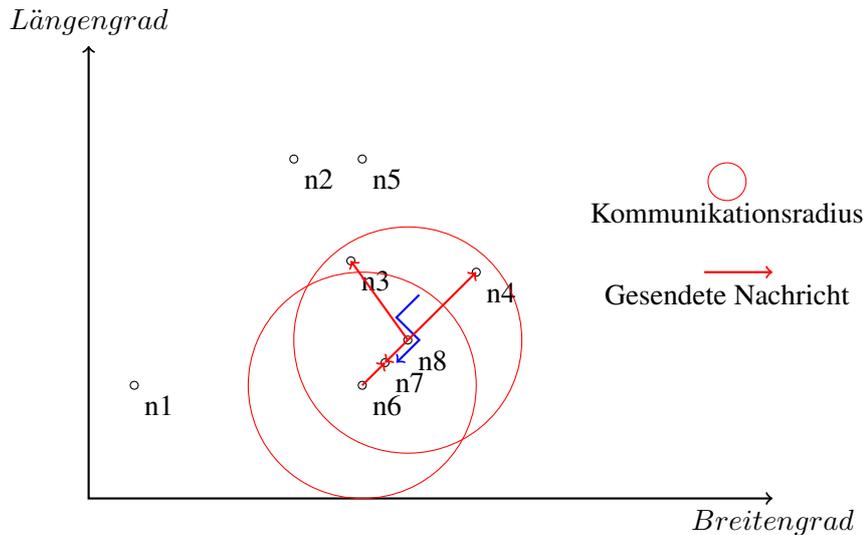


Abbildung 2.2: Kollision durch gleichzeitiges Senden. n8 und n6 haben gleichzeitig ein CCA durchgeführt und den Kanal so als frei wahrgenommen. Sie fangen nach Abschluss des CCAs gleichzeitig an zu senden. Hierdurch kommt es zu einer Kollision der beiden Signale auf n7, da sich die Signale überlagern.

sollen kleine und energieeffiziente Lösungen möglich werden, die in vielen Bereichen Anwendung finden können.

Der Standard beschreibt unter anderem, welche technischen Parameter von Geräten erfüllt werden müssen, die sich selbst als IEEE 802.15.4 konform bezeichnen wollen. Einige dieser Parameter werden in Definition 2.3.1 verwendet.

Definition 2.3.1 (Transmissionsparameter) p_{min} bezeichne die Empfängerempfindlichkeit, also die minimale Signalstärke, bis zu der ein Signal von einem Empfänger erfolgreich demoduliert werden kann. Dies setzt voraus, dass keinerlei störendes Rauschen, z.B. durch andere Sender vorhanden ist.

SNR_{min} beschreibe das minimale Signal-zu-Rausch-Verhältnis, das eine Übertragung aufweisen muss um von einem Empfänger verstanden werden zu können.

Im Rahmen dieser Arbeit wurden die Transmissionsparameter entsprechend der im IEEE 802.15.4-Standard definierte Wert gewählt. Diese werden in der folgenden Tabelle dargestellt

p_{min}^a	mind. $-85dBm = 3.16 \cdot 10^{-9}mW$
p_{max}^b	$0dBm = 1mW \forall x$
SNR_{min}^c	4

^a [4, S. 49] IEEE 802.15.4

^b [4, S. 263] IEEE 802.15.4

^c [4, S. 269] IEEE 802.15.4

Eine umfassende Erläuterung der im IEEE 802.15.4 Standard vorgestellten Techno-

logien liegt außerhalb des Rahmens dieser Arbeit. Von Interesse für die vorliegenden Untersuchungen ist lediglich der eingangs schon erwähnte und vom Standard angewandte CSMA/CA Algorithmus zur Koordination des Netzwerkverkehrs und der Kollisionsvermeidung. Der Algorithmus wird in Abschnitt 2.4 vorgestellt.

2.4 Unslotted CSMA/CA nach IEEE 802.15.4

Der CSMA/CA-Algorithmus nach IEEE 802.15.4-Standard⁵ wird in zwei Varianten beschrieben, der *slotted* und der *unslotted* Variante. Bei der *slotted* Variante wird jedem Knoten im Laufe des Algorithmus ein festgeschriebener Zeitraum (Slot) zum Senden der Nachricht zugewiesen. Um dies zu koordinieren werden periodisch Signale von sog. Netzwerkkoordinatoren ausgesendet, die es ermöglichen die Slots über alle Knoten innerhalb der Reichweite des Netzwerkkoordinators zu synchronisieren. Die *slotted* Version des CSMA/CA-Algorithmus wird in dieser Arbeit nicht weiter behandelt, da sie sehr viel weniger verbreitet ist, als die *unslotted* Variante des Algorithmus. Wer sich für eine detaillierte Beschreibung interessiert sei auf den IEEE 802.15.4-Standard verwiesen. Die in dieser Arbeit vorgestellten Techniken lassen sich bei Bedarf mit relativ wenig Aufwand, auch für eine Verwendung bei der Simulation des *slotted* CSMA/CA-Algorithmus anpassen.

Im Rahmen dieser Arbeit wird eine effiziente Möglichkeit vorgestellt werden, um die *unslotted* Version des CSMA/CA-Algorithmus, unter Berücksichtigung von Signalstärken und Rauschverhältnissen, zu simulieren. Diese Version des Algorithmus soll innerhalb dieses Abschnitts beschrieben werden. Eine graphische Darstellung des Algorithmus findet sich in Abbildung 2.3.

2.4.1 Funktionsweise des CSMA/CA-Algorithmus

Um den CSMA/CA-Algorithmus ausführen zu können, muss jeder Sender bei jedem Senderversuch zwei Variablen pflegen.

- *NB* - bezeichnet die Anzahl an Backoffs (*Number of Backoffs*), die ein Sender bisher durchlaufen hat. Der Backoff ist die zufälligen Zeitspanne, die ein Sender gewartet hat, bevor er den nächsten Schritt des Algorithmus ausführt. *NB* wird mit 0 initialisiert.
- *BE* - ist der *Backoff Exponent*, der Einfluss auf die maximale Länge der Backoffs nimmt. Er hängt davon ab, wie viele Backoffs ein Sender schon warten musste. Er wird mit dem Wert *macMinBE* initialisiert. *macMinBE* wird im IEEE 802.15.4-Standard definiert und Standardmäßig mit 3 initialisiert.

Im Folgenden werden die Schritte erklärt die während des CSMA/CA-Algorithmus durchlaufen werden:

1. **Initiales Backoff:** Der erste Schritt im CSMA/CA-Algorithmus besteht darin, ein zufälliges Zeitintervall mit maximaler Länge von $(2^{BE} - 1)backoffPeriod$

⁵ [4, S. 170]Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

2 Grundlagen

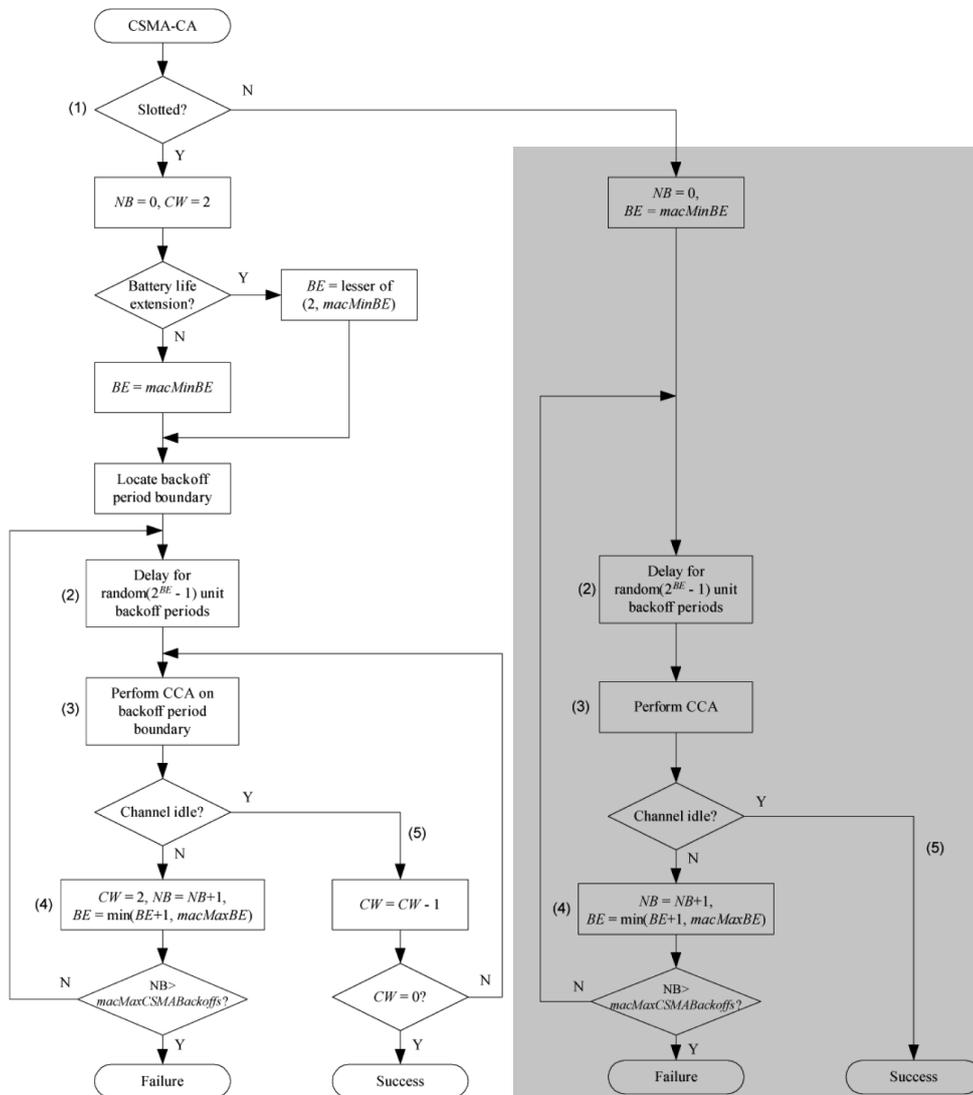


Abbildung 2.3: Ablaufdiagramm des CSMA/CA-Algorithmus. Der grau hinterlegte Bereich beschreibt die für diese Arbeit relevante "unslotted" Version des Algorithmus. NB bezeichnet die Anzahl der bisherigen Backoffs, BE bezeichnet den Backoff Exponent, der benutzt wird um die zufälligen Backoff Intervalle zu berechnen. BE wird während des Algorithmus Schrittweise von $macMinBE$ auf $macMaxBE$ erhöht. $macMaxCSMABackoffs$ bezeichnet die maximale Anzahl, der Versuche auf das Medium zuzugreifen, bevor der Algorithmus aufgibt und den Zugriff als gescheitert erklärt. Abbildung aus IEEE 802.15.4-Standard [4]

auszuwürfeln. *backoffPeriod* bezeichnet hier einen vom Standard vorgegebenen Wert. Für die in dieser Arbeit betrachteten Techniken beträgt dieser Wert ca. $0.32ms$. Bevor der Versuch unternommen wird zu senden, werden durch dieses initiale Warten, Kollisionen durch gleichzeitigen Sendebeginn vermieden. Dies ist wichtig, insbesondere in Fällen bei denen viele Knoten gleichzeitig beabsichtigen zu senden. Ist die Zeitspanne des initialen Backoffs vergangen, so führt der Knoten ein CCA durch.

2. **CCA - Clear Channel Assessment:** Das Clear Channel Assessment, also der Vorgang des Lauschens, ob der Kanal frei zum Senden ist, ist ein essentieller Teil des CSMA/CA-Algorithmus. Physikalisch wird hierbei die Stärke des Signals gemessen, das in diesem Moment auf dem Medium empfangen wird. Dies entspricht der Summe aller Signale, welche augenblicklich bei dem Knoten der senden möchte, empfangen werden. Das Rauschen wird hierbei durch die umliegenden Knoten die gerade senden, verursacht. Deren Signalstärke unterliegt einem exponentiellen Abfall, abhängig von der Entfernung zum Empfänger. Die Amplituden sämtlicher in der Umgebung gesendeten Signale addieren sich bei dem lauschenden Knoten zum Gesamtrauschen auf. Überschreitet das Rauschen einen bestimmten Grenzwert, so wird der Kanal als belegt gewertet. Laut IEEE 802.15.4-Standard ist ein Kanal als frei zu werten, wenn die empfangene Signalstärke höchstens $10dB$ über der spezifizierten Empfängerempfindlichkeit liegt. Im vorliegenden Fall beträgt diese $-85dBm$, der Kanal ist also bis zu einer Signalstärke von $-75dBm = 3.16E - 8mW$ als frei zu werten. Wird beim CCA festgestellt, dass der Kanal nicht belegt ist, so kann der Knoten mit dem Senden beginnen. Andernfalls wird im 3. Schritt ein erneutes Zeitintervall ermittelt, dass der Knoten warten muss, bevor er erneut ein CCA durchzuführen kann.
3. **Backoff im Falle eines belegten Mediums:** Stellt sich beim CCA heraus, dass der Kanal als belegt zu werten ist, so wird NB um eins erhöht, gleichermaßen wird BE angepasst nach der Vorschrift $BE = \min(BE + 1, macMinBE)$. Ist $NB \leq macMaxCSMABackoffs$, so wird wie schon am Anfang, erneut eine zufällige Zeitspanne ermittelt, die maximal $(2^{BE} - 1)backoffPeriod$ beträgt. Der Sender wartet nun diese Zeitspanne, bevor er ein erneutes CCA durchführt, um zu prüfen ob der Zeitpunkt zum Senden günstig ist. *macMaxCSMABackoffs* beträgt laut IEEE 802.15.4 standardmäßig 4, d.h. ein Knoten kann bis zu 5 Sendeversuche unternehmen, bevor er den Zugriff als gescheitert erklären muss. Tritt dieser Fall ein, so muss der Sender entscheiden, wie er mit dem Zugriffsfehler umgeht, die Behandlung dieses Falls liegt nicht mehr in der Verantwortung des CSMA/CA-Algorithmus.
4. **Inter Frame Spacings nach Abschluss einer Übertragung:** Das beschriebene Vorgehen in diesem Schritt gehört eigentlich nicht mehr zum CSMA/CA-Algorithmus, spielt aber im weiteren Verlauf der Arbeit eine Rolle bei der Koordination der Sendevorgänge. es ist deshalb eng verknüpft mit dem Ablauf des CSMA/CA-Algorithmus. Um einem Empfänger Zeit zu geben das gerade empfangene Signal zu verarbeiten, muss ein Sender mindestens eine bestimmte Zeit, ein sog. Inter Frame Spacing (IFS), lang warten, bevor er die nächste Nachricht versenden kann. Die Länge des IFS richtet sich dabei nach der Größe der Nachricht. Bis zu

2 Grundlagen

einer bestimmten Größe muss lediglich ein Short Inter Frame Spacing (SIFS) (ca. $0.19\mu s$) gewartet werden. Ist die Nachricht größer, so muss mindestens ein Long Inter Frame Spacing (LIFS) (ca. $0.64\mu s$) gewartet werden.

Werden die Schritte eins bis drei vor jedem Sendevorgang durchgeführt, so kann ein Großteil der Kollisionen, die ansonsten durch gleichzeitiges Senden entstehen würden, vermieden werden.

2.5 Simulation drahtloser Ad-hoc Netzwerke

Drahtlose Ad-hoc Netzwerke haben in den letzten Jahren den Sprung von einem rein theoretischen Forschungsgebiet hin zu Anwendungen in der Praxis geschafft. Dies ist auch der Entwicklung spezialisierter Simulationsumgebungen zu verdanken, die es ermöglichen Routingalgorithmen und neue verteilte Anwendungen zunächst zu simulieren, anstatt aufwendige Praxistests durchzuführen. Vor allem bei sehr hohen Knotenzahlen werden praktische Tests sehr schnell extrem aufwändig und teuer, so dass in vielen Fällen auf Simulationen zurückgegriffen werden sollte, bevor an eine praktische Erprobung gedacht wird. Darüber hinaus lassen sich manche Effekte innerhalb drahtloser Netzwerke gar nicht oder nur sehr schwer in der Praxis beobachten und messen. Ein Beispiel dafür sind z.B. die physikalischen Aspekte der Nachrichtenübertragung, wie das Signal zu Rauschverhältnis eines Signals während einer Übertragung.

Im Bereich der Sensornetze haben sich maßgeblich zwei Simulatoren etabliert, *Ns-2*⁶ sowie *Shawn*⁷. *Ns-2* legt großen Wert auf die umfassende Simulation aller Parameter der Nachrichtenübertragung, während die Zielsetzung von *Shawn* eher eine algorithmische Simulation des Netzwerks ist. Darunter ist zu verstehen, dass die Simulation der unteren Netzwerkschichten durch stark vereinfachte Modelle übernommen wird, die ein ausreichend exaktes Bild der Wirklichkeit bilden, um z.B. die Funktionsweise von Routingalgorithmen zu überprüfen.

Gegenstand dieser Arbeit ist eine Erweiterung von *SHAWN*, um eine effiziente Art, die physikalischen Parameter der Nachrichtenübertragung wie Signalstärken und Signal-zu-Rausch-Verhältnisse zu simulieren. Aus diesem Grund soll hier nicht weiter auf *Ns-2* eingegangen werden. Wer sich dafür interessiert sei auf das *Ns-2*-wiki⁹ verwiesen.

2.5.1 SHAWN

Wie schon erwähnt ist *SHAWN* eher auf die algorithmische Sicht des Netzwerkes ausgerichtet. Hierfür werden z.B. probabilistische Modelle herangezogen, die aufgrund von bestimmten Wahrscheinlichkeiten entscheiden, ob die Übertragung einer Nachricht während der Simulation erfolgreich beendet werden konnte oder nicht. Da die Simulation der unteren Netzwerkschichten sehr aufwändig ist, kann durch diese Vereinfachungen viel Zeit gespart werden, was Simulationen mit sehr hohen Knotenzah-

⁶ [5]Ns-2: Network simulator-2

⁷ [6]Shawn: A new approach to simulating wireless sensor networks

⁸ [7]Shawn: The fast, highly customizable sensor network simulator

⁹ [5]http://nslam.isi.edu/nslam/index.php/Main_Page

len ermöglicht. Dabei ist jedoch zu berücksichtigen, dass derartige Vereinfachungen oft stark auf Kosten der Übertragbarkeit der Ergebnisse in die Praxis gehen.

2.5.2 Architektur von SHAWN

Eine detaillierte Einführung in die Architektur von SHAWN würde den Rahmen dieser Arbeit sprengen, deswegen soll hier nur ein kurzer Überblick gegeben werden. Für eine umfassendere Beschreibung sei auf die Einführungen von Dr. Alexander Kroeller¹⁰¹¹ sowie auf die Diplomarbeit von Thorsten Teubler¹² verwiesen.

SHAWN ist in C++ implementiert und wird als open-source-Software bereitgestellt. Das Ziel der Implementierung ist es, ein modulares Framework zur Verfügung zu stellen, das von Benutzern leicht an ihre Bedürfnisse angepasst werden kann. Generell besteht SHAWN aus drei Schichten:

1. Applikationsschicht
2. Verwaltungsschicht
3. Kommunikationsschicht

2.5.2.1 Applikationsschicht

Auf der Applikationsschicht werden konkrete Simulationsszenarien entworfen und gesteuert. Hier werden z.B. Routingprotokolle, Simulationsabläufe und das Verhalten von Knoten beim Senden oder Empfangen einer Nachricht implementiert. Die Applikationsschicht spielt in dieser Arbeit eine untergeordnete Rolle und wird lediglich für die Evaluation der Erweiterungen in der Kommunikationsschicht relevant. Dabei werden jedoch im Rahmen dieser Arbeit ausschließlich einfache Standardszenarien in SHAWN getestet. Aus diesem Grund soll an dieser Stelle auf eine detaillierte Erläuterung dieser Schicht verzichtet werden.

2.5.2.2 Verwaltungsschicht

Die Verwaltungsschicht ist das Herzstück von SHAWN. Sie verwaltet die Simulationszeit, die in Shawn in Runden angegeben wird, eine Runde wird dabei in der Regel als eine Sekunde interpretiert. SHAWN ist ereignisgesteuert, was bedeutet, dass während einer Simulation verschiedene Ereignisse erstellt werden, wie z.B. "Sende Nachricht zum Zeitpunkt x". Diese Ereignisse werden dann von der Verwaltungsschicht ausgelöst und deren Behandlung veranlasst, sobald die Simulationszeit den für das Ereignis geplanten Zeitpunkt erreicht hat. Es ist möglich mehrere Ereignisse zum gleichen Zeitpunkt zu planen. In einem solchen Fall hält die Verwaltungsschicht die Simulationszeit zum geplanten Zeitpunkt an, bis alle Ereignisse abgearbeitet sind. Auf diese Art kann

¹⁰ [6]Shawn: A new approach to simulating wireless sensor networks

¹¹ [7]Shawn: The fast, highly customizable sensor network simulator

¹² [8, 10]Optimierung, Implementierung und Evaluation von Routing-Protokollen unter Verwendung globalen Wissens

2 Grundlagen

auch ohne parallele Programmierung Parallelität simuliert werden. Dies ist z.B. wichtig beim Sendebeginn eines Knotens, da alle umliegenden Knoten die ausgesendete Nachricht in der Realität gleichzeitig empfangen, sie auf einem Computer allerdings nur nacheinander abgearbeitet werden können.

Die Verwaltungsschicht stellt auf diese Art das Bindeglied zwischen der Kommunikationsschicht und der Applikationsschicht dar. Sie stellt außerdem Funktionen zum Hinzufügen und Entfernen von Knoten bereit, sowie weitere Funktionen zur Verwaltung der Netzwerktopologie.

2.5.2.3 Kommunikationsschicht

Die Kommunikationsschicht regelt die Übertragung einer Nachricht von einem Knoten an einen Anderen. Grob gesprochen werden innerhalb der Kommunikationsschicht die 1. und die 2. Schicht des ISO/OSI-Modells simuliert, d.h., die Bitübertragungs- und die Sicherungsschicht.

Die Kommunikationsschicht ist in drei Module eingeteilt:

1. Das Kantenmodell
2. Das Kommunikationsmodell
3. Das Transmissionsmodell

Das *Kantenmodell* verwaltet einen Graphen, der alle Knoten enthält. Es beantwortet die Frage welche Knoten einander nah genug sind um miteinander zu kommunizieren. Insbesondere sollte es auf effiziente Weise die Frage nach der gesamten Nachbarschaft eines Knotens beantworten können. Mit Nachbarschaft sind z.B. alle Knoten gemeint, die innerhalb der Kommunikationsreichweite eines Knotens liegen. Bei statischen Netzwerken, also Netzwerke deren Knoten nicht mobil sind, kann dies gelöst werden, indem am Anfang der Simulation sämtliche Nachbarschaften berechnet und im Speicher vorgehalten werden. Sind die Knoten jedoch mobil, so muss die Nachbarschaft evtl. bei jeder Anfrage neu berechnet werden. Bei den Simulationen in dieser Arbeit wurde stets von einem statischen Netzwerk ausgegangen

Das *Kommunikationsmodell* beantwortet die Frage, ob zwei Knoten, die laut Kantenmodell nah genug beieinander liegen um aus geographischer Sicht kommunizieren zu können, dies auch wirklich können. Das Kommunikationsmodell bildet auf diese Weise auf dem durch das Kantenmodell definierten Graphen einen "overlay" Graphen, der es ermöglicht bestimmte Kanten auszuschließen. Das simpelste Kommunikationsmodell ist das sog. "Disk-graph communication model", welches einen Disk-graph auf dem Graphen des Kantenmodells aufbaut. Auf diese Weise wird eine perfekt kreisförmige Ausbreitung der Funksignale der Knoten simuliert. Stellt man beim Kantenmodell und dem Kommunikationsmodell unterschiedliche Kommunikationsreichweiten ein, so lässt sich z.B. simulieren, dass Knoten nicht mit voller Leistung senden. Des weiteren gibt es Kommunikationsmodelle, die es erlauben jeder Kannte des Graphen des Kantenmodells eine Wahrscheinlichkeit zuzuordnen, mit der zwei Knoten miteinander kommunizieren können. Auf diese Weise lassen sich ungleichmäßige Ausbreitungen von Funksignalen simulieren.

2.5 Simulation drahtloser Ad-hoc Netzwerke

Das *Transmissionsmodell* ist das Modul der Kommunikationsschicht, das mit der Verwaltungsschicht spricht. Es entscheidet, wann ein Knoten, der senden will, wirklich mit dem Senden beginnt und welche Knoten die Nachricht erfolgreich empfangen. Dies geschieht zum einen unter Zuhilfenahme des Kommunikationmodells und des Kantenmodells, sowie evtl. durch die Anwendung bestimmter Algorithmen zur Vermeidung von Kollisionen, wie z.B. dem CSMA/CA-Algorithmus. Transmissionsmodelle können innerhalb von Shawn so programmiert werden, dass sie "verkettbar" sind, d.h., dass mehrere Transmissionsmodelle hintereinander geschaltet werden können. Auf diese Weise kann sich z.B. ein Transmissionsmodell darum kümmern, wann ein geeigneter Zeitpunkt zum Senden der Nachricht ist, während sich das Nächste in der Kette darum kümmert ob die Nachricht ankommt. Eine mögliche Kombination wäre hier z.B. das *Aloha*-Modell, welches ohne Rücksicht auf stattfindende Übertragungen immer sofort mit dem Senden beginnt, kombiniert mit dem *reliable*-Modell, welches Nachrichten immer ausliefert. Es wird also die Übertragung garantiert. Diese Kombination würde allerdings ein sehr stark vereinfachtes Bild der Realität zeichnen. Andere Transmissionsmodelle führen eine zufällige Komponente in die Simulation ein, aufgrund derer entschieden wird ob eine Nachrichtenübertragung erfolgreich ist oder nicht. Es werden also Nachrichten zufällig verworfen, so kann auf sehr einfache Weise eine unzuverlässige Übertragung simuliert werden, bei der es zu unvorhersehbaren Verlusten von Nachrichten kommt.

Besonders hervorgehoben werden soll an dieser Stelle das sog. CSMA/CA-Transmissionsmodell. Das CSMA/CA-Transmissionsmodell simuliert den in Abschnitt 2.4 bereits erwähnten 'Carrier Sense Multiple Access/ Collision Avoidance' (CSMA/CA) Algorithmus, der verwendet wird um Kollisionen von Nachrichten bei der drahtlosen Übertragung zu vermeiden. Im Rahmen dieser Arbeit soll eine Erweiterung des CSMA/CA-Transmissionsmodells implementiert werden, die Signalstärkeberechnungen und Signal-zu-Rausch-Verhältnisse mit in die Simulation einbezieht und so eine realistischere Simulation des CSMA/CA-Algorithmus ermöglicht, als dies bisher der Fall war.

2 Grundlagen

3 Simulationsansätze

Im folgenden Kapitel wird zunächst eine Einführung in die generelle Herangehensweise bei der Simulation der Nachrichtenübertragung zwischen zwei Knoten unter Zuhilfenahme des CSMA/CA-Algorithmus gegeben. Im weiteren Verlauf werden die Vorgehensweise des bisherigen CSMA/CA-Transmissionsmodells von SHAWN vorgestellt, sowie die Verbesserungen, die im Rahmen dieser Arbeit implementiert wurden.

3.1 Übertragungsablauf während der Simulation

Der CSMA/CA-Algorithmus wird in der Realität von jedem Sender, vor dem Senden unabhängig ausgeführt. Aus diesem Grund läuft der Algorithmus auf vielen Sendern Parallel. Um dies in SHAWN zu modellieren, wird die Simulation der Nachrichtenübertragung in vier Schritte unterteilt. Beabsichtigt ein Knoten eine Nachricht zu senden, so erzeugt er ein Event zum Zeitpunkt t . Zu diesem Zeitpunkt signalisiert die Verwaltungsschicht von SHAWN der Kommunikationsschicht, dass der Knoten mit dem Senden beginnen möchte. Die Kommunikationsschicht simuliert dann die Übertragung auf Basis des Zustandes der Simulationsumgebung zum Zeitpunkt t , indem sie die vier, in den nächsten Abschnitten vorgestellten, Schritte durchläuft. Beim Durchlaufen wird in der Regel der Zustand der Simulationsumgebung verändert, so dass jeder Sender beim Beginn des Sendevorgangs einen unterschiedlichen Zustand der Simulationsumgebung, abhängig von seiner Position und dem Sendezeitpunkt, vorfindet. Um den jeweiligen Zustand auszuwerten kann an vielen Stellen globales Wissen über die Simulationsumgebung zur Hilfe genommen werden.

3.1.1 CCA, Schritt 1

Das CCA ist die erste Aktion, die von einem Sender im Prozess der Nachrichtenübertragung ausgeführt wird. Je nach Ergebnis des CCA kann der Sender mit dem Senden beginnen oder wartet eine zufällige Zeitspanne um es erneut zu versuchen. Dies wiederholt sich bis zu einer gegebenen maximalen Anzahl an Versuchen, bevor der Sendeversuch als gescheitert erklärt wird.

Zum Zeitpunkt des CCA hat der durchführende Sender noch keinerlei Informationen über den Zustand der Simulationswelt. Der physikalische Vorgang in der Realität sieht an dieser Stelle vor, dass der Sender die momentane Energie misst, die an seiner Antenne anliegt und feststellt ob sie höchstens $10dB$ über der Empfängerempfindlichkeit p_{min} liegt. Aus diesem Grund müssten bei naiver Herangehensweise die empfangenen Signale aller Sender in der Simulation summiert werden.

3.1.2 Kollisionsbehandlung, Schritt 2

Hat der neue Sender vom CCA grünes Licht zum Starten des Sendevorgangs erhalten, so beginnt er mit dem Senden und benachrichtigt andere Knoten darüber. Diese Knoten überprüfen dann, ob die hinzugekommene Nachricht keine Kollision mit der für sie aktuell dominanten Nachricht erzeugt. Mit der dominanten Nachricht ist hier das Signal gemeint, das sie aktuell demodulieren.

Physikalisch korrekt müssten in diesem Schritt alle Knoten der Simulation benachrichtigt werden, da sich ein Signal prinzipiell unendlich weit ausbreitet. Dies hätte allerdings eine sehr lange Laufzeit zur Folge, da die Menge der Empfänger im Laufe der Simulation je nach Knotendichte sehr groß werden kann. Es gilt hierfür bessere Vorgehensweisen zu finden.

3.1.3 Ermitteln neuer Empfänger, Schritt 3

Im 3. Schritt werden alle Knoten, die innerhalb des Kommunikationsradius des neuen Senders liegen und noch keine Nachricht empfangen, benachrichtigt. Die neue Nachricht wird bei ihnen dann als dominante Nachricht eingetragen, wenn sie in der Lage sind sie zu verstehen.

3.1.4 Beenden des Sendevorgangs, Schritt 4

Nach Ablauf der Zeit, die für das Übertragen der Nachricht benötigt wird, werden sämtliche Empfänger, die in Schritt 2 und 3 benachrichtigt wurden, wiederum darüber benachrichtigt, dass der Sendevorgang abgeschlossen wurde. Bei allen Empfängern, die innerhalb des Kommunikationsradius des Senders liegen, wird überprüft, ob die Nachricht des Senders immer noch die dominante Nachricht ist. Ist dies der Fall, so war die Übertragung der Nachricht erfolgreich. Andernfalls wurde die Nachricht während der Übertragung durch den Sendevorgang eines anderen Senders zerstört.

Im Anschluss wird das IFS berechnet, das sowohl der Sender als auch alle Empfänger innerhalb des Kommunikationsradius warten müssen, bevor sie wieder mit dem Senden beginnen.

3.2 Das bisherige CSMA/CA-Modell in SHAWN

Der SHAWN Simulator enthält bereits ein Transmissionmodell, das die Nachrichtenübertragung gemäß dem IEEE 802.15.4¹ Standard simuliert. Im weiteren Verlauf dieser Arbeit wird dieses Modell als das bisherige CSMA/CA-Modell bezeichnet. Das bisherige CSMA/CA-Modell verzichtet darauf, die physikalischen Eigenschaften der Signalübertragung mit in die Simulation einzubeziehen. Sämtliche Entscheidungen innerhalb des CCAs und der Kollisionsermittlung werden auf Basis der Kommunikati-

¹ [4] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

onsreichweite getroffen, die einen Grenzwert darstellt, der über die gesamte Simulation zu jedem Zeitpunkt gleich ist.

3.2.1 Übertragungsablauf im bisherigen CSMA/CA-Modell

Der in Abschnitt 3.1 definierte Ablauf bei der Simulation der Signalübertragung wird vom bisherigen CSMA/CA-Modell auf die im Folgenden beschriebene Weise implementiert.

3.2.1.1 CCA im bisherigen CSMA/CA-Modell

sobald ein Knoten mit dem Senden beginnt, werden im bisherigen CSMA/CA-Modell alle Knoten innerhalb seines Kommunikationsradius benachrichtigt und als Empfänger der Nachricht registriert. Diese Knoten setzen ihren Status infolgedessen für die Dauer der Übertragung zuzüglich eines angemessenen IFS auf "busy".

Der CCA-Algorithmus des bisherigen CSMA/CA-Modells besteht nun lediglich darin sicherzustellen, dass kein Knoten, der aktuell "busy" ist, mit dem Senden beginnt. Auf diese Weise stellt das bisherige CSMA/CA-Modell sicher, dass innerhalb eines beliebigen kreisförmigen Gebiets mit einem Durchmesser gleich dem Kommunikationsradius höchstens ein Sender zu finden ist. Diese Simulation des CCA berücksichtigt jedoch nicht, dass sich das Rauschen, das von mehreren Sendern erzeugt wird, so stark summieren kann, dass das Medium laut IEEE 802.15.4² auch für einen Knoten, der sich nicht innerhalb des Kommunikationsradius eines anderen Knotens befindet, als belegt gelten muss.

3.2.1.2 Kollisionsbehandlung im bisherigen CSMA/CA-Modell und Ermitteln neuer Empfänger

Der dritte Schritt wird beim bisherigen CSMA/CA-Modell gemeinsam mit dem zweiten Schritt durchgeführt. Dies ist möglich, da die Menge an Knoten, für die Kollisionen überprüft werden müssen, identisch ist mit der Menge an Knoten, die als neue Empfänger in Frage kommen. Das Modell ordnet jedem Knoten einen Zustand zu, der alle übertragungsrelevanten Informationen speichert und stellt alle Anfragen bezüglich Knoten in einem bestimmten Bereich an die gesamte Knotenmenge.

Um herauszufinden, welche Knoten innerhalb des Kommunikationsradius liegen, stellt das bisherige CSMA/CA-Modell für jeden Sender eine Anfrage an die Simulationsumgebung, um die adjazenten Knoten zu ermitteln. Das Ergebnis dieser Anfrage wird für jeden Sender über die Dauer des Übertragungsvorgangs gespeichert. Da alle Betrachtungen während der Übertragung auf den Bereich des Kommunikationsradius beschränkt sind, ist das Ergebnis der Anfrage die einzige Knotenmenge, die im Laufe des Sendevorgangs betrachtet werden muss. Auf diese Weise kommt das bisherige

² [4, S. 66] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

3 Simulationsansätze

CSMA/CA-Modell mit nur einer einzigen Adjazenzanfrage an die Simulationswelt pro Sendevorgang aus.

Kollisionen treten im bisherigen CSMA/CA-Modell immer dann auf, wenn ein Empfänger im Kommunikationsradius von mindestens zwei Sendern gleichzeitig liegt. Das heißt, dass ein Sender beim Beginn des Sendevorgangs über alle Knoten innerhalb seines Kommunikationsradius iteriert und alle Nachrichten zerstört, die aktuell auf diesen Knoten empfangen werden. Dies geschieht wiederum unabhängig von der wirklichen Entfernung des neuen Senders zu den jeweiligen Empfängern. Auf diese Weise wird nicht berücksichtigt, dass ein Nachricht eines weit entfernten Senders, evtl. nicht die nötige Signalstärke aufweist um die aktuelle Nachricht auf einem Empfänger zu zerstören, deren Sender sehr nahe ist. Eine solche Situation wird in Abbildung 3.1 illustriert.

Darüber hinaus wird ebenfalls vernachlässigt, dass Übertragungen eines Senders auch über den Kommunikationsradius hinaus Einfluss auf andere Übertragungen haben, selbst wenn die eigentliche Nachricht nicht mehr verstanden werden kann. Knoten auf denen es zu einer Kollision kam, warten mindestens bis zum Abschluss des Sendevorgangs aller Sender innerhalb des Kommunikationsradius zuzüglich eines LIFS bevor sie selbst wieder eine Nachricht senden.

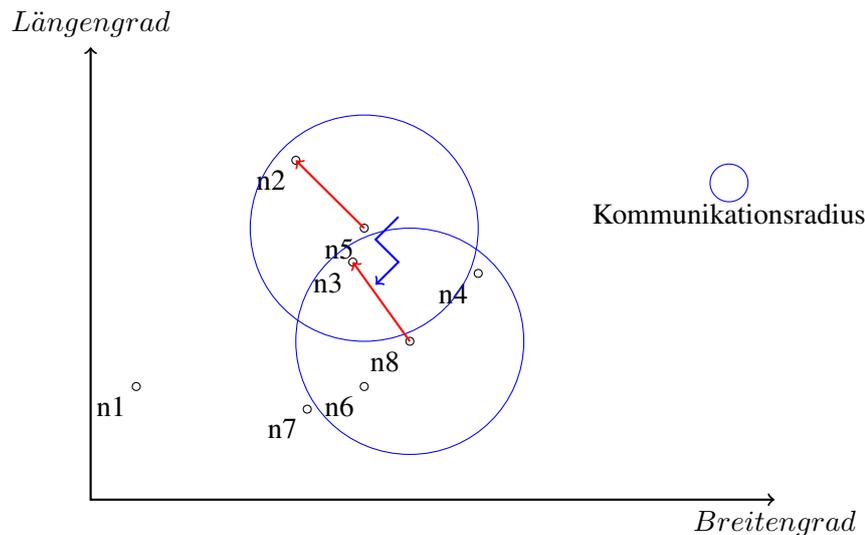


Abbildung 3.1: Kollisionsbehandlung im bisherigen CSMA/CA-Modell. n5 hat vor n8 mit dem Senden begonnen, so dass seine Nachricht auf dem Knoten n3 empfangen wird. n8 kann in dieser Situation ebenfalls mit dem Senden beginnen, da n8 außerhalb des Kommunikationsradius von n5 liegt, also ein "Kanal frei" vom CCA bekommt. Dabei erzeugt n8 eine Kollision auf dem Knoten n3 und zerstört die dort empfangene Nachricht. Dies ist jedoch unrealistisch, da n8 sehr weit von n3 entfernt ist, während n5 sehr nahe an n3 liegt. Aus diesem Grund sollte das Signal von n5 bei n3 immer noch stark genug sein, so dass das Rauschen von n8 die Übertragung nicht stört.

3.3 Konzepte zur Verbesserung des bisherigen CSMA/CA-Modells

Um neue Empfänger zu erstellen, wird beim Iterieren über die Knoten innerhalb des Kommunikationsradius, bei allen Knoten deren aktueller Zustand nicht "busy" ist, bei denen es also nicht zu einer Kollision kam, die neue Nachricht als dominante Nachricht eingetragen. Diese Knoten werden dann ebenfalls als "busy" markiert.

3.2.1.3 Beenden des Sendevorgangs

Beendet der Sender den Sendevorgang, so benachrichtigt er alle Knoten, die er als Ergebnis der Adjazenzanfrage erhalten hat. Es wird nun überprüft, ob bei diesen Knoten nach wie vor die Nachricht des Senders als dominante Nachricht eingetragen ist. Ist dies der Fall so war die Übertragung erfolgreich und die Nachricht wird an den Empfänger übergeben. Andernfalls wird der Empfänger darüber benachrichtigt, dass es zu einer Kollision während der Übertragung der Nachricht gekommen ist. Wie der Empfänger mit einer Kollision umgeht liegt nicht mehr in der Verantwortung des Transmissionsmodells, bzw. des CSMA/CA-Algorithmus.

3.3 Konzepte zur Verbesserung des bisherigen CSMA/CA-Modells

Im Rahmen dieser Arbeit sollen verbesserte Varianten zur Simulation des CSMA/CA Algorithmus entwickelt werden, die insbesondere die physikalischen Aspekte der Nachrichtenübertragung, wie Signalstärken und Signal zu Rausch Verhältnisse, mit in die Simulation einbeziehen. Diese Aspekte werden vom bisherigen CSMA/CA-Modell vollkommen vernachlässigt, wodurch nur ein sehr eingeschränktes Bild der Wirklichkeit simuliert werden kann.

Das bisherige CSMA/CA-Modell kam mit einer einzigen Anfrage auf die gesamte Knotenmenge pro gesendeter Nachricht aus, dies wird mit den Modellen, die die Signal zu Rausch Verhältnisse mit berücksichtigen nicht mehr möglich sein. Aus diesem Grund wird die Knotenmenge einer Simulationswelt gemäß Definition 3.3.1 in mehrere Teilmengen zerlegt, die gemeinsame Eigenschaften aufweisen. Auf diese Art können Anfragen auf kleineren Knotenmengen realisiert werden, was sich positiv auf die Laufzeit auswirkt. Als zugrunde liegende Datenstruktur für diese Teilmengen wurden k-d-Bäume gewählt, da sie effizient Anfragen bezüglich Datenpunkten innerhalb eines Bereichs beantworten können. Auf diesen Aspekt wird im Abschnitt 4.2 eingegangen. Der folgende Abschnitt beschäftigt sich damit, wie auf effiziente Art die Signalstärken und Signal zu Rauschverhältnisse im Laufe der Simulation berechnet werden können.

Definition 3.3.1 (Simulationsumgebung) *Eine Simulation besteht aus einer Menge $K \subset \mathbb{R}^3$ von Knoten $k_i \in K$, $i \in \{1, 2, \dots, n\}$. Darüber hinaus existieren zu jedem Zeitpunkt $t \in \mathbb{N}$, zwei Mengen S_t (Sender), R_t (Receiver) $\subset \mathbb{R}^3$. Es gilt $S_t \cup R_t \subseteq K$ und $S_t \cap R_t = \emptyset$. Die Menge S_t enthält alle zum Zeitpunkt t sendenden Knoten s_j , $j \in \{1, 2, \dots, m\}$ $m \leq n$ und die Menge R_t alle nicht sendenden Knoten r_k , $k \in \{1, 2, \dots, l\}$, die gerade eine Nachricht empfangen könnten, sich also im Kommunikationsreichweite von mindestens einem Knoten $s_j \in S_t$ befinden.*

3 Simulationsansätze

Da in den verbesserten Modellen physikalische Parameter wie z.B. die Signalstärke mit einbezogen werden sollen, ist der Kommunikationsradius im Gegensatz zum bisherigen CSMA/CA-Modell kein fester Wert mehr. Er errechnet sich vielmehr gemäß Definition 3.3.2 aus den Simulationsparametern.

Definition 3.3.2 (Kommunikationsradius $cr(k_i)$) *Der Kommunikationsradius $cr(k_i)$ eines Knotens k_i errechnet sich durch Umstellen der Formel 2.1 nach der Distanz, bei einer gegebenen Empfänger empfindlichkeit p_{min} und einer Sendeleistung $p(k_i)$*

$$cr(k_i) : K \rightarrow \mathbb{R} \text{ mit } cr(k_i) = \|k_i - r_k\| = \sqrt{\frac{p(k_i)}{p_{min}}} \quad (3.1)$$

Der Maximal mögliche Kommunikationsradius entspricht dem eines Senders der mit Stärke p_{max} sendet und wird mit cr_{max} bezeichnet.

$$cr_{max} = \sqrt{\frac{p_{max}}{p_{min}}}$$

Mit den Werten aus Definition 2.3.1 ergibt sich für cr_{max} ein Wert von $cr_{max} = 17.79m$

3.3.1 Übertragungsablauf mit Rauschberechnung

Im Folgenden wird beschrieben, wie die Schritte, die in Abschnitt 3.1 definiert wurden, um Betrachtungen zur Signalstärke und der Signal-zu-Rausch-Verhältnisse erweitert werden.

3.3.1.1 CCA mit Rauschberechnung

Beim Simulieren des CCA stellt sich das Problem, dass der Vorgang des Summieren der Amplituden nicht von selbst erfolgt, sondern simuliert werden muss. Es muss dementsprechend über die Menge der Sender S_t iteriert werden, um mit Hilfe von Gleichung 2.1 zu ermitteln, wie groß die Energie ist, die von den Übertragungen der Sender aus S_t noch bei dem neuen Sender ankommt.

Da die physikalische Ausbreitung eines Signals in der Realität unendlich ist, müsste die Ermittlung des aktuellen Energielevels in der Position des neuen Senders theoretisch über alle Elemente von S_t erfolgen. Dies erscheint jedoch wenig ratsam, da vor allem bei großen Knotenzahlen die Menge S_t viele Elemente enthalten kann, von denen viele sehr weit entfernt von der Position des neuen Senders liegen können. Das Signal dieser weit entfernten Sender ist ab einer gewissen Entfernung so schwach, dass es vernachlässigt werden kann. Hier besteht Spielraum für Laufzeitoptimierungen durch geeignete Abschätzungen. Dafür wird ein Bereich um den neuen Sender ermittelt, innerhalb dessen sendende Knoten noch einen nennenswerten Einfluss auf den Betrag des aktuellen Rauschens haben. Dieser Bereich soll von nun an Rauschradius nr ("noise radius") genannt werden. Auf diese Weise muss nur über Knoten

3.3 Konzepte zur Verbesserung des bisherigen CSMA/CA-Modells

innerhalb des ermittelten Bereichs iteriert werden und nicht über die gesamte Knotenmenge.

Definition 3.3.3 (Rauschradius $nr_t(k_i)$) Der Rauschradius $nr_t(k_i)$ beschreibt die Entfernung, ab der davon ausgegangen wird, dass das Senden von Knoten k_i keinen nennenswerten Einfluss mehr auf andere Knoten hat, bzw den Abstand, den andere Knoten zu k_i haben müssen, um keinen nennenswerten Einfluss mehr auf k_i auszuüben. Es gilt, $nr_t(k_i) \geq cr(k_i) \forall i$.
Es ist während der Simulation an mehreren Stellen notwendig Rauschradien zu schätzen. Diese Schätzungen müssen nicht immer den gleichen Wert ergeben. Im weiteren Verlauf werden die folgenden Radien betrachtet:

$nr_{cca,t}(k_i)$	Bezeichnet den Rauschradius, der der Berechnung des wahrgenommenen Rauschens während des CCAs zugrunde liegt.
$nr_{col,t}(k_i)$	Bezeichnet den Rauschradius, der der Berechnung des wahrgenommenen Rauschens während der Kollisionsbehandlung zugrunde liegt

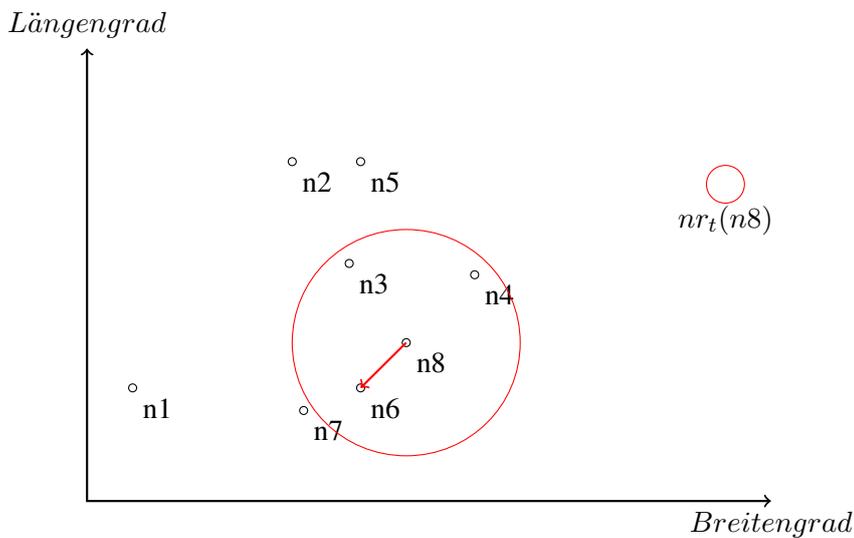


Abbildung 3.2: CCA, Knoten n8 sendet eine Nachricht (u.a. an Knoten n6), dabei erzeugt er Rauschen auf allen Knoten, die innerhalb des Kreises, der durch den Rauschradius nr beschrieben wird, liegen. Außerhalb des Kreises ist das Signal, das von n8 ausgesendet wird aufgrund des exponentiellen Abfalls zu schwach um noch einen nennenswerten Beitrag zum Rauschen zu liefern, das z.B. auf Knoten n7 empfangen wird

Im Rahmen dieser Arbeit wurden Untersuchungen mit einem festen Wert für $nr_t(k_i)$ durchgeführt, sowie ein Ansatz der den Wert für $nr_t(k_i)$ adaptiv anhand der aktuellen Simulationssituation ermittelt.

3.3.1.2 Kollisionsbehandlung

Hat der neue Sender vom CCA grünes Licht zum Starten des Sendevorgangs erhalten, so wird er selbst ein Element der Sendermenge S_t . Er muss nun die Knoten in der Empfängergruppe R_t darüber benachrichtigen, dass er mit dem Senden begonnen hat, damit diese überprüfen können, ob das hinzugekommene Signal keine Kollision mit dem von ihnen empfangenen dominanten Signal erzeugt. Hierbei ist zu beachten, dass die Knoten, die innerhalb des Kommunikationsradius des neuen Senders liegen und noch keine Elemente von R_t sind, erst im nächsten Schritt behandelt werden. Die Menge R_t enthält in diesem Schritt also noch die gleichen Elemente wie vor dem Sendebeginn des neuen Senders.

Aufgrund der in Schritt 1 schon erwähnten unendlichen Signalausbreitung müsste prinzipiell jeder Knoten innerhalb von R_t das SNR der aktuell von ihm empfangenen Nachricht überprüfen. Dies hätte allerdings eine sehr lange Laufzeit zur Folge, da die Menge der Empfänger im Laufe der Simulation je nach Knotendichte sehr groß werden kann. Es gilt hier wiederum einen Bereich abzuschätzen, der alle Empfänger beinhaltet, auf die der neue Sender einen nennenswerten Einfluss hat. Dieser Bereich soll von nun an Benachrichtigungsradius rnr ("receiver notification radius") genannt werden.

Definition 3.3.4 (Benachrichtigungsradius rnr ('receiver notification radius'))
Der Benachrichtigungsradius rnr eines Senders im Punkt x beschreibt den Bereich innerhalb dessen der Sender alle Empfänger darüber informiert, dass er mit dem Senden begonnen hat. Die Empfänger in diesem Bereich überprüfen dann, ob das neue Signal bei dem von ihnen bisher empfangenen Signal zu einer Kollision führt.

Jeder Knoten r_k aus R_t , dessen Abstand zum neuen Sender kleiner ist als rnr , überprüft nun zunächst welches der bei ihm wahrnehmbaren Signale das Stärkste ist. Für dieses Signal des stärksten Senders s_j überprüft er nun, ob $SNR(s_j, r_k) \geq SNR_{min}$ ist. Ist dies der Fall, so wird die von s_j ausgehende Nachricht als dominante Nachricht gesetzt bzw. beibehalten, sofern das Signal stärker ist als die Empfängerempfindlichkeit p_{min} .

Um nicht jedes Mal, wenn ein Sender mit dem Senden beginnt, für alle Empfängern innerhalb des Benachrichtigungsradius die Rauschsituation in ihrer Position neu zu ermitteln, wird das aktuelle Rauschen auf jedem Empfänger kumulativ ermittelt. Das bedeutet, dass ein Empfänger in dem Moment, in dem er in R_t aufgenommen wird, eine Anfrage an S_t stellt, die alle Sender innerhalb seines Rauschradius zurückgibt. Der neue Empfänger ermittelt dann über all diese Sender das momentane Rauschlevel in seiner Position. Wird er nun von einem neuen Sender darüber benachrichtigt, dass ein zusätzliches Signal gesendet wird, so addiert er einfach dieses Signal zu dem bestehenden Rauschen hinzu und erhält so den aktuellen Rauschlevel, ohne eine neue Anfrage an S_t stellen zu müssen. Umgekehrt zieht er den Rauschwert eines Signals wieder vom Wert des aktuellen Rauschens ab, sobald er von einem Sender benachrichtigt wird, dass eine Übertragung abgeschlossen ist. Diese Vorgehensweise stellt sicher, dass ein Empfänger immer einen aktuellen Wert bzgl. des Rauschlevels innerhalb seines Rauschradius hat.

3.3.1.3 Ermitteln neuer Empfänger

Um die Empfänger einer Nachricht zu ermitteln wird eine Anfrage an die Simulationswelt gestellt, hinsichtlich aller Knoten, die sich innerhalb des Kommunikationsradius des neuen Senders befinden. Beim Iterieren über diese Knoten wird dann zunächst überprüft, ob sich der jeweilige Knoten schon in der Menge R_t befindet. Ist dies der Fall so wurde er in Schritt 2 schon über den Sendebeginn informiert und kann in diesem Schritt ignoriert werden.

Ist der betrachtete Knoten noch kein Element von R_t , so wird für den Knoten ein geeigneter Rauschradius ermittelt, auf dessen Grundlage der neue Empfänger das aktuelle Rauschlevel in seiner Position approximiert. Auf Basis dieses Rauschlevels bestimmt der neue Empfänger dann, ob er die neue Nachricht verstehen kann, oder ob das $SNR \leq 4$ ist, es also zu einer Kollision gekommen ist.

Dieser Schritt ist der einzige, in dem Knoten in die Menge R_t aufgenommen werden. Da dabei nur Knoten in Frage kommen, die innerhalb des Kommunikationsradius eines anderen Knotens liegen und noch nicht Element von R_t sind, wird sichergestellt, dass höchstens ein Sender in jedem beliebigen kreisförmigen Bereich der Simulationsfläche mit einem Durchmesser $d \leq cr_{max}$ zu finden ist. Dies entspricht dem Verhalten des bisherigen CSMA/CA-Modells.

Im Gegensatz zum bisherigen CSMA/CA-Modell werden bei den neuen Modellen Schritt 2 und 3 nicht gleichzeitig abgearbeitet. Dies liegt an der Unterteilung der Knotenmenge K in die Teilmengen R_t und S_t . Die Statusinformationen über Sender bzw. Empfänger sind nur bei den Elementen innerhalb der Teilmengen vorhanden. Auf diese Weise soll sichergestellt werden, dass bei Bereichsanfragen nur die Knoten behandelt werden, die auch wirklich von Interesse sind. Dies ist wichtig, da insbesondere an die Menge der Sender sehr viele Anfragen gestellt werden um das Rauschen zu ermitteln. Es wäre sehr teuer, hierbei jedes Mal über die gesamte Knotenmenge zu iterieren.

3.3.1.4 Beenden des Sendevorgangs

Beendet ein Sender s_j den Sendevorgang, so benachrichtigt er alle Knoten, die er schon in Schritt 2 und Schritt 3 benachrichtigt hat. Hierbei handelt es sich sowohl um Knoten, die innerhalb des Kommunikationsradius des Senders liegen, als auch um Knoten die außerhalb des Kommunikationsradius und nur innerhalb des Rauschradius liegen.

Liegt ein Knoten innerhalb des Kommunikationsradius des Senders, so wird überprüft, ob die beim ihm eingetragene dominante Nachricht nach wie vor die von s_j gesendete ist. Ist dies der Fall, so wird die Nachricht ausgeliefert und die Übertragung war erfolgreich. Andernfalls wird der Knoten darüber benachrichtigt, dass die Übertragung nicht erfolgreich war.

Liegt der Knoten darüber hinaus nicht innerhalb des Kommunikationsradius eines weiteren noch aktiven Senders, so wird er aus der Menge der Empfänger R_t entfernt. Um sicherzustellen, ob ein Knoten noch im Kommunikationsradius eines anderen Senders liegt, pflegt ein Empfänger eine Liste aller Sender, die ihn innerhalb der Schritte 2 und 3 benachrichtigen und deren Abstand zu ihm kleiner ist als $cr(s_j)$. Benachrichtigt ei-

3 Simulationsansätze

ner dieser Sender den Empfänger darüber, dass er nicht mehr sendet, so wird er aus der Liste der Sender im Kommunikationsradius entfernt. Ein Empfänger befindet sich genau dann nicht mehr in Kommunikationsreichweite eines Senders, wenn keine Sender mehr in der Liste vorhanden sind. Ist dies nicht der Fall, so wird, wie schon in Schritt 2 erwähnt, das aktuelle Rauschlevel des Empfängers um die Stärke des Signals von s_j verringert und der Empfänger in R_t belassen.

Liegt ein Knoten außerhalb des Kommunikationsradius von s_j , so wird nur das Rauschlevel entsprechend verringert, es findet keine Nachrichtenbehandlung statt.

3.4 Architektur der neuen Transmissionsmodelle

Das generelle Vorgehen der 3 neuen Transmissionsmodelle folgt den im Abschnitt 3.3.1 beschriebenen Schritten. Die Unterschiede bestehen hauptsächlich in der Wahl des Rausch- bzw. des Benachrichtigungsradius. Auf die Strategien, die zur Wahl dieser Radien benutzt werden, wird im Folgenden eingegangen.

3.4.1 NoSNR Ansatz

Der *NoSNR*-Ansatz wurde implementiert um das grundsätzliche Vorgehen des Transmissionsmodells mit dem des bisherigen CSMA/CA-Transmissionsmodells zu vergleichen. Um dies zu erreichen wurde ein Modell implementiert, das im Gegensatz zu den anderen beiden Modellen auf sämtliche Signalstärkeberechnungen verzichtet. Entscheidungen bezüglich Kollisionen und Kanalbelegungen werden nach den gleichen Regeln getroffen, die schon im bisherigen CSMA/CA-Modell von SHAWN benutzt werden. Bei Auswertungen sollten die Ergebnisse des bisherigen CSMA/CA-Modells ähnlich zu denen des NoSNR-Modells sein, wodurch sichergestellt wird, dass die grundsätzliche Vorgehensweise beider Modelle gemäß der Spezifikation des CSMA/CA-Algorithmus im IEEE 802.15.4-Standard ³ gleich ist. Die Implementierung der einzelnen Schritte beim NoSNR-Modell werden in den folgenden Unterabschnitten beschrieben.

3.4.1.1 CCA

Es wird eine Anfrage an S_t gestellt, welche sendenden Knoten sich innerhalb des Kommunikationsradius des neuen Senders befinden. Wird innerhalb dieses Bereichs ein Knoten gefunden, so gilt der Kanal als belegt. Eine Berechnung des Rauschens das durch die Sender in S_t generiert wird, erfolgt nicht.

Im Gegensatz zum bisherigen CSMA/CA-Modell muss bei dieser Herangehensweise jeder neue Sender selbst feststellen ob der Kanal belegt ist. Beim bisherigen Modell liegt die Verantwortung dafür bei den sendenden Knoten, die allen Empfängern mitteilen, dass der Kanal belegt ist. Das Ergebnis sollte jedoch identisch sein, da nur die

³ [4, S. 66] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)

3.4 Architektur der neuen Transmissionsmodelle

Betrachtungsweise umgedreht wurde. Der Rauschradius ist bei diesem Modell immer gleich dem Kommunikationsradius, d.h. $nr(s_j) = cr(s_j) \forall j$

3.4.1.2 Kollisionsbehandlung

bei der Kollisionsbehandlung im NoSNR-Modell wird ermittelt, welche Knoten aus R_t innerhalb des Rauschradius liegen. Der Rauschradius ist in diesem Fall wiederum gleich dem Kommunikationsradius. Dabei werden bei allen diesen Knoten, genauso wie beim bisherigen CSMA/CA-Modell, die dominanten Nachrichten zerstört. Dies geschieht ohne irgendwelche Betrachtungen der Rauschverhältnisse.

3.4.1.3 Ermitteln neuer Empfänger

Das Erstellen funktioniert indem bei allen Knoten innerhalb des Kommunikationsradius, die noch nicht Elemente von R_t sind, die neue Nachricht als dominante Nachricht eingetragen wird und diese Knoten zu R_t hinzugefügt werden. Dies entspricht vom Ergebnis ebenfalls dem Vorgehen des bisherigen CSMA/CA-Modells.

3.4.1.4 Beenden des Sendevorgangs

Das Beenden des Sendevorgangs läuft genau so wie in Abschnitt 3.3.1.4 beschrieben ab. Dabei wird zwar der Rauschlevel auf den Empfängern verringert, dies ist jedoch unerheblich, da der Wert innerhalb dieses Modells nie verwendet wird. Der dadurch entstehende zusätzliche Aufwand wird in Kauf genommen, da dieses Modell lediglich zur Überprüfung der Ähnlichkeit der Modelle mit dem bisherigen CSMA/CA-Modell dient und deshalb nicht auf optimale Laufzeit getrimmt werden muss.

3.4.2 SimpleSNR Ansatz

Beim *SimpleSNR*-Ansatz werden im Gegensatz zum NoSNR-Ansatz Signalstärken und Signal-zu-Rausch-Verhältnisse während des Übertragungsprozesses berechnet und auf deren Basis sämtliche Simulationsentscheidungen getroffen. Das Modell folgt dem in Abschnitt 3.3.1 beschriebenen Simulationsablauf wobei am Anfang der Simulation ein fester Wert $nr_t(k_i) = NR \forall k_i$ gewählt wird, der während der ganzen Simulation konstant bleibt. Des weiteren wird der Benachrichtigungsradius $nr(s_j) = RNR \forall s_j$ ebenfalls fest gewählt und bleibt über die gesamte Simulation gleich. in der Regel gilt $RNR = NR$. Auf diese Art und Weise können Untersuchungen durchgeführt werden, welchen Einfluss die Wahl des Rauschradius auf das Simulationsergebnis hat.

Gleichzeitig ermöglicht dieses Modell den kanonischen Ansatz zur Rauschberechnung zu testen, bei dem die Rauschverhältnisse während der Simulation immer über das gesamte Simulationsgebiet ermittelt werden. Dafür muss bei der Wahl der Radien ein genügend großer Wert gewählt werden, so dass das von diesen Radien beschriebene Gebiet, unabhängig von der Knotenposition, immer das gesamte Simulationsgebiet mit einschliesst. Diese Konfiguration simuliert die Transmission dann mit der höchstmöglichen Genauigkeit, da immer das Rauschen aller Sender mit in die SNR- und

3 Simulationsansätze

CCA-Berechnungen einfließt. Es ist jedoch zu erwarten, dass diese Genauigkeit einen stark negativen Einfluss auf die Laufzeit hat.

3.4.3 Wahl des Rauschradius unter Berücksichtigung der aktuellen Simulationssituation

Eine konstante Wahl der Radien über die gesamte Simulation hinweg ist aus mehreren Gründen nicht der idealste Weg, um Signalverhältnisse und Rauschen während der Simulation zu berechnen. Sind z.B. viele Sender auf dem Simulationsgebiet aktiv, so hat das je nach Dichte der Knoten zur Folge, dass es auch mehr empfangende Knoten gibt. Es könnte aus diesem Grund ratsam sein einen sehr großen Benachrichtigungs bzw. Rauschradius zu wählen, um sicherzustellen, dass man nicht viele Kollisionen übersieht oder mit dem Senden beginnt, obwohl der Kanal eigentlich belegt sein müsste. Dieses würde das Problem zwar lösen, die Lösung hätte jedoch zur Folge, dass selbst für den Fall, dass nur wenige Sender vorhanden sind, eine sehr großflächige Anfrage an R_t , bzw. S_t gestellt wird, obwohl sicherlich auch, aufgrund der Berechnungen auf einem kleineren Gebiet, hinreichend genaue Aussagen über die Rauschverhältnisse getroffen werden könnten. Es liegt also nahe $nr(k_i)$ und $rnr(s_j)$ bei jeder Berechnung adaptiv aufgrund der aktuellen Simulationssituation, insbesondere der aktuellen Anzahl an Sendern, zu bestimmen.

Der Rauschradius sollte so gewählt werden, dass alle Sender, die auf einen Knoten einen nennenswerten Einfluss haben in die Rauschberechnung mit einfließen. Nennenswert bedeutet in diesem Zusammenhang, dass das Hinzukommen des Signals eines Senders das Potential hat eine Kollision zu erzeugen, also das SNR der dominanten Nachricht eines Empfängers unter SNR_{min} zu drücken. Ein Ansatz, wie der Rauschradius abhängig von der Simulationssituation gewählt werden kann, wird in Satz 3.4.1 beschrieben.

Satz 3.4.1 (Geeignete Wahl von $nr_t(k_i)$) Sei S_t die Menge aller zum Zeitpunkt t sendenden Knoten und S'_t eine Teilmenge von S_t die alle Sender enthält, die sich innerhalb einer Fläche mit Radius $nr_t(r_k)$ um einen Empfänger r_k befinden.

$N_{S'_t}$ bezeichne das Rauschen das von r_k , von allen Sendern innerhalb von S'_t , außer von s_{dom} wahrgenommen wird, wobei s_{dom} derjenige Sender ist, der die in r_k dominante Nachricht sendet.

Es gilt:

$$nr_t(r_k) \geq \sqrt{p_{max}(|S_t| - |S'_t|) \left(\frac{p(s_{dom}, r_k)}{10^{\frac{SNR_{min}}{10}}} - N_{S'_t} \right)^{-1}}$$

$$\Rightarrow SNR_{S_t}(s_{dom}, r_k) \geq SNR_{min}$$

des weiteren gilt:

$$SNR_{S'_t}(s_{dom}, r_k) \geq SNR_{S_t}(s_{dom}, r_k).$$

Beweis siehe Seite 30

Interpretiert man die von Satz 3.4.1 vorgeschlagene untere Schranke für den Rauschradius, so stellt man fest, dass eine kleine Differenz zwischen der Signalstärke des dominanten Signals und dem Rauschen, das durch andere Sender innerhalb des Rauschradius erzeugt wird, einen vergrößernden Einfluss auf $nr(k_i)$ hat. Genauso kann der

3.4 Architektur der neuen Transmissionsmodelle

Rauschradius umso kleiner gewählt werden, je mehr Sender schon innerhalb des Radius bekannt sind. Dies entspricht den Erwartungen.

Satz 3.4.1 beschreibt so die Entfernung, die alle Sender außerhalb des Rauschradius mindestens von r_k entfernt sein müssen, damit das gesamte von ihnen ausgehende Rauschen nicht mehr imstande ist, auf dem Empfänger eine Kollision zu erzeugen. Darüber hinaus geht aus dem Satz hervor, dass der SNR Wert, der über einen Bereich mit dem Radius $nr(k_i)$ ermittelt wird, immer größer oder gleich dem tatsächlichen Signal-zu-Rausch-Verhältniss ist. Aus diesem Grund sollte das Transmissionsmodell eine geeignete Strategie zur Verfügung stellen, wie der Fall zu behandeln ist, dass

$$nr_t(r_k) \leq \sqrt{p_{max}(|S_t| - |S'_t|) \left(\frac{p(S_{dom}, r_k)}{10^{\frac{SNR_{min}}{10}}} - N_{S'_t} \right)^{-1}} \text{ ist.}$$

Hier bieten sich zwei Vorgehensweisen an. Die einfachste Strategie wäre es in diesem Fall das Rauschen über alle Sender zu ermitteln und so ein Exaktes SNR zu berechnen, auf dessen Basis die Kollisionsbehandlung durchgeführt werden kann. Diese würde allerdings wiederum eine sehr hohe Laufzeit zur Folge haben.

Ein anderer Ansatz wäre es, den Rauschbereich auf die von Satz 3.4.1 vorgeschlagene Größe zu erweitern und mit den neugewonnen Informationen erneut zu überprüfen, ob der Radius nun groß genug ist, um sicher zu bestimmen, ob es zu einer Kollision kommt. Dieses Erweitern des Radius macht eine erneute Überprüfung des Radius notwendig, da innerhalb des erweiterten Radius weitere Sender liegen könnten, die vorher noch nicht Teil der Rauschberechnung waren. Ist dies der Fall, so haben sich die Parameter $N_{S'_t}$ sowie $|S'_t|$ in der Gleichung verändert und der minimale Rauschradius muss neu berechnet werden. Wählt man diese Strategie muss man allerdings beachten, dass man sicherstellt beim Erweitern des Radius nicht noch einmal über alle schon bekannten Knoten zu iterieren. Ist dies nicht möglich, so sollte die erste Variante bevorzugt werden.

Beweis 3.4.1 (Satz 3.4.1)

$$\begin{aligned}
 \underbrace{nr_t(r_k)}_{nr} &\geq \sqrt{p_{max}(|S_t| - |S'_t|) \left(\frac{p(s_{dom}, r_k)}{10^{\frac{SNR_{min}}{10}}} - N_{S'_t} \right)^{-1}} \\
 &\Leftrightarrow nr^2 \geq p_{max}(|S_t| - |S'_t|) \left(\frac{p(s_{dom}, r_k)}{10^{\frac{SNR_{min}}{10}}} - N_{S'_t} \right)^{-1} \\
 &\Leftrightarrow \frac{nr^2}{p_{max}(|S_t| - |S'_t|)} \geq \left(\frac{p(s_{dom}, r_k)}{10^{\frac{SNR_{min}}{10}}} - N_{S'_t} \right)^{-1} \\
 \Leftrightarrow nr^{-2} p_{max}(|S_t| - |S'_t|) &\leq \frac{p(s_{dom}, r_k)}{10^{\frac{SNR_{min}}{10}}} - N_{S'_t} \\
 &\Leftrightarrow \frac{1}{10^{\frac{SNR_{min}}{10}}} \geq \frac{N_{S'_t} + nr^{-2} p_{max}(|S_t| - |S'_t|)}{p(s_{dom}, r_k)} \\
 &\Leftrightarrow 10^{\frac{SNR_{min}}{10}} \leq \frac{p(s_{dom}, r_k)}{N_{S'_t} + nr^{-2} p_{max}(|S_t| - |S'_t|)} \\
 &\Leftrightarrow SNR_{min} \leq 10 \log_{10} \frac{p(s_{dom}, r_k)}{N_{S'_t} + nr^{-2} p_{max}(|S_t| - |S'_t|)} \\
 &\Leftrightarrow SNR_{min} \leq 10 \log_{10} \frac{p(s_{dom}, r_k)}{\sum_{s_i \in S'_t} p(s_i, r_k) + \sum_{s_i \in S_t \setminus S'_t} nr^{-2} p_{max}} \\
 &\Leftrightarrow SNR_{min} \leq 10 \log_{10} \frac{p(s_{dom}, r_k)}{\sum_{s_i \in S'_t} p(s_i, r_k) + \sum_{s_i \in S_t \setminus S'_t} \underbrace{nr^{-2}}_{\leq \|s_i - r_k\|^{-2}} p_{max}} \\
 &\quad nr \leq \|s_i - r_k\| \text{ gilt, da alle Elemente aus } S_t \text{ ausserhalb des Rauschradius liegen, also mindestens } nr \text{ von } r_k \text{ entfernt liegen} \\
 &\leq 10 \log_{10} \frac{p(s_{dom}, r_k)}{\sum_{s_i \in S'_t} p(s_i, r_k) + \sum_{s_i \in S_t \setminus S'_t} p(s_i, r_k)} \\
 &= 10 \log_{10} \frac{p(s_{dom}, r_k)}{\sum_{s_i \in S_t} p(s_i, r_k)} \\
 &= SNR_{s'_t}(s_{dom}, r_k) \\
 &\leq 10 \log_{10} \frac{p(s_{dom}, r_k)}{\sum_{s_i \in S'_t} p(s_i, r_k)} = SNR_{s'_t}(s_{dom}, r_k)
 \end{aligned} \tag{3.2}$$

3.4.4 ExtendedSNR Ansatz

Das ExtendedSNR-Modell verfolgt den in Abschnitt 3.4.3 vorgestellten Ansatz. Es verwendet die vorgestellte Idee zur Wahl des Rauschradius abhängig von der aktuellen Simulationssituation. Darüber hinaus wird der Rauschradius schrittweise erweitert, festgestellt wird, dass er nicht ausreichend groß ist. Auf diese Weise sollte eine hohe Genauigkeit mit einer deutlich geringeren Laufzeit zu erreichen sein, verglichen mit dem Ansatz, das SimpleSNR-Modell mit maximalen Radien zu verwenden. Im Folgenden wird die Implementierung der Simulationsschritte beim ExtendedSNR-Modell erläutert.

3.4.4.1 CCA mit adaptivem Rauschradius im ExtendedSNR-Modell

Um während des CCA den Bereich zu bestimmen, über den das aktuelle Rauschen auf dem Medium bestimmt wird, soll der minimale Abstand berechnet werden, den alle anderen aktiven Sender innerhalb der Simulation zu dem neuen Sender haben müssten, damit das gesamte von ihnen erzeugte Rauschen nicht mehr als $10dB$ über der Empfängersensitivität p_{min} liegt. Unter diesen Voraussetzungen liefert jeder Sender den gleichen Beitrag zum Gesamtrauschen. Satz 3.4.2 beschreibt wie $nr_{cca,t}$ gewählt werden kann um diese Bedingungen zu erfüllen.

Satz 3.4.2 (Geeignete Wahl von $nr_{cca,t}(k_i)$) Sei S_t die Menge aller zum Zeitpunkt t sendenden Knoten. SNR_{max} beschreibe das maximal zulässige Rauschen relativ zur Empfängersensitivität in dB, bis zu dem der Kanal als frei gewertet werden kann.

$$\text{Es Gilt } \|s_j - r_k\| \geq nr_{cca,t}(r_k) \geq cr_{max} \sqrt{\frac{|S_t|}{10^{\frac{SNR_{max}}{10}}}} \quad \forall s_j \in S_t$$

$$\Rightarrow 10 \log \frac{p_{max}|S_t|nr_{cca,t}^{-2}}{p_{min}} \leq SNR_{max}.$$

Beweis siehe Seite 32

Wählt man $nr_{cca,t} = cr_{max} \sqrt{\frac{|S_t|}{10^{\frac{SNR_{max}}{10}}}}$, so ist sichergestellt, dass, selbst wenn kein einziger Sender innerhalb des betrachteten Bereichs liegt, davon ausgegangen werden kann, dass die übrigen Sender nicht genug Rauschen produzieren um den Kanal zu belegen. Tritt jedoch der Fall ein, dass innerhalb des Rauschradius einige Sender gefunden werden, deren erzeugtes Gesamtrauschen noch nicht ausreicht, um den Kanal als belegt zu werten, so war die Wahl von $nr_{cca,t}$ nicht groß genug. In diesem Fall muss eine der schon in Abschnitt 3.4.3 erwähnten Methoden angewendet werden um sicher prüfen zu können, ob der Kanal als frei gewertet werden kann oder nicht. In der vorliegenden Arbeit wurde hierfür die Methode gewählt, das CCA noch einmal über die gesamte Simulationsumgebung durchzuführen.

3 Simulationsansätze

Beweis 3.4.2 (Satz 3.4.2)

$$\begin{aligned}
 nr_{cca,t} &\geq cr_{max} \sqrt{10^{\frac{|S_t|}{SNR_{max}}}} \\
 \Leftrightarrow nr_{cca,t}^2 &\geq \frac{p_{max}|S_t|}{p_{min}10^{\frac{SNR_{max}}{10}}} \\
 \Leftrightarrow nr_{cca,t}^{-2} &\leq \frac{p_{min}10^{\frac{SNR_{max}}{10}}}{p_{max}|S_t|} \tag{3.3} \\
 \Leftrightarrow \frac{p_{max}|S_t|nr_{cca,t}^{-2}}{p_{min}} &\leq 10^{\frac{SNR_{max}}{10}} \\
 \Leftrightarrow 10 \log \frac{p_{max}|S_t|nr_{cca,t}^{-2}}{p_{min}} &\leq SNR_{max} \\
 &\Rightarrow \text{Behauptung}
 \end{aligned}$$

Ist der Kanal frei, muss für den nächsten Schritt der Benachrichtigungsradius bestimmt werden. Der Benachrichtigungsradius sollte so gewählt werden, dass alle Empfänger auf die der neue Sender nennenswerten Einfluss hat, über den Sendebeginn informiert werden. Nennenswert bedeutet in diesem Zusammenhang, dass das Hinzukommen des Signals des neuen Senders das Potential hat eine Kollision zu erzeugen, also das SNR der dominanten Nachricht eines Empfängers unter 4 zu drücken.

Satz 3.4.3 schlägt eine Möglichkeit vor wie ein geeigneter Benachrichtigungsradius ermittelt werden kann.

Satz 3.4.3 (Geeignete Wahl von rnr) Sei $s_j \in S_t$ ein Sender und $|S_t|$ die Anzahl aller Sender. Sei r_k ein Empfänger welcher $rnr(S_t)$ Meter von s_j entfernt liegt und dessen Signal der dominanten Nachricht ausgehend von einem Sender s_k eine Stärke von p_{min} aufweist.

Es wird angenommen, dass alle Sender im Mittel gleich weit von r_k entfernt liegen, sie liefern also alle den gleichen Beitrag zum Gesamttrauschen: $N_{S_t} = (|S_t| - 1)p_{max}rnr(S_t)^{-2}$ auf r_k . rnr beschreibt also die Entfernung die alle Sender außer s_k theoretisch von r_k entfernt sein müssen, damit das von ihnen erzeugte Rauschen keine Kollision erzeugt.

Befinden sich alle Sender in einem Abstand von

$$rnr(S_t) \geq cr_{max} \sqrt{10^{\frac{SNR_{min}}{10}} (|S_t| - 1)} \text{ zu } r_k, \text{ dann folgt daraus, dass} \\
 SNR_{dom} = 10 \log_{10} \frac{p_{min}}{N_{S_t}} \geq SNR_{min}.$$

Beweis Seite 33.

Der Benachrichtigungsradius ist einer der Hauptfaktoren bei der Laufzeit der Simulation, da jeder Empfänger, der benachrichtigt wird, eine Kollisionsüberprüfung durchführt. Aus diesem Grund sollte er möglichst eng gewählt werden. Die in Satz 3.4.3 vorgestellte Methode versucht eine geeignete automatische Wahl des Benachrichtigungsradius durchzuführen.

Beweis 3.4.3 (Satz 3.4.3)

$$\begin{aligned}
 rnr(S_t) &\geq cr_{max} \sqrt{10^{\frac{SNR_{min}}{10}} (|S_t| - 1)} \\
 \Leftrightarrow rnr(S_t)^2 &\geq \frac{p_{max} 10^{\frac{SNR_{min}}{10}} (|S_t| - 1)}{p_{min}} \\
 \Leftrightarrow rnr(S_t)^{-2} &\leq \frac{p_{min}}{p_{max} 10^{\frac{SNR_{min}}{10}} (|S_t| - 1)} \\
 \Leftrightarrow 10^{\frac{SNR_{min}}{10}} &\leq \frac{p_{min}}{(|S_t| - 1) p_{max} rnr(S_t)^{-2}} \tag{3.4} \\
 \Leftrightarrow SNR_{min} &\leq 10 \log_{10} \frac{p_{min}}{(|S_t| - 1) p_{max} rnr(S_t)^{-2}} \\
 \Leftrightarrow SNR_{min} &\leq 10 \log_{10} \frac{p_{min}}{N_{S_t}} \\
 &\Rightarrow \text{Behauptung}
 \end{aligned}$$

3.4.4.2 Wahl des Rauschradius bei der Kollisionsbehandlung

Da während des CCAs jeder Sender einen individuellen Wert für den Benachrichtigungsradius berechnet, kann der Rauschradius $nr_{col,t}$ eines Empfängers r_k , also der Bereich innerhalb dessen dem Empfänger Sender bekannt sind, bei jedem CCA vergrößert werden. Dies geschieht jedes Mal, wenn ein Sender, der weiter entfernt ist als $nr_{col,t}$, es für notwendig hält, den Empfänger über seinen Sendebeginn zu informieren. Der neue Rauschradius ist dann gleich der Entfernung zwischen dem Empfänger und dem neuen Sender. Auf diese Weise kann der Bereich, in dem einem Empfänger Sender bekannt sind, während der Zeit die er Element in R_t ist, nur größer werden.

Um zu entscheiden, ob ein neues Signal zu einer Kollision führt, wird das Signal-zu-Rausch-Verhältniss des stärksten Signals, dessen Signalstärke mindestens der Empfängersensitivität entspricht, berechnet. Ist das $SNR(s_{dom}, r_k) \geq SNR_{min}$ so wird anhand der in Satz 3.4.1 angegebenen Ungleichung überprüft, ob der der Berechnung zugrunde liegende Rauschradius ausreichend ist, um sicher zu entscheiden, dass die Nachricht verstanden werden kann. Wird festgestellt, dass dem nicht so ist, wird der Rauschradius schrittweise erweitert. Dies ist in Abschnitt 4.3 beschrieben.

3.4.4.3 Ermitteln neuer Empfänger

Für alle neu ermittelten Empfänger r_k innerhalb des Kommunikationsradius des neuen Senders, muss zunächst ein initialer Rauschradius $nr_{col,t}(r_k)$ gewählt werden, auf dessen Basis das Rauschen in der Umgebung von r_k approximiert wird. Hierzu wird wiederum der in Satz 3.4.3 verwendete Benachrichtigungsradius verwendet. Auf diese Weise soll sichergestellt werden, dass der neue Empfänger alle Sender findet, die ihn bei ihrem Sendebeginn benachrichtigt hätten, wenn er zu diesem Zeitpunkt schon ein Empfänger gewesen wäre. Es gilt also $nr_{col,t}(r_k) = rnr(S_t)$. Jeder Empfänger ermittelt nun über einen Bereich mit dem Radius $nr_{col,t}(r_k)$ das Signal zu Rausch Verhältnis der Nachricht des neuen Senders. Erzeugen die Sender, die er in diesem Bereich findet,

3 Simulationsansätze

nicht genug Rauschen, damit es auf dem neuen Empfänger zu einer Kollision kommt, so gilt wieder, dass wie in 3.4.1 beschrieben $nr_{col,t}(r_k)$ vergrößert werden muss um ein aussagekräftiges SNR zu berechnen.

4 Implementierungsdetails

In diesem Kapitel wird auf die Verwendung von k-d-Bäumen, als die den neuen Transmissionsmodellen zugrundeliegende Datenstruktur, eingegangen. Es wird zunächst eine kurze Einführung in die Funktionsweise der k-d-Bäume gegeben und danach deren Verwendung in den Transmissionsmodellen erläutert.

4.1 Anforderungen an die zugrunde liegende Datenstruktur

Im Verlauf der Simulation kommt es wie in Kapitel 3 beschrieben immer wieder zu Situationen, in denen alle Knoten innerhalb einer Umgebung um einen Punkt gefunden werden müssen. Während dies im bisherigen CSMA/CA-Modell nur einmal beim Sendebeginn eines Knotens der Fall ist, müssen diese Anfragen bei den beiden neuen Transmissionsmodellen SimpleSNR und ExtendedSNR sehr viel häufiger beantwortet werden. Schon beim SimpleSNR-Modell muss bei jedem Sendevorgang mindestens eine solche Anfrage im Laufe des CCA beantwortet werden. In Schritt 3 kommt für jeden Empfänger innerhalb des Kommunikationsradius eine weitere Anfrage hinzu, um das initiale Rauschlevel bei den Empfängern zu ermitteln. Das ExtendedSNR-Modell benötigt sogar noch mehr Anfragen, um die Fälle zu behandeln, in denen die bisherige Größe des Anfragebereichs nicht ausreicht, um eine eindeutige Aussage treffen zu können.

Aus diesem Grund ist es aus Laufzeitgesichtspunkten essentiell wichtig, dass zur Beantwortung der Bereichsanfragen eine Datenstruktur zur Verfügung steht, die diese Aufgabe auf effiziente Art zu lösen vermag.

Ein erster Ansatz um diesen neuen Anforderungen zu begegnen, wurde bereits in Kapitel 3 vorgestellt, indem zusätzlich zur Menge aller Knoten, zwei weitere Mengen S_t und R_t gepflegt werden, die Verweise auf die Sender bzw. die Empfänger beinhalten. Auf diese Weise werden bei Bereichsanfragen, z.B. bezüglich der Sender in einem Bereich, nur relevante Knoten betrachtet und nicht mehr die gesamte Knotenmenge, die auch nicht sendende Knoten enthalten würde. Um diesen Ansatz weiter zu beschleunigen sollte für die Mengen S_t und R_t eine Datenstruktur gewählt werden, die in der Lage ist auf effiziente Art Bereichsanfragen zu beantworten.

Da sich die Mengen S_t und R_t bei jedem Sendebeginn und Abschluss durch das Hinzukommen, bzw. Wegfallen von Sendern und Empfängern verändern können, sollte die zugrunde liegende Datenstruktur ebenfalls sehr dynamisch sein. D.h. sie sollte in der Lage sein Einfüge- und Löschooperationen schnell durchzuführen.

4.2 k-d-Bäume als zugrundeliegende Datenstruktur

Eine Datenstruktur, die die in Abschnitt 4.1 genannten Anforderungen größtenteils erfüllt, ist ein k-d-Baum. Rolf Klein beschreibt diese Datenstruktur in seinem Buch *Algorithmische Geometrie*¹.

Ein k-d-Baum ist eine Verallgemeinerung des eindimensionalen Suchbaums auf die Dimension k .

4.2.1 Aufbau eines k-d-Baumes

Der Aufbau eines k-d-Baums lässt sich in Zeit $O(n \log(n))$ durchführen, wobei n die Anzahl der enthaltenen Punkte darstellt².

Dazu wird zunächst eine Splitkoordinate gewählt, z.B. die 1. Koordinate, also die x Koordinate. Die Punktmenge D wird nun nach der 1 Koordinate sortiert und es wird ein Splitwert s berechnet. Im Rahmen dieser Arbeit wird der Splitwert als der Mittelwert der Splitkoordinaten der Punkte an den Positionen $\frac{n}{2}$ und $\frac{n}{2} + 1$ gewählt. dabei wird an dieser Stelle noch vorausgesetzt, dass sich alle Punkte innerhalb der Knotenmenge in der jeweiligen Splitkoordinate unterscheiden, d. h., es lässt sich keine achsenparallele Gerade durch die Punktmenge legen, auf der mehr als ein Punkt der Punktmenge liegt. Es gibt verschiedene Möglichkeiten der Splitwertbestimmung für k-d-Bäume. Weitere Methoden werden z.B von Andrew W. Moore³ beschrieben. In dieser Arbeit wurde der genannte Ansatz aus numerischen Gründen gewählt, da bei einer solchen Wahl des Splitwertes die Wahrscheinlichkeit minimiert wird, dass teure Vergleiche von sehr ähnlichen *double* Werten vorkommen.

Die Punktmenge wird nun anhand des Splitwerts in zwei Teilmengen zerlegt, von denen eine alle Punkte enthält, deren Splitkoordinate kleiner ist als s und die andere alle übrigen mit größerer Splitkoordinate.

$$\begin{aligned} D_{<s} &= \{(x, y, \dots) : (x, y, \dots) \in D \wedge x < s\} \\ D_{>s} &= \{(x, y, \dots) : (x, y, \dots) \in D \wedge x > s\} \end{aligned} \quad (4.1)$$

Als nächstes wiederholt man den Vorgang rekursiv auf den beiden Teilmengen $D_{<s}$ und $D_{>s}$, nimmt jedoch in diesem Schritt die nächste, also die y Koordinate, und so weiter, bis nur noch einelementige Teilmengen übrig bleiben. Kommt man beim Teilen bis zur k . Koordinate und hat noch nicht ausschließlich einelementige Teilmengen, so wird im nächsten Sortier- und Teilschritt wieder die 1. Koordinate verwendet.

Auf diese Weise wird bei jedem Teilungsschritt ein Baumknoten erzeugt, dem der jeweilige Splitwert zugeordnet ist. Jeder Baumknoten beschreibt so im 2D Fall eine achsenparallele Splitgerade, die die Teilmenge der Punkte, in seinem Erzeugungsschritt in zwei Teile teilt. Für $k > 2$ entspricht der Splitwert einer Ebene, bzw. einer Hyperebene. Die Punkte innerhalb der einelementigen Teilmengen sind die Blätter des Baumes. Jedem Blatt ist durch die Erstellung der Splitgeraden ein Bereich des R^k zugeordnet.

¹ [9, S. 126ff]Rolf Klein - Algorithmische Geometrie

² [9, S. 134]Rolf Klein - Algorithmische Geometrie

³ [10]An introductory tutorial on k-d-trees

4.2 k-d-Bäume als zugrundeliegende Datenstruktur

Dieser Bereich ist nicht notwendigerweise beschränkt, enthält jedoch höchstens einen Knoten. Die Abbildungen 4.1 und 4.2 zeigen beispielhaft einen 2-d-Baum.

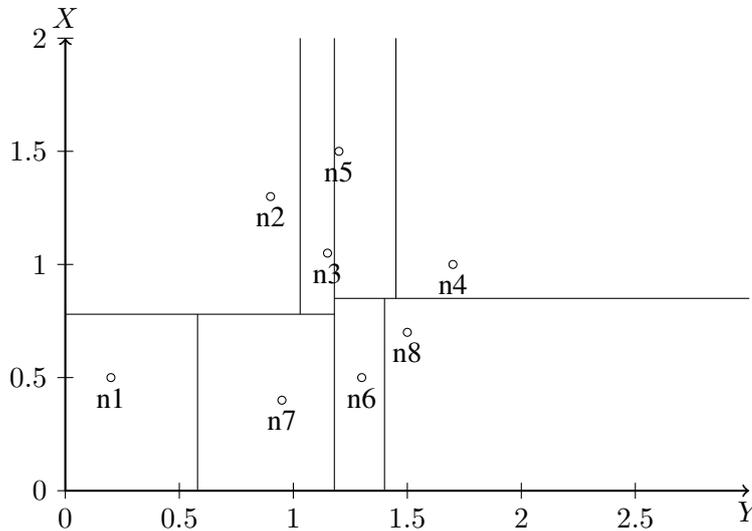


Abbildung 4.1: Ein zweidimensionaler k-d-Baum, der alle Knoten im Netzwerk enthält und ihnen jeweils ein Rechteck zuordnet.

4.2.1.1 k-d-Bäume auf beliebigen Punktmenge

Die Voraussetzung, dass es keine Knoten mit identischen Splitkoordinaten gibt, ist in der Praxis nicht realistisch, d. h., es ist nicht immer möglich innerhalb einer Punktmenge eine Splitgerade zu finden, auf der kein Punkt liegt. Aus diesem Grund wird noch eine zusätzliche Menge $D_{=s}$ eingeführt, die in jedem Schritt alle Punkte enthält, deren Splitkoordinate gleich dem Splitwert ist. Mit dieser Menge wird dann ein $(k-1)$ -d-Baum konstruiert, bei dem die aktuelle Splitkoordinate bei den Teilungsschritten nicht berücksichtigt wird. Auf diese Weise werden die Punkte in $D_{=s}$ auf den $k-1$ dimensionalen Subraum projiziert, den man unter Weglassung der Splitkoordinate erhält. Ein innerer Baumknoten verweist nun nicht mehr nur auf die Knoten die den Teilmengen unterhalb und oberhalb seiner Splitgeraden zugeordnet sind, sondern auch auf einen weiteren k-d-Baum, der alle Knoten enthält, die auf der Splitgeraden liegen. Der Baum ist also nicht mehr binär, sondern ternär. Dieses Vorgehen macht es notwendig, dass einem k-d-Baum immer bekannt ist, welche der k Dimensionen der ursprünglichen Punktmenge für ihn relevant sind.

Abbildung 4.3 beschreibt einen Klassenentwurf für einen k-d-Knoten.

Abbildung 4.4 zeigt eine Pseudocode Implementierung der Vorgehensweise beim Erstellen des Baumes. Das Zerlegen der Punktmenge geschieht bei dieser Implementierung rekursiv, indem bei jedem Rekursionsschritt die Start- und End-Indizes berechnet werden, die die Teilmengen begrenzen, die im nächsten Schritt behandelt werden müssen. Der Algorithmus beschränkt sich so in jedem Schritt auf die von den Indizes definierte Teilmenge der gesamten Punktmenge. Auf diese Weise wird unnötiges

4 Implementierungsdetails

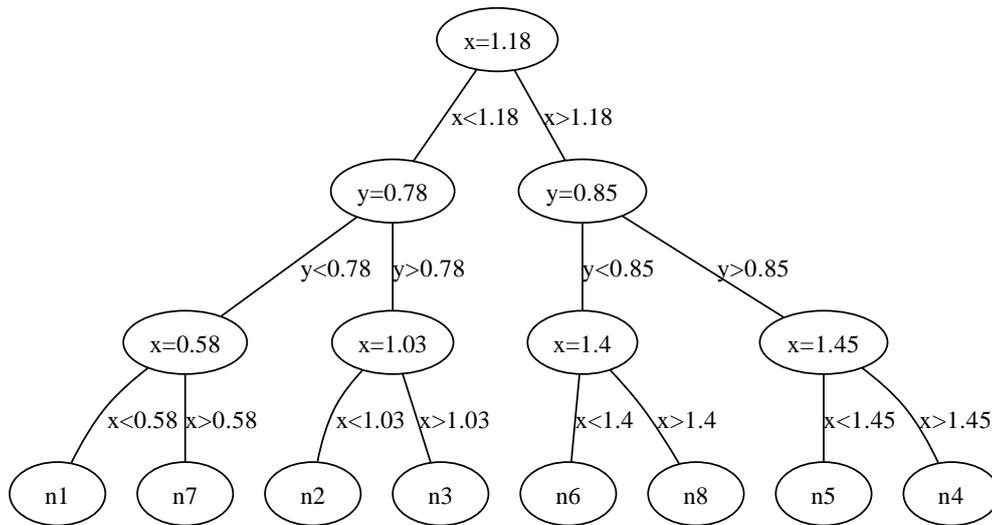


Abbildung 4.2: Die Knotenstruktur des in Abbildung 4.1 dargestellten k-d-Baumes. Jeder innere Knoten ist mit dem Wert der Splitgeraden beschriftet. Auf der untersten Ebene finden sich alle Blätter, die den Punkten aus Abbildung 4.1 entsprechen.

Kopieren beim Erzeugen der Teilmengen vermieden.

In jedem Schritt werden immer kleiner werdende Teilsequenzen der Gesamten Punktmenge nach der jeweiligen Splitkoordinate sortiert. Hierzu wird dem Algorithmus im ersten Schritt eine Liste *'dims'* aller k Dimensionen der Punktmenge übergeben, sowie der Index innerhalb dieser Liste, an dem die im aktuellen Schritt zu verwendende Splitdimension steht. Dieses zunächst etwas umständlich anmutende Verfahren zahlt sich dadurch aus, dass für beliebig viele Dimensionen eine einzige Methode benutzt werden kann um das Splitten und Sortieren anhand beliebiger Koordinaten zu behandeln.

Ein weiterer Vorteil wird beim Erzeugen der $(k-1)$ -d-Bäume mit den Punkten auf der Splitgeraden deutlich. Um einen solchen Baum zu erzeugen, kann die gleiche Methode verwendet werden, es muss lediglich die Splitdimension des Vaterknotens aus der Liste *'dims'* entfernt werden. Startet man den Algorithmus z.B. für $k = 3$ mit $dims = \{1; 2; 3\} = \{x, y, z\}$, so muss man nur die Konstruktionsmethode mit $dims = \{1; 3\}$ aufrufen um einen $(k - 1)$ -d-Baum zu erzeugen, der die Punkte in der Projektion auf die x, z Ebene verwaltet. Dieses geschieht in Zeile 7 des Listings.

4.2 k-d-Bäume als zugrundeliegende Datenstruktur

KDNODE

- | | | |
|---|-------------------|---|
| 1 | <i>Point</i> | ▷ k dimensionaler Punkt |
| 2 | <i>splitValue</i> | ▷ Wert der Splitgeraden, wenn innerer Knoten |
| 3 | KDNODE $D_{<s}$ | ▷ Linke Verzweigung des Baumes |
| 4 | KDNODE $D_{>s}$ | ▷ Rechte Verzweigung des Baumes |
| 5 | KDTREE $D_{=s}$ | ▷ (k-1)-d-Baum für Punkte auf der Splitgerade |
| 6 | <i>isLeaf</i> | ▷ Innerer Knoten oder Blatt? |
| 7 | <i>dimIndex</i> | ▷ Die Dimension der Splitgerade |
| 8 | <i>deleted</i> | ▷ Wurde dieser Knoten als gelöscht markiert? |
| 9 | <i>dims</i> | ▷ Die für diesen Baum relevanten Dimensionen? |

Abbildung 4.3: Klassenentwurf für einen k-d-Knoten

4 Implementierungsdetails

```

CONSTRUCTTREE(sortedPointList, startIndex, endIndex,
              dimIndex, dims)
1  splitValue
2  if SortByXPointList.size > 1
3    then
4      halfList  $\leftarrow$  startIndex +  $\lceil (\textit{endIndex} - \textit{startIndex}) / 2.0 \rceil - 1$ 
5      splitValue  $\leftarrow$  (sortedPointList [halfList] [dims [dimIndex]] +
                          sortedPointList [halfList + 1] [dims [dimIndex]]) / 2
6      onSplitValues  $\leftarrow$ 
          { $P_i : P_i \in \textit{sortedPointList} \wedge P_i[\textit{dims}[\textit{dimIndex}]] = \textit{splitValue}$ }
7      TreeOnSplit  $\leftarrow$  CONSTRUCTTREE(onSplitValues,
                                          0, |onSplitValues|, 0, dims \ dims [dimIndex])

8      lowerEndIndex  $\leftarrow$ 
           $\max(\{i : P_i \in \textit{sortedPointList} \wedge$ 
               $i \leq \textit{halfList} \wedge P_i[\textit{dims}[\textit{dimIndex}]] < \textit{splitValue}\})$ 
9      upperStartIndex  $\leftarrow$ 
           $\min(\{i : P_i \in \textit{sortedPointList} \wedge$ 
               $i \geq \textit{halfList} \wedge P_i[\textit{dims}[\textit{dimIndex}]] > \textit{splitValue}\})$ 
10     nextSplit  $\leftarrow$  dimIndex + 1
11     if dimIndex > |dims|
12       then
13         dimIndex  $\leftarrow$  0
14     SORT sortedPointList [startIndex, lowerEndIndex] BY dims [dimIndex]
15     SORT sortedPointList [upperStartIndex, endIndex] BY dims [dimIndex]

16     return INNERKDNODE(
          CONSTRUCTTREE(
            sortedPointList, startIndex, lowerEndIndex, dimIndex, dims),
          CONSTRUCTTREE(
            sortedPointList, upperStartIndex, endIndex, dimIndex, dims),
          TreeOnSplit)
17   else
18     if |sortedPointList| = 0
19       then return NULL
20     else return LEAFKDNODE(sortedPointList [0])

```

Abbildung 4.4: Aufbau eines K-d-Baumes aus einer Liste von Knoten.

4.2.2 Dynamische Eigenschaften eines K-d-Baums

4.2.2.1 Einfügeoperationen

Im Hinblick auf die Anforderungen, dass die den Mengen R_t und S_t zugrunde liegende Datenstruktur effizient mit Einfüge- und Löschooperationen umgehen können muss, stellt sich der *k-d*-Baum zunächst nicht als die ideale Wahl dar. Im Gegensatz zu einem gewöhnlichen Suchbaum gibt es keine Strategie, wie nach dem Einfügen bzw. dem Löschen eines Knotens eine Rebalancierung durchgeführt werden kann, ohne den gesamten Baum neu aufzubauen. Da Einfüge- und Löschooperationen während der Simulation sehr häufig auf den Knotenmengen durchgeführt werden, ist es aus Laufzeitgründen nicht möglich nach jeder Operation, die den Baum verändert, den Baum komplett neu aufzubauen um ihn in der Balance zu halten.

Aus diesem Grund wird während der Simulation ein nicht vollständig balancierter Baum in Kauf genommen. D.h. beim Einfügen neuer Knoten wird zunächst die Position gesucht, an der der Knoten anhand der schon bestehenden Struktur der Splitgeraden im Baum eingehängt werden müsste. Ist diese Position noch frei, so wird der Knoten einfach als neues Blatt hinzugefügt. Andernfalls wird das bestehende Blatt in einen inneren Knoten umgewandelt, dessen Splitgerade aus dem bestehenden Punkt und dem neu eingefügten berechnet wird. Der bestehende Knoten und der neue werden dann als Kinder des bisherigen Blattknotens in den Baum eingefügt. Dieses Vorgehen wird auch von Andrew W. Moore ⁴ beschrieben und ist im Pseudocode in Abbildung 4.5 dargestellt.

Wird der Baum auf diese Weise aufgebaut, so ist die Wahl der Splitgeraden nicht mehr ideal und der Baum bleibt nicht perfekt balanciert. Diese beiden Faktoren wirken sich nachteilig auf die Laufzeit bei Anfragen an den Baum aus. Untersuchungen im Rahmen dieser Arbeit haben jedoch gezeigt, dass ein unbalancierter *k-d*-Baum während der Simulation deutlich schneller ist, als ein perfekt balancierter, der ständig neu aufgebaut werden muss. Dies ist nicht verwunderlich, da die Punkte bei Simulationen in der Regel relativ gleichmäßig über das Simulationsgebiet verteilt sind. Es ist deshalb unwahrscheinlich, dass es während der Simulation zu einer zufälligen Einfügereihenfolge der Punkte in den Baum kommt, die dazu führt, dass der Baum so stark entartet, dass die Laufzeit der Anfragen auf das Niveau einer normalen Liste sinkt.

4.2.2.2 Löschooperationen

Löschooperationen werden ähnlich wie Einfügeoperationen behandelt, es wird dabei wiederum in Kauf genommen, dass der Baum nach dem Löschen nicht mehr perfekt balanciert ist. Der in dieser Arbeit gewählte Weg zum Löschen eines Knotens unterscheidet sich von dem im Buch "Algorithmische Geometrie" von Rolf-Klein ⁵ vorgeschlagenen schwachen Entfernen der Knoten. Die Knoten werden nicht lediglich als gelöscht markiert. Dies hätte zur Folge, dass sie bei Bereichsanfragen immer noch Einfluss auf die Laufzeit nehmen, da sie zwar im Ergebnis übergangen werden, bei

⁴ [10, S. 6-18]An introductory tutorial on *k-d*-trees

⁵ [9, S. 119ff]Rolf Klein - Algorithmische Geometrie

4 Implementierungsdetails

```

INSERT(point, parentDimIndex, KDNODE node)
1  if  $\neg$  node.isLeaf
2    then
3      if point[dims[node.splitIndex]] = node.splitValue
4        then INSERT(point, |dims|-1, node.D=s.root)
5        elseif point[dims[node.splitIndex]] <
           node.point[dims[node.splitIndex]]
6          then if node.lower = NULL
7            then node.lower  $\leftarrow$  LEAFKDNODE(point)
8            else INSERT(point, node.splitIndex, node.D<s)
9          else
10           if node.bigger = NULL
11             then node.bigger  $\leftarrow$  LEAFKDNODE(point)
12             else INSERT(point, node.splitIndex, node.D>s)
13 elseif node.point == point
14   then return  $\triangleright$  Keine Duplikate erlaubt
15 else
16   node.splitIndex  $\leftarrow$  parentDimIndex +1
17   if node.splitIndex > |dims|
18     then
19       node.splitIndex  $\leftarrow$  0
20   node.isLeaf  $\leftarrow$  FALSE
21   node.splitValue  $\leftarrow$  (point[dims[node.splitIndex]]
           + node.point[dims[node.splitIndex]])/2
22   if point[dims[node.splitIndex]] $\neg$  = node.point[dims[node.splitIndex]]
    $\triangleright$  Die Splitdimensionen sind nicht gleich.
    $\triangleright$  Transformiere das Blatt in einen inneren Knoten mit zwei Kindern
23   then
24     if point[dims[node.splitIndex]] <
           node.point[dims[node.splitIndex]]
25       then node.D<s  $\leftarrow$  LEAFKDNODE(point)
26       node.D>s  $\leftarrow$  LEAFKDNODE(node.point)
27       else node.D<s  $\leftarrow$  LEAFKDNODE(node.point)
28       node.D>s  $\leftarrow$  LEAFKDNODE(point)
29   else
    $\triangleright$  Die Split Dimensionen sind Gleich.
    $\triangleright$  Transformiere das Blatt in einen inneren Knoten ohne Kinder,
    $\triangleright$  mit Zwei Punkten in node.D=s
30   INSERT(point, |dims|-1, node.D=s.root)
31   INSERT(point, |dims|-1, node.D=s.root)

```

Abbildung 4.5: Einfügealgorithmus. Der hier beschriebene Algorithmus nimmt keine Rücksicht darauf, ob der Baum nach dem Einfügen noch balanciert ist. Er erhält jedoch die Bedingungen des k-d-Baums, bezüglich der Aufteilung der Punktmenge durch die Splitgeraden

4.2 *k*-d-Bäume als zugrundeliegende Datenstruktur

der Suche allerdings nach wie vor berücksichtigt werden. Stattdessen werden die Knoten echt aus dem Baum gelöscht.

Beim echten Löschen muss beachtet werden, dass wenn beide Blätter eines Knotens gelöscht werden, der Vaterknoten kinderlos ist und ebenfalls gelöscht werden muss, damit keine "toten" Pfade im Baum entstehen. Dies führt dazu, dass bei einer Löschanfrage der Algorithmus komplett bis zum zu löschenden Blatt absteigen muss, was in Zeit $O(\log(n))$ möglich ist. Anstatt nun wie beim schwachen Löschen den Vorgang zu beenden, müssen im schlimmsten Fall alle verwaisten Knoten auf dem Pfad zum gelöschten Blatt ebenfalls entfernt werden. Dies ist wiederum in Zeit $O(\log n)$ möglich. Auf diese Weise lässt sich ein Knoten im schlimmsten Fall in Zeit $O(2 \log n)$ löschen. Im besten Fall erzeugt das Löschen eines Blattes keinen einzigen verwaisten Knoten, so dass der Vorgang in Zeit $O(\log n)$ abgeschlossen werden kann, also genauso schnell wie das schwache Entfernen.

4.2.2.3 Dynamisierung von außen

In Kapitel "3.2 Dynamisierung" seines Buches⁶ beschreibt Rolf Klein wie sich Datenstrukturen "von aussen" dynamisieren lassen, so dass sie im Mittel effizient arbeiten. Um dies zu erreichen wird die gesamte Punktmenge anhand der Binärzerlegung der Punktzahl in Teilmengen zerlegt. Für jede dieser Teilmengen wird eine eigene Datenstruktur gepflegt.

Bei Einfügeoperationen werden jedes Mal ein Teil der Datenstrukturen verworfen und aus den darin enthaltenen Punkten und dem neuen Punkt wiederum neue Datenstrukturen aufgebaut, die die Punktmenge anhand der Binärzerlegung der Punktzahl unterteilen. Auf diese Weise wird bei jeder Einfügeoperation nur ein Teil der Datenstrukturen neu aufgebaut und nicht jedes Mal die gesamte Punktmenge zum Aufbau der neuen Struktur herangezogen. Die Teilstrukturen sind so immer perfekt balanciert und die Kosten um dies zu erreichen amortisieren sich über die gesamte Laufzeit, dadurch, dass die Zeit zur Beantwortung von Anfragen konstant bleibt.

Beim Löschen wird lediglich schwach gelöscht, also das zu löschende Blatt nur als gelöscht markiert. Enthält die gesamte Punktmenge zu einem Zeitpunkt genauso viele gelöschte wie nicht gelöschte Punkte, so werden alle Datenstrukturen verworfen und aus der Punktmenge der nicht gelöschten Knoten neu aufgebaut.

Wer sich für die beschriebene Technik im Detail interessiert sei auf das o.g. Buch von Rolf Klein verwiesen, da eine detaillierte Beschreibung der Herangehensweise den Rahmen dieser Arbeit sprengen würde.

Beim Anwenden der Dynamisierung von aussen auf den *k*-d-Baum, der für die Transmissionsmodelle verwendet wurde, wurde festgestellt, dass die Laufzeit im Vergleich zum kompletten Neubau nach jeder Einfüge- und Löschoperation zwar deutlich abnimmt, jedoch immer noch höher ist als die der Variante, die ohne Rücksicht auf Balancierung einfügt und löscht. Es wurden demnach für die weiteren Untersuchungen die Variante gewählt, die im vorherigen Abschnitt beschrieben ist.

⁶ [9, S. 110ff] Rolf Klein - Algorithmische Geometrie

4.2.3 Orthogonale Bereichsanfragen an einen k-d-Baum

Auf einem k-d-Baum können in Zeit $O(n^{1-\frac{1}{k}} + a)$ orthogonale Bereichsanfragen durchgeführt werden⁷. Unter einer orthogonalen Bereichsanfrage versteht man die Anfrage, welche Punkte sich in einem Bereich befinden, dessen Kanten Orthogonal zueinander und parallel zu den Koordinatenachsen stehen. In einem 2-dimensionalen Fall hat eine orthogonale Bereichsanfrage die Form eines achsenparallelen Rechtecks, siehe Abbildung 4.7. Im 3D Fall entspricht eine orthogonale Bereichsanfrage einem Würfel und für $k > 3$ einem Hyperwürfel.

Um eine Bereichsanfrage durchzuführen, muss für jedes Blatt, dessen ihm zugeordneter Bereich des R^k nicht disjunkt zum Anfragebereich ist, überprüft werden, ob der enthaltene Punkt innerhalb des Anfragebereichs liegt. Dies geschieht indem der Baum durchlaufen wird, wobei vor dem Abstieg nach $D_{<s}$ geprüft wird, ob die untere Grenze des Anfragebereichs in der entsprechenden Splitdimension des aktuellen Knotens kleiner ist als s . Entsprechend wird vor jedem Abstieg nach $D_{>s}$ geprüft ob die obere Grenze des Anfragebereichs in der entsprechenden Splitdimension größer ist als s . Schneidet die Splitgerade den Anfragebereich, so muss die Anfrage ebenfalls an dem $(k - 1)$ -d-Baum $D_{=s}$ des aktuellen Knotens weitergegeben werden, der alle Knoten auf der Splitgeraden enthält.

Eine solche Anfrage ist im Pseudocode in Abbildung 4.6 beschrieben. In der Arbeit wurde jedoch nicht die hier vorgeschlagene rekursive Vorgehensweise verwendet, sondern ein Iterator auf dem k-d-Baum implementiert, mit dem man sich, ähnlich wie bei andern C++ STL Containern, Iterativ von einem Punkt der Lösungsmenge zum nächsten hangeln kann, ohne dabei eine Liste der gesamten Lösungsmenge pflegen zu müssen.

4.2.4 Approximation des Rauschradius

Wie im vorangehenden Absatz beschrieben, lassen sich an einem k-d-Baum effiziente orthogonale Bereichsanfragen stellen. Es stellt sich jedoch das Problem, dass sich Funkwellen im allgemeinen kreisförmig und nicht rechteckförmig ausbreiten. Dementsprechend sind die Bereiche, die durch die während der Simulation gewählten Rauschradien beschrieben werden, ebenfalls kreisförmig.

Um dies zu berücksichtigen, wird der durch den Rauschradius beschriebene Kreis bei der Anfrage an den k-d-Baum durch ein Quadrat mit Kantenlänge gleich dem doppelten Rauschradius ersetzt, siehe Abbildung 4.7. Da die Fläche des dabei entstehenden Anfragequadrats größer ist als die des Kreises, der durch den Rauschradius beschrieben wird, könnte es dazu kommen, dass in den Ecken des Quadrates Knoten bei der Rauschberechnung mit einbezogen werden, die eigentlich außerhalb des Rauschradius liegen. Dieses Verhalten ist allerdings tolerierbar, da es in der Realität ohnehin zu einer unendlichen Ausbreitung des Signals käme. Dabei würden die Knoten in den Ecken ebenfalls benachrichtigt werden. Man kann also sagen, dass die zusätzliche Betrachtung der Knoten in den Ecken der orthogonalen Bereichsanfrage den Realismus der Simulation erhöhen.

⁷ [9, S. 134]Rolf Klein - Algorithmische Geometrie

4.2 k-d-Bäume als zugrundeliegende Datenstruktur

```
AREAQUERY(lowerBounds, upperBounds, ResultPointList, KDNODE node)
1  if node . isLeaf
2    then
3      if POINTISINSIDEAREA(node . point, lowerBounds, upperBounds)
4        then ResultPointList  $\leftarrow$  ResultPointList  $\cup$  {node . point}
5    else
6      if node . splitValue  $\geq$  lowerBounds[dims[node . splitIndex]]  $\wedge$ 
          node . splitValue  $\leq$  upperBounds[dims[node . splitIndex]]
7        then AREAQUERY(lowerBounds, upperBounds,
          ResultPointList, node . D=s . root)
8      if node . D<s  $\neq$  NULL  $\wedge$ 
          lowerBounds[dims[node . splitIndex]] < node . splitValue
9        then AREAQUERY(lowerBounds, upperBounds,
          ResultPointList, node . D<s)
10     if node . D>s  $\neq$  NULL  $\wedge$ 
          upperBounds[dims[node . splitIndex]] > node . splitValue
11     then AREAQUERY(lowerBounds, upperBounds,
          ResultPointList, node . D>s)
```

Abbildung 4.6: Pseudocode Implementierung einer orthogonalen Bereichsanfrage. Es wird deutlich, dass bei der Anfrage im schlimmsten Fall alle drei Söhne eines Knotens besucht werden müssen.

4.3 Strategien zur schrittweisen Erweiterung des Rausradius

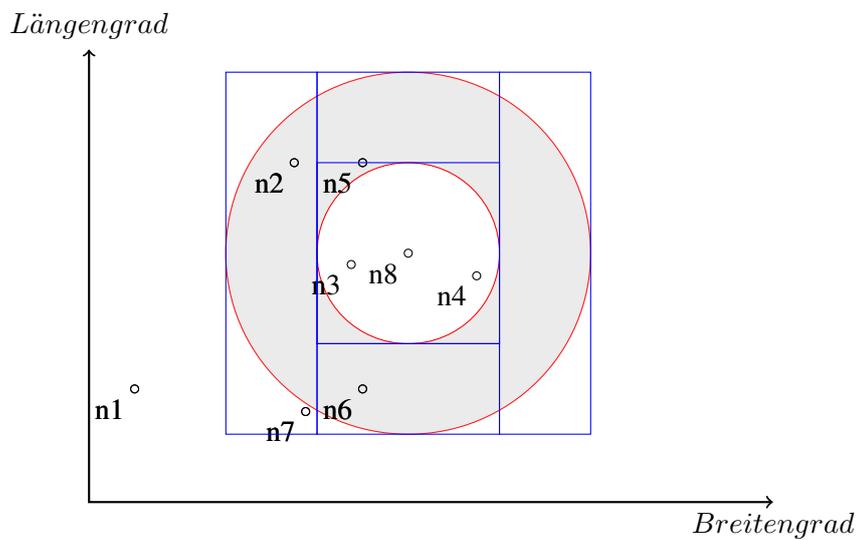


Abbildung 4.8: Bei einer Erweiterung des Rausradius (grauer Bereich) wird das den grauen Bereich einschliessende Quadrat in 5 Teilrechtecke unterteilt. Die Knoten innerhalb des inneren Quadrates sind schon bekannt aus der Anfrage bezüglich des bisherigen Rausradius. Die vier, das innerer Quadrat umschließenden Rechtecke, erweitern die Anfrage, so dass sie den gesamten erweiterten Rausradius mit einschließt. Für jedes der äußeren Rechtecke wird eine orthogonale Bereichsanfrage durchgeführt. Auf diese Weise wird vermieden, dass schon bekannte Bereiche doppelt durchsucht werden.

4 Implementierungsdetails

Rauschradius durch ein Quadrat mit Kantenlänge gleich dem Doppelten Rauschradius approximiert. Allerdings wird der k-d-Baum nicht nach allen Knoten innerhalb dieses Quadrats gefragt, da die Knoten innerhalb des bisherigen Rauschradius ja schon bekannt sind. Stattdessen werden im 2D Fall vier Anfragen durchgeführt mit Rechtecken, die das Anfragequadrat des bisherigen Rauschradius umschließen und so annäherungsweise eine ringförmige Anfrage erlauben. Das Vorgehen ist in Abbildung 4.8 illustriert.

Wird nun anhand von Satz 3.4.1 der Fall festgestellt, dass der benutzte Rauschradius nicht ausreichend ist, so wird

$$nr_t(r_k) = \sqrt{p_{max}(|S_t| - |S'_t|) \left(\frac{p(s_{dom}, r_k)}{10^{\frac{SNR_{min}}{10}}} - N_{S'_t} \right)^{-1}}$$

gewählt und anhand des oben beschriebenen Verfahrens das zusätzliche Rauschen von Sendern innerhalb des erweiterten Bereichs ermittelt. Danach wird die Größe des Rauschradius erneut überprüft und das Verfahren gegebenenfalls so lange wiederholt, bis der Rauschradius ausreichend groß ist. Auf diese Weise wird der Rauschradius schrittweise auf die notwendige Größe erweitert um eine sichere Aussage treffen zu können.

5 Evaluation

Im folgenden Kapitel werden die Simulationsergebnisse vorgestellt, die mit dem neuen Transmissionsmodell erzeugt wurden. Zunächst werden die Signalsimulation und die Simulation der Signal-zu-Rausch-Verhältnisse evaluiert. Im zweiten Abschnitt werden dann die Ergebnisse der Simulationen mit den verschiedenen Transmissionsmodellen, die in dieser Arbeit vorgestellt wurden, verglichen und diskutiert.

5.1 Anzahl der Knoten in der Simulationswelt

Zur Evaluation des Transmissionsmodells wurde als Simulationswelt eine Quadratischen Fläche, deren Seitenlänge einem Vielfachen der maximalen Kommunikationsreichweite $n \times cr_{max}$ entspricht, erstellt.

Um eine geeignete Anzahl an Knoten zu ermitteln, die bei einer zufälligen Verteilung auf der Simulationsfläche eine vollständige Vernetzung aller Knoten erlaubt, wird die Grundfläche mit einem Gitter der Seitenlänge cr_{max} überzogen. Positioniert man nun auf jedem Gitterpunkt der nicht auf dem Rand liegt einen Knoten, so erhält man eine Knotenverteilung bei der jeder Knoten 2-4 Nachbarn hat, die genau cr_{max} entfernt von ihm liegen. Eine solche Konfiguration ist in Abbildung 5.1 illustriert, hierbei sind $(n - 1)^2$ Gitterpunkte mit Knoten belegt.

Satz 5.1.1 (Knotendichte) *Gegeben sei eine quadratische Simulationsfläche mit Seitenlänge $n \times cr_{max}$, $n \in \mathbb{N}$. Positioniert man auf dieser Fläche $d \times (n - 1)^2$ mit $d \in \mathbb{R}$ Knoten, so habe die Simulation eine Knotendichte von d . Bei dieser Knotenzahl ist eine Anordnung der Knoten möglich, bei der im Umkreis von cr_{max} um jeden Gitterpunkt des in Abschnitt 5.1 definierten Gitters mindestens d Knoten liegen.*

Da die Punkte im Rahmen der Simulation zufällig auf dem Simulationsgebiet verteilt werden und nicht auf einem imaginären Gitter angeordnet werden, ist davon auszugehen, dass es bei einer Knotendichte von $d = 1$ zu keiner vollständigen Vernetzung aller Knoten miteinander kommt. Bei einer derartigen Dichte kommt es schnell zu nicht abgedeckten Räumen innerhalb des Simulationsgebietes, wenn mehrere Knoten dichter als cr_{max} beieinander liegen. Abbildung 5.2 verdeutlicht dies, indem sie die Knotenverteilungen für verschiedene Werte von d zeigt. Aus diesem Grund wurden für die meisten Simulationen Knotenverteilungen mit einer Dichte von $d = 3$ generiert.

5 Evaluation

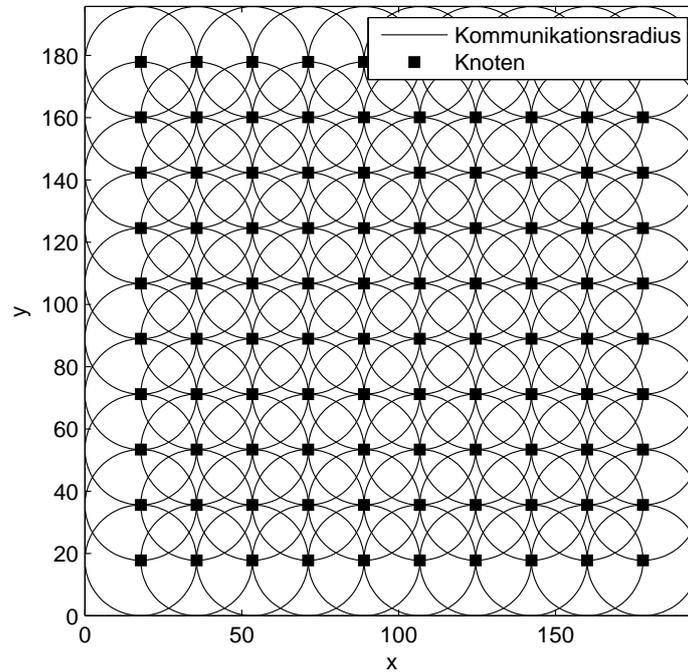


Abbildung 5.1: würde man auf jedem Gitterpunkt eines Gitter mit Seitenlänge cr_{max} einen Knoten positionieren, so hat jeder Knoten 2-4 Nachbarn, die genau cr_{max} von ihm entfernt sind. Im vorliegenden Bild entspricht die Seitenlänge $11cr_{max}$, wodurch 100 Knoten auf der Fläche verteilt werden.

5.2 Evaluations-Szenarien

Um Ergebnisse für die Evaluation zu erzeugen, wurden zwei einfache Simulationsszenarien herangezogen, die mit den beschriebenen Transmissionsmodellen simuliert wurden. Die beiden Szenarien werden im Folgenden beschrieben.

5.2.1 Hello-World

Das *Hello-World*-Szenario ist das simpelste getestete Szenario. Hierbei wird eine quadratische Welt mit einer zufälligen Knotenverteilung erzeugt. zu Beginn der Simulation versuchen sämtliche Knoten gleichzeitig genau eine broadcast Nachricht zu senden. Nach Abschluss des Sendens, ob erfolgreich oder nicht, inaktiviert sich der Knoten. Das Szenario endet, sobald alle Knoten inaktiv sind oder die maximale Anzahl an Simulationsrunden erreicht wurde.

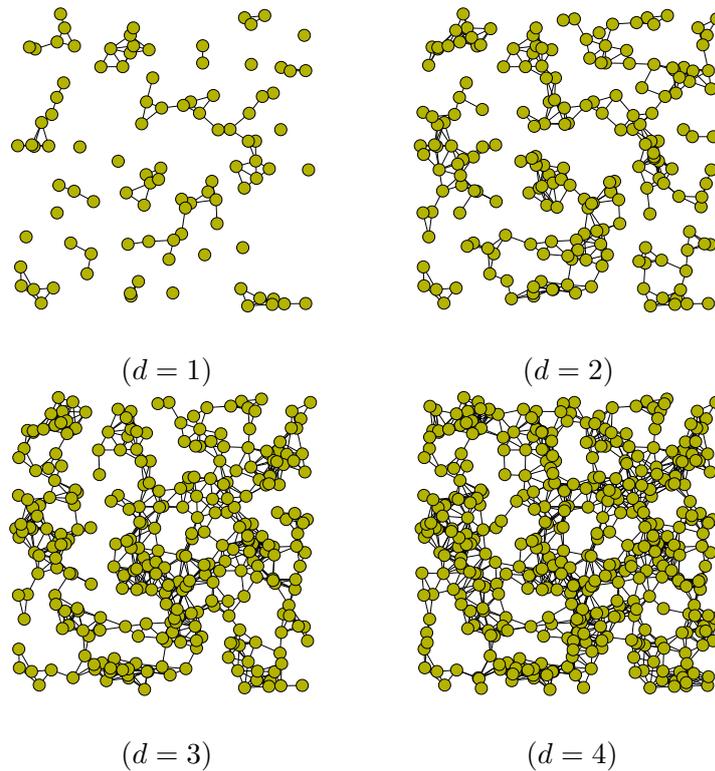


Abbildung 5.2: Knotenverteilungen auf einer Fläche von $(11cr_{max})^2$ mit zunehmender Dichte. Erst ab einer Knotendichte von 3 kommt es zu einer vollständigen Vernetzung aller Knoten miteinander. Die Knoten wurden mit identischer Zufallsaat generiert, was die Ähnlichkeit der Verteilungen erklärt.

5.2.2 Flood

Beim *Flood*-Szenario wird zu Beginn der Simulation ein einzelner Knoten gewählt. Dieser Knoten sendet nun eine Nachricht im Broadcast Modus, die von allen umliegenden Knoten empfangen wird. Jeder Knoten, der diese Nachricht empfängt, überprüft ob er die Nachricht schon einmal gehört hat. Ist ihm die Nachricht schon bekannt, so tut er nichts, andernfalls merkt er sich die Nachricht und versendet sie selbst erneut. Auf diese Weise breitet sich eine Nachricht vom initialen Sender ringförmig über das gesamte Simulationsgebiet aus, sofern die Knotendichte ausreichend hoch ist.

5.3 Evaluation der Signalsimulation

Mit Hilfe der Infrastruktur zur Berechnung der Signalstärken und der Signal-zu-Rausch-Verhältnisse, die den neuen Transmissionsmodellen zugrunde liegt, besteht die Möglichkeit zu jedem Simulationszeitpunkt ein Bild von der Verteilung der Signalintensität auf dem gesamten Simulationsgebiet zu zeichnen. Dies wurde verwendet um die Si-

mulation der Signalausbreitung zu evaluieren.

5.3.1 Simulation der Verteilung des Gesamtrauschens

Auf einen $195m \times 195m$ großen Simulationsgebiet mit 300 Knoten wurde ein Flood-Szenario simuliert. Diese Knotenverteilung entspricht laut Satz 5.1.1 einer Knotendichte von 3. Zu verschiedenen Zeitpunkten wurden das Simulationsgebiet mit einem Gitter aus Punkten überzogen die mindestens 1m voneinander entfernt sind. In jedem dieser Punkte wurde daraufhin das Gesamtrauschen zum Messzeitpunkt gemessen. Dies geschah, indem für jeden Punkt eine orthogonale Bereichsanfrage in Form eines Quadrats auf der Menge aller Sender durchgeführt wurde. Das Aufaddieren aller Signale, der so im Umkreis des aktuellen Messpunkts gefundenen Sender, ergibt dann das Gesamtrauschen in diesem Punkt.

Um ein visuellen Eindruck vom Einfluss der Größe der Approximationsfläche, also der Größe der orthogonalen Bereichsanfrage auf das Messergebnis zu erhalten, wurden die Messungen mit 4 verschiedenen Größen für das Anfragequadrat durchgeführt. Zunächst wurde eine Messung durchgeführt, bei der das Rauschen nur in einem Umkreis gleich dem maximalen Kommunikationsradius cr_{max} um jeden Punkt gemessen wurde. Die zweite Messung erfolgte in einem doppelt so großen Umkreis, die dritte mit dreifachem Radius und die letzte Messung über die gesamte Simulationsfläche. Die Ergebnisse dieser Messungen werden in Abschnitt 5.3.1 beschrieben. Exemplarisch werden in diesem Abschnitt die Ergebnisse der Messungen des Gesamtrauschens beschrieben, die sich zu einem Zeitpunkt ergaben, an dem 8 Sender gleichzeitig sendeten. Die Ergebnisse sind in den Abbildungen 5.3 und 5.4 dargestellt.

Die Abbildungen zeigen die Verteilung des Gesamtrauschens auf der ganzen Simulationsfläche in Form einer Konturkarte, ähnlich einer Wetterkarte mit Isobaren. Nicht sendende Knoten sind durch die kleinen Kreise dargestellt, während Sender mit einem kleinen x markiert sind. Die hellsten Bereiche kennzeichnen Bereiche, in denen das Gesamtrauschen kleiner als 10^{-9} ist. Die rosa Fläche, die jeden Sender umschließt, beschreibt den Bereich, innerhalb dessen das Rauschen stärker als $3,16 \cdot 10^{-8}$ ist. Innerhalb dieses rosa Bereichs liegt das Rauschen mindestens $10dB$ über der Empfängersensitivität, der Kanal gilt also als belegt. Darüber hinaus existiert um jeden Sender ein roter Kreis, welcher zur Orientierungshilfe den maximalen Kommunikationsradius cr_{max} darstellt.

Betrachtet man die Rauschverteilung der ersten Messung mit einem Approximationsradius von cr_{max} , so fällt zunächst die anscheinend quadratische Form der Signalausbreitung auf. Dies hängt mit der in Abschnitt 4.2.4 beschriebenen Verwendung von quadratischen Anfragen zur Approximation der kreisförmigen Signalausbreitung zusammen. Auf diese Eigenschaft sind auch die Unregelmäßigkeiten in den Niveaulinien der Messungen mit größerem Rauschradius zurückzuführen.

Abgesehen von den eckigen Strukturen, die am Rande der Approximationsradien im Umkreis der Sender durch die quadratische Anfrageform entstehen, zeigt sich, dass die Feldlinien erwartungsgemäß verlaufen. Dies wird besonders deutlich bei der Messung mit unendlich großem Approximationsradius. Hier entfallen die, durch die quadratische Anfrageform verursachten Artefakte und es zeigt sich eine gleichmäßige Aus-

breitung des Rauschens über die gesamte Simulationsfläche, abhängig von der Senderanzahl und Position. Es ist deutlich zu sehen, wie durch die Vergrößerung des Rauschradius in jedem Schritt ein genaueres Bild der Verteilung des Gesamtrauschens gezeichnet wird.

Überraschenderweise ist die Fläche der rosa Bereiche um die Sender, also die Fläche, auf der ein Sender den Kanal für andere Sender blockiert, deutlich kleiner, als die durch den Kommunikationsradius umschlossene Fläche. Selbst mehrere eng beieinander liegende Sender tragen nicht maßgeblich dazu bei, dass sich die Fläche stark vergrößert. Dies steht im Widerspruch zu der Herangehensweise des bisherigen CSMA/CA-Modells, bei dem ein sender Knoten den Kanal innerhalb des gesamten Kommunikationsbereichs blockiert. Dies zeigt einen Bereich, in dem sich das bisherige CSMA/CA-Modell erheblich anders verhält, als es in der Realität zu erwarten wäre. Diese Erkenntnis verdeutlicht auch, dass die in Definition 3.3.1 getroffene Annahme das $S_t \cap R_t = \emptyset$ ebenfalls nicht realistisch ist. Diese Annahme wurde ursprünglich aus dem bisherigen CSMA/CA-Modell übernommen. Es zeigt sich jedoch, dass das CCA es durchaus zulässt, dass ein Knoten mit dem Senden beginnt, obwohl er sich innerhalb des Kommunikationsradius eines anderen Senders befindet.

Betrachtet man die Abbildung, die mit dem kleinsten Approximationsradius erzeugt wurde, so fällt im Vergleich zur Abbildung mit unendlichem Approximationsradius auf, dass der Einfluss, den die Sender auf das Gesamtrauschen haben, deutlich über den Kommunikationsradius hinaus geht. Dies zeigt wiederum auf, wie ungenau die Simulation des bisherigen CSMA/CA-Modells ist, welches sich nur auf die Betrachtung des Kommunikationsradius eines Senders beschränkt. Für das bisherige CSMA/CA-Modell stellt sich die Simulationsumgebung also ähnlich dar wie in der Abbildung mit dem kleinsten Approximationsradius. Dies führt vor allem beim Erkennen von Kollisionen zu deutlichen Unterschieden, wie im nächsten Abschnitt bei der Evaluation der Signal-zu-Rausch-Verhältnisse deutlich wird.

5 Evaluation

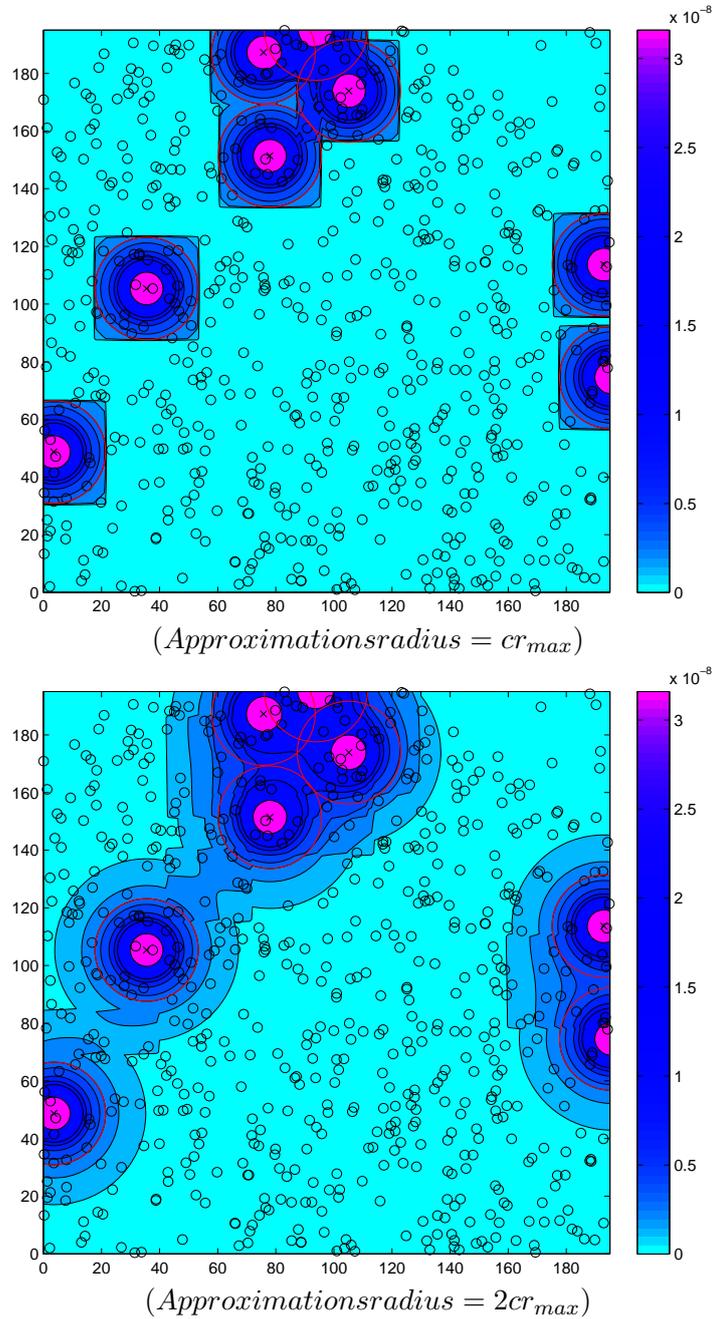


Abbildung 5.3: Ausbreitung des Gesamtrauschens aller Sender auf der Simulationsumgebung. Die Simulationsumgebung hat eine Kantenlänge von $11cr_{max}$ mit 300 Knoten.

5.3 Evaluation der Signalsimulation

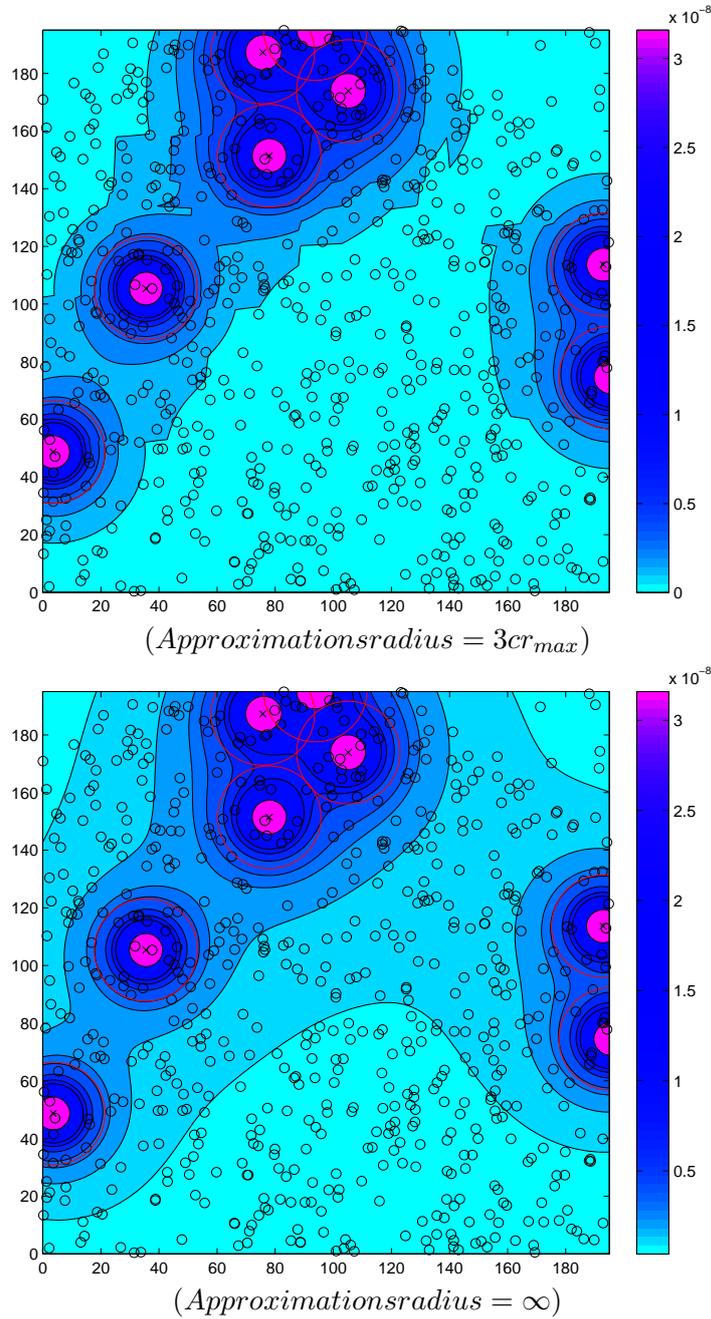


Abbildung 5.4: Ausbreitung des Gesamtrauschens aller Sender auf der Simulationsumgebung. Die Simulationsumgebung hat eine Kantenlänge von $11cr_{max}$ mit 300 Knoten.

5.3.2 Simulation der Signal-zu-Rausch-Verhältnisse

Im zweiten Schritt wurde in allen Punkten, ähnlich wie im vorherigen Abschnitt, das Signal-zu-Rausch-Verhältnis einer Nachricht gemessen, die von einem der zum Messzeitpunkt aktiven Sender ausgesendet wurde. Zu diesem Zweck wurde wiederum das Gesamtrauschen gemessen und danach anhand der Formel aus Definition 2.2.1 das Signal-zu-Rausch-Verhältnis der Nachricht auf dem gesamten Simulationsgebiet ermittelt. Die Ergebnisse dieser Messungen sind in den Abbildungen 5.5 und 5.6 dargestellt. Es wurde das Signal-zu-Rausch-Verhältnis des Senders gemessen, der oben links innerhalb der Gruppe von vier Sendern am oberen Rand des Simulationsgebietes liegt. Genau wie in Abschnitt 5.3.1 wurden vier Messungen mit wachsendem Approximationsradius durchgeführt. Die Bilder wurden während der gleichen Simulation und zum gleichen Zeitpunkt aufgenommen wie die aus Abschnitt 5.3.1. Die Bilder sind wiederum Konturkarten, die den Wert des Signal-zu-Rausch-Verhältnisses auf dem ganzen Simulationsgebiet darstellen. Dabei ist das Signal-zu-Rausch-Verhältnis umso kleiner, je dunkler die Fläche eingefärbt ist.

Aufgrund der Untersuchung zur Koexistenz von Drahtlosen Netzwerken die im IEEE 802.15.4-Standard dokumentiert sind, wird davon ausgegangen, dass eine Nachricht nur verstanden werden kann, wenn sie ein $SNR \geq 4dB$ aufweist. Bereiche, in denen dies der Fall ist, sind in den Abbildungen weiß dargestellt. Der Kommunikationsradius ist in diesem Fall aufgrund der Farbgebung der Konturkarte als blauer Kreis um die Sender zu sehen. Ausschließlich Knoten die in weißen Bereichen innerhalb des Kommunikationsradius eines Senders, also in den weißen Flächen innerhalb des blauen Kreises, sind in der Lage die Nachricht des Empfängers erfolgreich zu demodulieren. Auf allen Empfängern, die im restlichen Bereich des Kommunikationsradius liegen, kommt es zu einer Kollision der Nachricht mit anderen Übertragungen.

Auf den Bildern zeigt sich wie schon im vorherigen Abschnitt, insbesondere bei der Abbildung mit unendlichem Rauschradius, ein Bild der SNR Verteilung, das den Erwartungen entspricht. Die Signale können lediglich im Umkreis des Senders verstanden werden und das SNR wird umso kleiner, je näher man an andere Sender herankommt und je weiter man sich von dem betrachteten Sender entfernt. In gleicher Weise wird auch deutlich, wie sich die Verkleinerung des Approximationsradius auf die Genauigkeit der Simulation auswirkt.

Das erste Bild, mit einem Approximationsradius gleich dem maximalen Kommunikationsradius cr_{max} , entspricht in etwa dem Verhalten des bisherigen CSMA/CA-Modells. Es kommt ausschließlich in dem Bereich zu Kollisionen in denen sich die Kommunikationsradien zweier Sender schneiden. Allerdings wird schon bei dieser geringen Genauigkeitsstufe der erhöhte Realismus durch die Hinzunahme der Rauschberechnung sichtbar. Betrachtet man die beiden Empfänger, die direkt rechts neben dem Sender liegen und sich ebenfalls im Kommunikationsradius des Senders am oberen Rand des Simulationsgebietes befinden, so fällt auf, dass sie innerhalb der weißen Fläche liegen, die die Nachricht also erfolgreich demodulieren können. Beim bisherigen CSMA/CA-Modell wäre auch auf diesen Empfängern eine Kollision erzeugt worden, da sie sich im Kommunikationsradius von zwei Sendern befinden. Dies entspricht offensichtlich nicht der Realität, da die beiden Sender so nahe am Sender ihrer dominanten Nachricht liegen,

5.3 Evaluation der Signalsimulation

dass das Signal des zweiten Senders nicht stark genug ist um eine Kollision zu erzeugen.

Betrachtet man im weiteren Verlauf die Bilder die mit größerem Approximationsradius ermittelt wurden, so wird deutlich, wie sich der Bereich, in dem die Nachricht verstanden werden kann, immer weiter zusammenzieht, je mehr Rauschen von den anderen Sendern in der Simulation berücksichtigt wird. Aufgrund dessen kommt es auch auf Knoten, die nicht innerhalb des Kommunikationsradius von mindestens zwei Sendern liegen, zu Kollisionen von Nachrichten. Dies zeigt einmal mehr, wie sich die Signale aller Sender auch über deren Kommunikationsradius hinaus auf Empfänger von anderen Sendern auswirken und Kollisionen erzeugen können. Es legt nahe, dass die Methoden, die im bisherigen CSMA/CA verwendet werden eine zu vereinfachte Version der Realität simulieren.

5 Evaluation

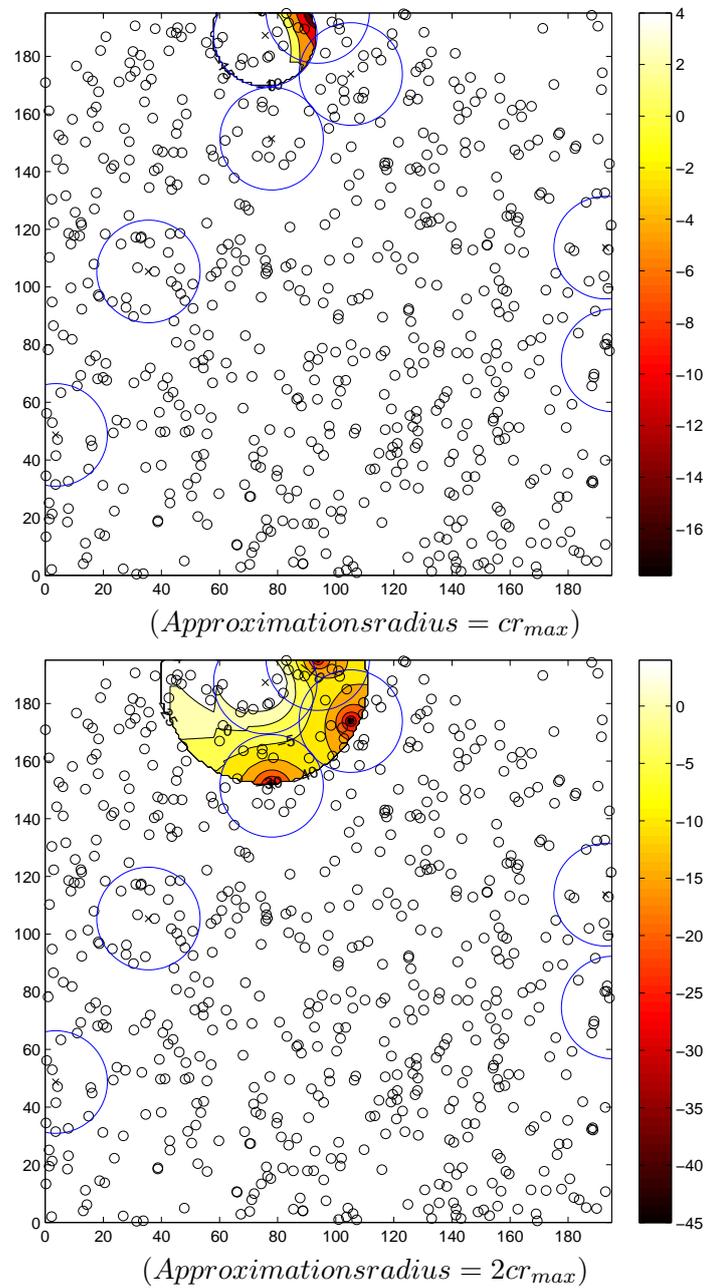


Abbildung 5.5: Ausbreitung des Signal-zu-Rausch-Verhältniss der Nachricht des Senders, der oben links innerhalb der Gruppe von vier Sendern am oberen Rand des Simulationsgebietes liegt. Die Simulationsumgebung hat eine Kantenlänge von $11cr_{max}$ mit 300 Knoten. Die Nachricht des betrachteten Senders kann nur in der Schnittfläche empfangen werden, die man erhält, wenn man die weiße Fläche mit dem blauen Kreis schneidet, der dem Kommunikationsradius des Senders entspricht.

5.3 Evaluation der Signalsimulation

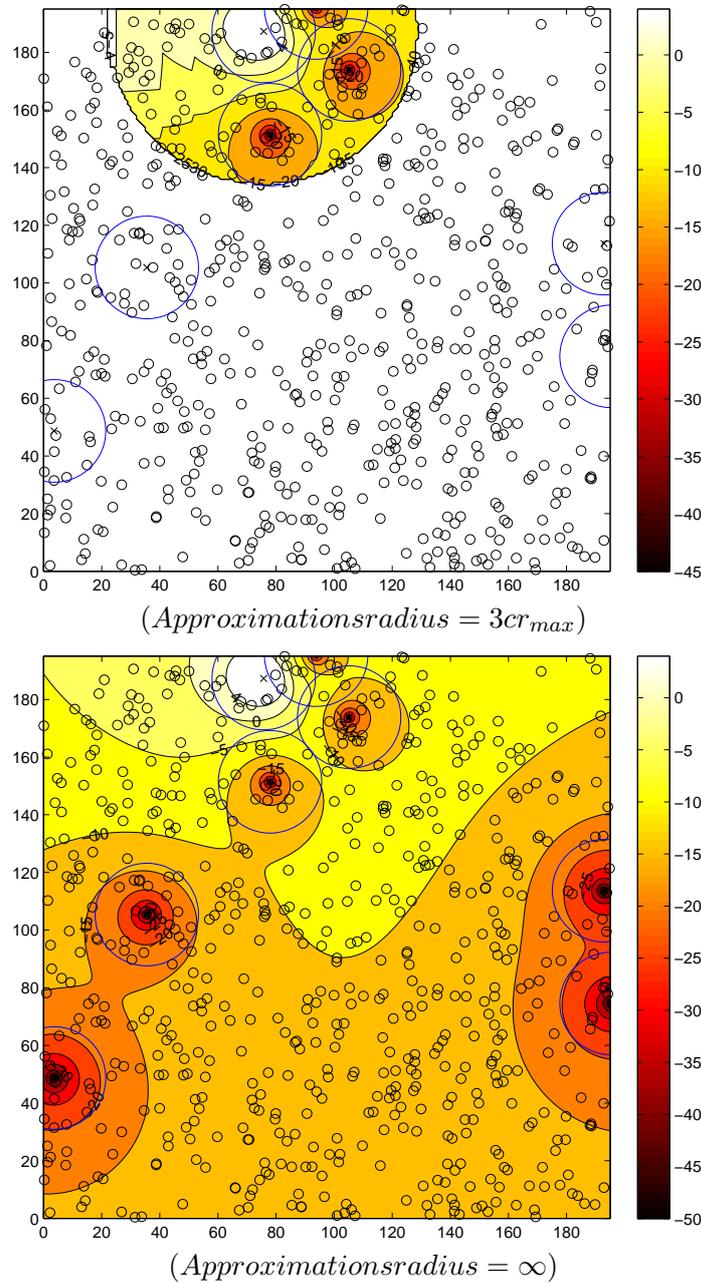


Abbildung 5.6: Ausbreitung des Signal-zu-Rausch-Verhältniss der Nachricht des Senders, der oben links innerhalb der Gruppe von vier Sendern am oberen Rand des Simulationsgebietes liegt. Die Simulationsumgebung hat eine Kantenlänge von $11cr_{max}$ mit 300 Knoten. Die Nachricht des betrachteten Senders kann nur in der Schnittfläche empfangen werden, die man erhält, wenn man die weiße Fläche mit dem blauen Kreis schneidet, der dem Kommunikationsradius des Senders entspricht.

5.4 Vergleich der Transmissionsmodelle untereinander

Nachdem im vorigen Abschnitt die Qualität der Signalsimulation evaluiert wurde, werden im nächsten Abschnitt die Transmissionsmodelle untereinander verglichen, um festzustellen, ob die Verbesserungen der einzelnen Modelle wie erwartet greifen.

5.4.1 Gütekriterien

Um die Güte des implementierten Transmissionsmodells zu untersuchen, werden die vorgestellten Modelle hinsichtlich der Kollisionswahrscheinlichkeit und ihrer Laufzeiten untereinander verglichen. Als Referenzmodell dient dabei eine Simulation des SimpleSNR-Modells, die mit einem unendlich großen Rauschradius durchgeführt wird. Bei einer solchen Simulation werden immer alle Sender und Empfänger bei der Ermittlung von Kollisionen mit einbezogen. Sie liefert also die größte Genauigkeit, die mit den vorgestellten Modellen zu erreichen ist.

5.4.2 Geeignete Größe des Rauschradius

Die Untersuchungen in Abschnitt 5.3 haben gezeigt, dass die Größe des Bereichs, über dem das Rauschen ermittelt wird, starken Einfluss auf das Simulationsergebnis hat. Um den Grad dieses Einflusses zu untersuchen, wurden Simulationen mit dem SimpleSNR-Modell durchgeführt und die Kollisionswahrscheinlichkeit bei steigendem Rauschradius beobachtet. Als Referenzwert für die Kollisionswahrscheinlichkeit wurde, wie schon erwähnt, eine Simulation mit dem SimpleSNR-Modell herangezogen, in der der Rauschradius so groß ist, dass er für jeden Knoten das gesamte Simulationsgebiet mit einschließt, dies entspricht sozusagen einem unendlichen Rauschradius. Auf diese Weise werden die Rauschverhältnisse und Kollisionen immer über alle Knoten in der Simulation ermittelt. Diese Konfiguration bietet also den größtmöglichen Realismus, der mit den vorgestellten Transmissionsmodellen zu erreichen ist. Die Ergebnisse dieser Untersuchungen sind in Abbildung 5.7 dargestellt. Um zu vergleichen, wie die automatische Bestimmung der Approximationsradien im ExtendedSNR-Modell im Vergleich zum SimpleSNR abschneidet, wurde das gleiche Szenario auch mit dem ExtendedSnr-Modell simuliert. Die Untersuchung wurde sowohl für das Flood- als auch für das Hello- World-Szenario vorgenommen.

Die Simulationen wurden auf einer quadratischen Simulationsfläche mit $5cr_{max}$ Kantenlänge durchgeführt, auf der 4800 Knoten verteilt wurden. Es wurden drei verschiedene Durchgänge à 20 Simulationen simuliert. Beim ersten Durchgang wurde das SimpleSNR-Modell verwendet, wobei der Rauschradius in jedem Schritt von anfänglich cr_{max} um cr_{max} erhöht wurde, bis zu einem Wert von $20cr_{max}$. Die Ergebnisse des ersten Durchgangs sind durch die rote Linie mit kreisförmigen Punkten gekennzeichnet.

Im zweiten Durchgang wurden die gleichen Simulationen mit jeweils zehnfachem Rauschradius durchgeführt. Diese Ergebnisse entsprechen dem oben genannten Referenzwert und sind als schwarze Linie mit quadratischen Punkten gekennzeichnet. Die

5.4 Vergleich der Transmissionsmodelle untereinander

Werte der Simulation sind ab dem fünften Durchlauf identisch, da ab diesem Durchlauf der Rauschradius mindestens genauso groß ist wie die Simulationsfläche. Dieser Wert entspricht der höchsten Simulationsgenauigkeit.

Die grüne Gerade mit x-förmigen Punkten, die darüber hinaus in den Bildern zu sehen ist, entspricht dem Ergebnis der Simulationen im dritten Durchgang. Bei diesen Simulationen wurde das ExtendedSNR-Modell verwendet. Da das ExtendedSNR-Modell keine Parametrisierung benötigt, sondern versucht sich aus den Daten der Simulationsumgebung selbst optimal zu parametrisieren, sind die Ergebnisse dieser Simulationen bei allen 20 Durchläufen identisch. Die dadurch entstehende Gerade in der Abbildung dient als Vergleichsgerade, um zu ermitteln, ab welcher Größe des Rauschradius die Ergebnisse des SimpleSNR-Modells vergleichbar oder besser sind als die des ExtendedSNR.

Beim Auswerten der Bilder stellt man fest, dass sich die Kollisionswahrscheinlichkeit der Simulationen mit dem SimpleSNR-Modell wie erwartet mit steigendem Rauschradius an den Referenzwert annähert, wohingegen die Kollisionswahrscheinlichkeit des ExtendedSNR-Modells bei allen Simulationsrunden gleich ist und sich nicht erheblich vom Referenzwert unterscheidet. Im Falle des Flood-Szenarios produziert das SimpleSNR-Modell erst ab einem Rauschradius von $8c_{r_{max}}$ Ergebnisse, die mindestens so gut sind wie die des SimpleSNR-Modells. Im Hello-World-Szenario ist dies sogar erst ab einem Rauschradius von $17c_{r_{max}}$ der Fall. Diese Ergebnisse liefern einen ersten Hinweis darauf, dass die Methoden zur automatischen Wahl der Approximationsradien des ExtendedSNR-Modells einen wirkungsvollen Mechanismus darstellen.

Um die Funktion der in Abschnitt 3.4.4 beschriebenen Methoden zur adaptiven Wahl des Benachrichtigungsradius weiter zu evaluieren, wurde die maximale Anzahl der Sender untersucht, die während einer Simulation gleichzeitig senden. Die Anzahl der aktiven Sender zu einem Zeitpunkt der Simulation geht unter anderem als bestimmende Größe sowohl in die Berechnung des Benachrichtigungsradius als auch in die Berechnung des initialen Rauschradius für neue Empfänger mit ein. Sie ist deswegen eine zentrale Variable bei der adaptiven Wahl des Rauschradius. Abbildung 5.8 zeigt die Ergebnisse der Untersuchungen.

Es wurden 10 Simulationen auf einer wachsenden quadratischen Simulationsfläche mit konstanter Knotendichte von drei durchgeführt. der 4. Simulationslauf entsprach dabei dem gleichen Szenario das schon bei den Auswertungen in in Abbildung 5.7 benutzt wurde. Zunächst fällt auf, dass das Hello-World-Szenario eine deutlich höhere Anzahl gleichzeitig sendender Knoten aufweist als das Flood-Szenario. Dies ist allerdings nachvollziehbar, da sich die "Welle" des Signals beim Flood-Szenario erst nach und nach über das Simulationsgebiet ausbreiten muss, während beim Hello-World-Szenario alle Knoten gleichzeitig versuchen zu senden.

Interessant ist, dass beim vierten Durchlauf der Simulationen beim Flood Szenario maximal ca. 15 Sender gleichzeitig aktiv waren und beim Hello-World-Szenario ca. 89. Setzt man diese Werte in die Formel zur Berechnung des Benachrichtigungsradius aus Satz 3.4.3 ein, so erhält man für das Flood-Szenario einen Wert von $6.34c_{r_{max}}$ und beim Hello-World-Szenario $14.95c_{r_{max}}$. Vergleicht man diese Werte für den Benachrichtigungsradius mit den Ergebnissen aus Abbildung 5.7, so stellt man fest, dass

5 Evaluation

die Kollisionswahrscheinlichkeit des SimpleSNR-Modells unter Verwendung eines Rauschradius der Größe $6.34c_{r_{max}}$ beim Flood-, bzw. $14.95c_{r_{max}}$ beim Hello-World-Szenario fast identisch ist mit der des ExtendedSNR-Modells. Dies deutet darauf hin, dass die Anzahl der aktiven Sender zu Recht als eine der wichtigsten Größen bei der Berechnung der adaptiven Rauschradien eingeht.

5.4 Vergleich der Transmissionsmodelle untereinander

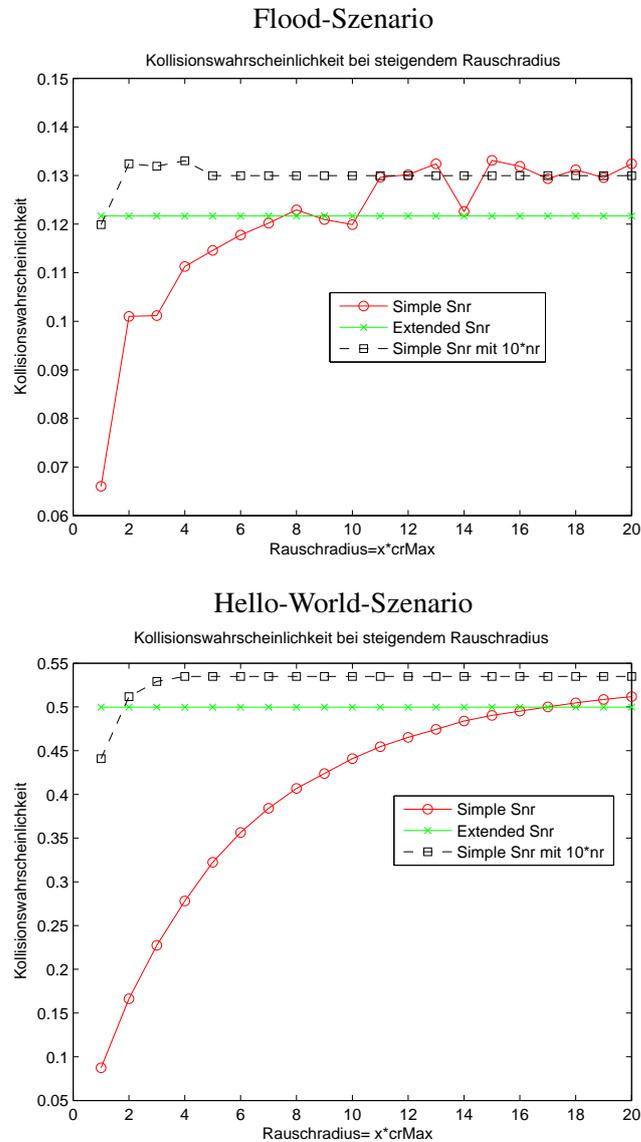


Abbildung 5.7: Einfluss des Rauschradius auf die Kollisionswahrscheinlichkeit. Es wurden auf einem quadratischen Simulationsgebiet mit Kantenlänge $= 5cr_{max}$ und 4800 Knoten, 3 Durchgänge à 20 Simulationen durchgeführt. 1. mit dem SimpleSNR-Modell mit steigendem Rauschradius, indem der Rauschradius bei jeder Simulationsrunde um cr_{max} erweitert wurde. 2. genau wie 1. nur mit jeweils zehnfachem Rauschradius. 3. Mit dem ExtendedSNR-Modell.

5 Evaluation

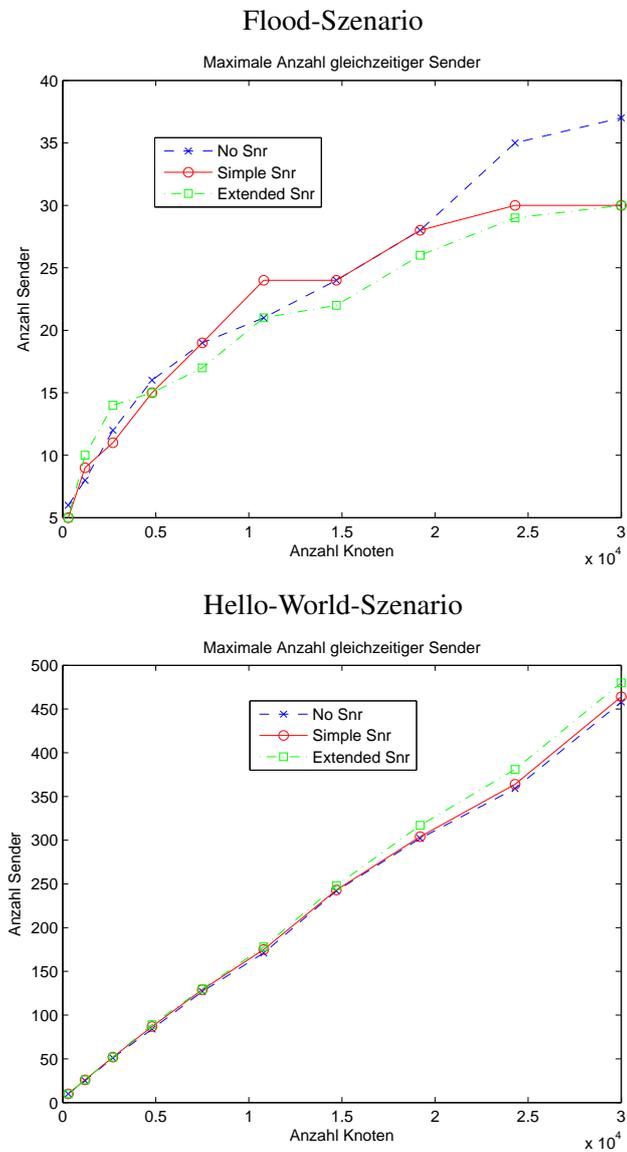


Abbildung 5.8: Maximale Anzahl gleichzeitig sendender Knoten, beim Flood-Szenario und beim Hello-World-Szenario. Es wurden 10 Simulationen auf einer quadratischen Simulationsfläche mit steigender Kantenlänge und Knotendichte = 3 durchgeführt.

5.4.3 Vergleich der Kollisionswahrscheinlichkeit bei verschiedenen Simulationsszenarien

In diesem Abschnitt soll verglichen werden, wie sich die Transmissionsmodelle auf verschiedenen Simulationskonfigurationen verhalten. Dazu wurden für alle beschriebenen Transmissionsmodelle jeweils 10 Simulationen mit wachsender quadratischer Simulationsfläche und steigender Knotenzahl durchgeführt. Dabei wurde bei allen Simulationen eine Knotendichte von 3 gewählt. Die Simulationsszenarien entsprechen damit denen, die auch schon bei den Untersuchungen zur maximalen Anzahl gleichzeitiger Sender verwendet wurden.

Bei jedem Simulationsdurchgang wurde für jedes Transmissionsmodell die Kollisionswahrscheinlichkeit über die gesamte Simulation ermittelt. Als Referenzwert zur Beurteilung der Güte der Ergebnisse wurde wiederum eine Simulation mit dem SimpleSNR-Modell mit unendlich großem Rauschradius durchgeführt. Darüber hinaus wurde eine weitere Simulation mit dem SimpleSNR-Modell durchgeführt. Bei dieser Simulation wurde ein Rauschradius verwendet, der den Werten aus Abschnitt 5.4.2 entspricht, ab denen das SimpleSNR-Modell mindestens genauso gute Ergebnisse liefert wie das ExtendedSNR-Modell. Diese Werte betragen $8cr_{max}$ beim Flood-Szenario bzw. $17cr_{max}$ beim Hello-World-Szenario. Die Ergebnisse dieser Simulationen sind in Abbildung 5.9 dargestellt, dabei wurde die Kollisionswahrscheinlichkeit gegen die Knotenanzahl aufgetragen.

Die Ergebnisse der Auswertungen zeigen, dass wie erwartet mit dem SimpleSNR und dem ExtendedSNR-Modell deutlich mehr Kollisionen auftreten, als bei den Simulationen mit dem bisherigen CSMA/CA-Modell. Beim NoSNR-Modell, dessen Sinn eine möglichst große Ähnlichkeit mit dem bisherigen CSMA/CA-Modell ist, ist die Kollisionswahrscheinlichkeit im Flood-Szenario fast identisch mit der des bisherigen CSMA/CA-Modells, während sie im Hello-World-Szenario etwa 10% darüber liegt. Da die Werte für die Kollisionswahrscheinlichkeit des bisherigen CSMA/CA-Modells aufgrund der bisherigen Erkenntnisse ohnehin als unrealistisch betrachtet werden können, wurden keine weiteren Anstrengungen unternommen zu untersuchen, welches der beiden Modelle im Falle des Hello-World-Szenarios plausible Werte liefert.

Interessanter ist, dass die Werte des SimpleSNR-Modells genauso wie die des ExtendedSNR-Modells relativ nah am Referenzwert liegen und eine ausreichend exakte Approximation der Kollisionswahrscheinlichkeit liefern. Besonders interessant ist der Verlauf der Kurven während der Simulationen mit dem Hello-World-Szenario. Hier ist zunächst das SimpleSNR-Modell im Vorteil, da es mit einem Rauschradius von $17cr_{max}$ anfänglich das gesamte Simulationsgebiet abdeckt. Ab der dritten Simulationsrunde, in der auf einem Quadrat mit einer Kantenlänge von $31cr_{max}$ simuliert wurde, kommt es jedoch schon zu Abweichungen der Kollisionswahrscheinlichkeit vom Referenzwert, da nicht mehr das gesamte Simulationsgebiet abgedeckt wird. Ab hier zeigt sich auch die Stärke des ExtendedSNR-Modells, das seine Rauschradien automatisch den gegebenen Simulationsparametern anpasst. Bei diesem Transmissionsmodell bleiben die Werte für die Kollisionswahrscheinlichkeit über alle 10 Simulationsrunden nahe beim Referenzwert, während sich die Werte des SimpleSNR-Modells mit zunehmender Knotenzahl deutlich vom Referenzwert entfernen.

5 Evaluation

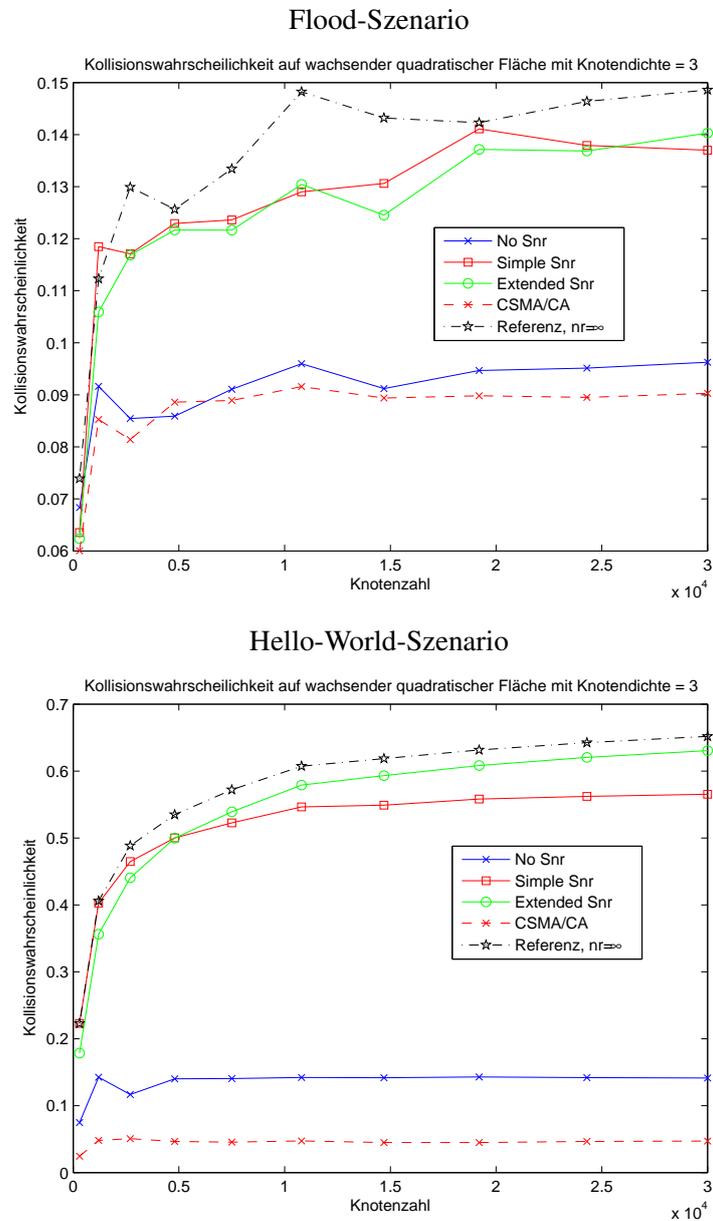


Abbildung 5.9: Entwicklung der Kollisionswahrscheinlichkeit bei steigender Knotenzahl auf wachsender quadratischer Fläche mit Dichte = 3. Die Simulation mit dem SimpleSNR wurde beim Flood-Szenario mit einem Rauschradius von $8c_{r_{max}}$ und beim Hello-World-Szenario mit einem Rauschradius von $17c_{r_{max}}$ durchgeführt. Der Referenzwert wurde durch eine Simulation mit dem SimpleSNR-Modell mit unendlichem Rauschradius ermittelt.

5.4.4 Laufzeitauswertungen

Als letzten Schritt zur Evaluation der neuen Transmissionsmodelle wurden die Laufzeiten der einzelnen Modelle gemessen und mit denen der anderen Modelle verglichen. Es wurden die Laufzeiten für die gleichen Simulationen gemessen, die schon bei den Untersuchungen in Abschnitt 5.4.3 verwendet wurden. Die Ergebnisse sind in Abbildung 5.10 dargestellt.

Zunächst fällt auf, dass die Laufzeiten des Flood-Szenarios generell deutlich kürzer sind als die des Hello-World-Szenarios. Dies hängt damit zusammen, dass wie schon in Abschnitt 5.4.2 festgestellt wurde, beim Flood-Szenario wesentlich weniger Sender gleichzeitig aktiv sind. Aus diesem Grund gibt es auch weniger Empfänger und es müssen somit deutlich weniger potentielle Kollisionssituationen während der Simulation analysiert werden. Gleichzeitig zeigen die Auswertungen, dass die Wahl des Szenarios und der Simulationsumgebung einen erheblichen Einfluss auf die Laufzeit haben.

Wie erwartet wird allgemein deutlich, dass sowohl die Laufzeiten des SimpleSNR-Modells als auch die des ExtendedSNR-Modells aufgrund der zusätzlichen Simulation der Signalstärken deutlich über denen des bisherigen CSMA/CA-Modells liegen. Besonders sticht im Falle des Hello-World-Szenarios der exponentielle Verlauf der Kurve heraus, während die Laufzeit des SimpleSNR-Modells annähernd linear wächst. Hierbei wird ein weiterer Unterscheid zwischen dem SimpleSNR-Modell und dem ExtendedSNR-Modell sichtbar.

Aufgrund des festen Rauschradius, der nicht erweitert wird, wird beim SimpleSNR-Modell das Rauschen, das auf den Empfängern wahrgenommen wird, kumulativ berechnet. Es wird dann allein aufgrund dieses Wertes entschieden, ob eine Kollision stattgefunden hat. Aus diesem Grund muss während der Kollisionsbehandlung keine weitere orthogonale Bereichsanfrage an die Menge aller Empfänger gestellt werden. Mit jedem zusätzlichen Sender kommen also nur höchstens N Bereichsanfragen beim Erstellen der neuen Empfänger hinzu. N bezeichnet hier die Anzahl der Empfänger innerhalb des Kommunikationsradius. Beim ExtendedSNR-Modell kommen für jeden Sender im schlimmsten Fall zu den N zusätzlichen Bereichsanfragen beim Erstellen der neuen Empfänger noch eine unbestimmte Anzahl an zusätzlichen Bereichsanfragen hinzu, die durchgeführt werden müssen, wenn festgestellt wird, dass der vorhandene Rauschradius nicht ausreicht um eine eindeutige Aussage zu treffen. Vor allem bei Simulationen mit hohen Knotenzahlen und vielen gleichzeitig aktiven Sendern tritt dieser Fall sehr oft ein. Des Weiteren werden beim ExtendedSNR-Modell auch bei der Kollisionsbehandlung weitere Bereichsanfragen fällig, wenn festgestellt wird, dass der Rauschradius nicht ausreichend ist. Dieser Umstand erklärt das exponentielle Wachstum der Laufzeiten beim Hello-World-Szenario.

Aufgrund der Untersuchungen in diesem Abschnitt wird deutlich, dass die erhöhte Simulationengenauigkeit, die durch das ExtendedSNR-Modell erlangt wurde, in bestimmten Situationen sehr stark zu Lasten der Laufzeit geht. Aus diesem Grund könnte es in bestimmten Situationen von Vorteil sein auf das SimpleSNR-Modell zurückzugreifen. Vor allem wenn man bereits Informationen über das Simulationsszenario besitzt wie die ungefähr erwartete Anzahl gleichzeitig aktiver Sender. Dies ist in der Regel jedoch

5 Evaluation

nicht der Fall und es ist relativ aufwändig eine gute Aussage über die richtige Größe des Rauschradius zu treffen. Aus diesem Grund sollte das ExtendedSNR-Modell verwendet werden, da sich zeigt, dass in weniger extremen Simulation, wie z.B. dem Flood-Szenario die Laufzeit des ExtendedSNR-Modells nicht sehr stark von der des SimpleSNR-Modells abweicht, bzw. teilweise sogar besser ist.

5.4 Vergleich der Transmissionsmodelle untereinander

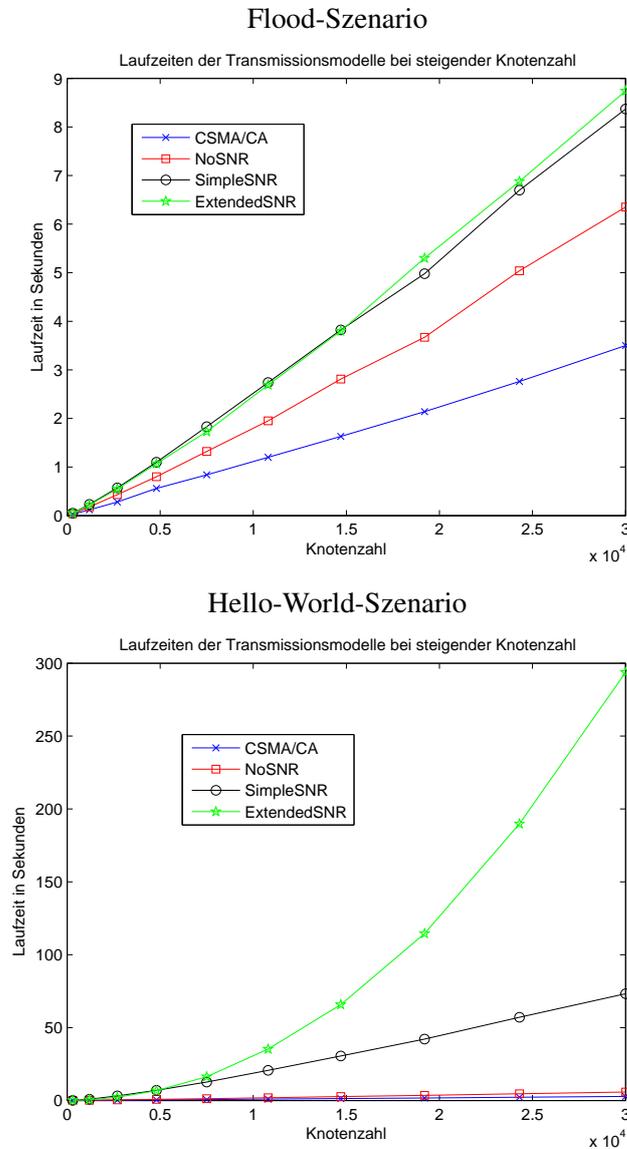


Abbildung 5.10: Entwicklung der Kollisionswahrscheinlichkeit bei steigender Knotenzahl auf wachsender quadratischer Fläche mit Dichte = 3. Die Simulation mit dem SimpleSNR wurde beim Flood- Szenario mit einem Rauschradius von $8cr_{max}$ und beim Hello-World-Szenario mit einem Rauschradius von $17cr_{max}$ durchgeführt. Der Referenzwert wurde durch eine Simulation mit dem SimpleSNR-Modell mit unendlichem Rauschradius ermittelt.

5 *Evaluation*

6 Zusammenfassung und Ausblick

In der vorliegenden Arbeit konnte gezeigt werden, dass auf der Grundlage der entwickelten Verfahren sowie der Verwendung zweier k-d-Bäume mit dem ExtendedSNR-Modell ein laufzeiteffizientes Modul implementiert werden konnte, welches den Ad-hoc Netzwerksimulator SHAWN in der Art erweitert, dass bei der Simulation der physikalischen Übertragung von Nachrichten Signalstärken und Signal-zu-Rausch-Verhältnisse mit einbezogen werden. Darüber hinaus wurde eine Methode vorgestellt und getestet, die es erlaubt dieses Modul ohne manuelle Konfiguration zu benutzen. Alle notwendigen Modulparameter werden während der Simulation adaptiv aus der aktuellen Simulationssituation errechnet. Dies macht die Verwendung des Moduls sehr einfach.

Die Untersuchungen zur Kollisionswahrscheinlichkeit, die durch dieses Modul berechnet wird, haben gezeigt, dass die Approximationsmechanismen des ExtendedSNR-Modells eine Genauigkeit bei der Ermittlung von Kollisionen erreichen, die in den meisten Fällen ausreichend und deutlich realistischer ist als die des bisherigen CSMA-/CA-Modells. Sollte eine Simulation es erfordern, dass eine noch höhere Genauigkeit erreicht werden muss, so besteht mit dem ebenfalls in dieser Arbeit entwickelten, SimpleSNR-Modell die Möglichkeit durch die Wahl eines unendlich großen Rauschradius eine Simulation mit größtmöglicher Genauigkeit durchzuführen. In den meisten Fällen ist das ExtendedSNR-Modell dem SimpleSNR-Modell jedoch vorzuziehen.

Es hat sich gezeigt, dass die erhöhte Genauigkeit des ExtendedSNR-Modells in bestimmten Situationen stark zu Lasten der Laufzeit gehen kann. Dies ist insbesondere bei Simulationsszenarien mit vielen gleichzeitig aktiven Sendern der Fall. Unter diesen Umständen kann es besser sein das SimpleSNR-Modell mit einem geeigneten Rauschradius zu verwenden. Da das Ermitteln eines geeigneten Rauschradius allerdings nicht ohne weiteres möglich ist und bei den meisten Simulationen nicht davon auszugehen ist, dass es zu solchen Situationen kommt, sollten diese Fälle nicht allzu oft auftreten.

Es ist geplant die im Rahmen dieser Arbeit entwickelten Verfahren in der Zukunft weiter zu verbessern und zu evaluieren, da sie sich als eine vielversprechende Erweiterung zur Verbesserung von Signalsimulationen durch Netzwerksimulatoren erwiesen haben.

6 Zusammenfassung und Ausblick

Im Einzelnen werden die folgenden nächsten Schritte in naher Zukunft durchgeführt:

- In der Evaluation hat sich gezeigt, dass die aus dem bisherigen CSMA/CA-Modell übernommene Annahme, dass die Menge der Sender und die der Empfänger disjunkt sind, nicht realistisch ist. Folgt man den Spezifikationen des IEEE 802.15.4-Standards, so kann es auch dazu kommen, dass ein Knoten, der sich bereits im Kommunikationsradius eines Senders befindet, ebenfalls mit dem Senden beginnt. Diese Erkenntnis in die Auswertungen dieser Arbeit mit einzubeziehen war aufgrund des zeitlichen Rahmens nicht mehr möglich. Es wird jedoch schnellstmöglich nachgeholt, damit eine realistische Simulation des CSMA/CA-Verfahrens entsprechend dem IEEE 802.15.4-Standard gewährleistet ist. Im Anschluss an diese Anpassungen wird eine detaillierte Untersuchung durchgeführt werden, die evaluiert, ob das vorgestellte Transmissionsmodell den CSMA/CA-Algorithmus hinreichend genau simuliert. In diesem Zusammenhang werden auch Vergleiche mit praktischen Tests durchgeführt werden.
- Im Rahmen dieser Arbeit wurden zwei einfache Simulationsszenarien, Flood und Hello-World, getestet. Weiterführende Untersuchungen der neuen Transmissionsmodelle mit komplexeren Simulationsszenarien sind geplant, um die Ergebnisse dieser Arbeit zu bestätigen. Hierfür bieten sich z.B. Tests zur Simulation komplexer Routingalgorithmen an.
- Insbesondere ist geplant, die Einschränkung auf rein statische Netzwerke fallen zu lassen, um auch Simulationen mit mobilen Knoten zu unterstützen. Dies kann aufbauend auf den in dieser Arbeit entwickelten Transmissionsmodellen einfach erreicht werden, da die intern von den Transmissionsmodellen verwendete Datenstruktur schon eine effiziente Berechnung der Nachbarschaften erlaubt. Durch eine Unterstützung mobiler Knoten würden einige Einfüge- und Löschoperationen auf der Datenstruktur hinzukommen. Es besteht jedoch die Hoffnung, dass sich diese Operationen nicht allzu negativ auf die Laufzeit auswirken, da bei der bisherigen Anwendung schon häufig Löscho- und Einfügeoperationen anfallen und die Datenstruktur dabei eine ausreichende Performanz aufwies.

Abbildungsverzeichnis

2.1	Hidden Station Kollision. n8 erhält vom CCA das Ergebnis "Kanal ist frei", da er außerhalb des Kommunikationsradius von n5 liegt. n8 sendet also seine Nachricht. Da n5 allerdings gerade ebenfalls eine Nachricht sendet und n3 im Kommunikationsradius von n5 liegt, geht die Nachricht von n8 bei n3 im Rauschen des Sendens von n5 unter. Die Nachrichten sind also auf dem Knoten n3 kollidiert.	7
2.2	Kollision durch gleichzeitiges Senden. n8 und n6 haben gleichzeitig ein CCA durchgeführt und den Kanal so als frei wahrgenommen. Sie fangen nach Abschluss des CCAs gleichzeitig an zu senden. Hierdurch kommt es zu einer Kollision der beiden Signale auf n7, da sich die Signale überlagern.	8
2.3	Ablaufdiagramm des CSMA/CA-Algorithmus. Der grau hinterlegte Bereich beschreibt die für diese Arbeit relevante "unslotted" Version des Algorithmus. <i>NB</i> bezeichnet die Anzahl der bisherigen Backoffs, <i>BE</i> bezeichnet den Backoff Exponent, der benutzt wird um die zufälligen Backoff Intervalle zu berechnen. <i>BE</i> wird während des Algorithmus Schrittweise von <i>macMinBe</i> auf <i>macMaxBE</i> erhöht. <i>macMaxCSMABackoffs</i> bezeichnet die maximale Anzahl, der Versuche auf das Medium zuzugreifen, bevor der Algorithmus aufgibt und den Zugriff als gescheitert erklärt. Abbildung aus IEEE 802.15.4-Standard [4]	10
3.1	Kollisionsbehandlung im bisherigen CSMA/CA-Modell. n5 hat vor n8 mit dem Senden begonnen, so dass seine Nachricht auf dem Knoten n3 empfangen wird. n8 kann in dieser Situation ebenfalls mit dem Senden beginnen, da n8 außerhalb des Kommunikationsradius von n5 liegt, also ein "Kanal frei" vom CCA bekommt. Dabei erzeugt n8 eine Kollision auf dem Knoten n3 und zerstört die dort empfangene Nachricht. Dies ist jedoch unrealistisch, da n8 sehr weit von n3 entfernt ist, während n5 sehr nahe an n3 liegt. Aus diesem Grund sollte das Signal von n5 bei n3 immer noch stark genug sein, so dass das Rauschen von n8 die Übertragung nicht stört.	20
3.2	CCA, Knoten n8 sendet eine Nachricht (u.a. an Knoten n6), dabei erzeugt er Rauschen auf allen Knoten, die innerhalb des Kreises, der durch den Rauschradius <i>nr</i> beschrieben wird, liegen. Außerhalb des Kreises ist das Signal, das von n8 ausgesendet wird aufgrund des exponentiellen Abfalls zu schwach um noch einen nennenswerten Beitrag zum Rauschen zu liefern, das z.B. auf Knoten n7 empfangen wird . . .	23

Abbildungsverzeichnis

4.1	Ein zweidimensionaler k-d-Baum, der alle Knoten im Netzwerk enthält und ihnen jeweils ein Rechteck zuordnet.	37
4.2	Die Knotenstruktur des in Abbildung 4.1 dargestellten k-d-Baumes. Jeder innere Knoten ist mit dem Wert der Splitgeraden beschriftet. Auf der untersten Ebene finden sich alle Blätter, die den Punkten aus Abbildung 4.1 entsprechen.	38
4.3	Klassenentwurf für einen k-d-Knoten	39
4.4	Aufbau eines K-d-Baumes aus einer Liste von Knoten.	40
4.5	Einfügealgorithmus. Der hier beschriebene Algorithmus nimmt keine Rücksicht darauf, ob der Baum nach dem Einfügen noch balanciert ist. Er erhält jedoch die Bedingungen des k-d-Baums, bezüglich der Aufteilung der Punktmenge durch die Splitgeraden	42
4.6	Pseudocode Implementierung einer orthogonalen Bereichsanfrage. Es wird deutlich, dass bei der Anfrage im schlimmsten Fall alle drei Söhne eines Knotens besucht werden müssen.	45
4.7	Die Anfrage nach allen Knoten innerhalb des Rauschradius von Knoten n_8 wird durch eine orthogonale Anfrage in Form eines Quadrates mit Kantenlänge gleich dem Rauschradius approximiert. Hierdurch liegt zusätzlich Knoten n_7 im Anfragebereich, der eigentlich außerhalb des Rauschradius liegt. Aufgrund der unendlichen Ausbreitung des Signals in der Realität ist die zusätzliche Betrachtung von n_7 allerdings zu tolerieren, sie erhöht quasi den Realismus der Simulation.	46
4.8	Bei einer Erweiterung des Rauschradius (grauer Bereich) wird das den grauen Bereich einschliessende Quadrat in 5 Teilrechtecke unterteilt. Die Knoten innerhalb des inneren Quadrates sind schon bekannt aus der Anfrage bezüglich des bisherigen Rauschradius. Die vier, das innerer Quadrat umschließenden Rechtecke, erweitern die Anfrage, so dass sie den gesamten erweiterten Rauschradius mit einschließt. Für jedes der äußeren Rechtecke wird eine orthogonale Bereichsanfrage durchgeführt. Auf diese Weise wird vermieden, dass schon bekannte Bereiche doppelt durchsucht werden.	47
5.1	würde man auf jedem Gitterpunkt eines Gitter mit Seitenlänge cr_{max} einen Knoten positionieren, so hat jeder Knoten 2-4 Nachbarn, die genau cr_{max} von ihm entfernt sind. Im vorliegenden Bild entspricht die Seitenlänge $11cr_{max}$, wodurch 100 Knoten auf der Fläche verteilt werden.	50
5.2	Knotenverteilungen auf einer Fläche von $(11cr_{max})^2$ mit zunehmender Dichte. Erst ab einer Knotendichte von 3 kommt es zu einer vollständigen Vernetzung aller Knoten miteinander. Die Knoten wurden mit identischer Zufallssaat generiert, was die Ähnlichkeit der Verteilungen erklärt.	51
5.3	Ausbreitung des Gesamttrauschens aller Sender auf der Simulationsumgebung. Die Simulationsumgebung hat eine Kantenlänge von $11cr_{max}$ mit 300 Knoten.	54

5.4 Ausbreitung des Gesamtrauschens aller Sender auf der Simulationsumgebung. Die Simulationsumgebung hat eine Kantenlänge von $11cr_{max}$ mit 300 Knoten. 55

5.5 Ausbreitung des Signal-zu-Rausch-Verhältniss der Nachricht des Senders, der oben links innerhalb der Gruppe von vier Sendern am oberen Rand des Simulationsgebietes liegt. Die Simulationsumgebung hat eine Kantenlänge von $11cr_{max}$ mit 300 Knoten. Die Nachricht des betrachteten Senders kann nur in der Schnittfläche empfangen werden, die man erhält, wenn man die weiße Fläche mit dem blauen Kreis schneidet, der dem Kommunikationsradius des Senders entspricht. . . 58

5.6 Ausbreitung des Signal-zu-Rausch-Verhältniss der Nachricht des Senders, der oben links innerhalb der Gruppe von vier Sendern am oberen Rand des Simulationsgebietes liegt. Die Simulationsumgebung hat eine Kantenlänge von $11cr_{max}$ mit 300 Knoten. Die Nachricht des betrachteten Senders kann nur in der Schnittfläche empfangen werden, die man erhält, wenn man die weiße Fläche mit dem blauen Kreis schneidet, der dem Kommunikationsradius des Senders entspricht. . . 59

5.7 Einfluss des Rauschradius auf die Kollisionswahrscheinlichkeit. Es wurden auf einem quadratischen Simulationsgebiet mit Kantenlänge = $5cr_{max}$ und 4800 Knoten, 3 Durchgänge à 20 Simulationen durchgeführt. 1. mit dem SimpleSNR-Modell mit steigendem Rauschradius, indem der Rauschradius bei jeder Simulationsrunde um cr_{max} erweitert wurde. 2. genau wie 1. nur mit jeweils zehnfachem Rauschradius. 3. Mit dem ExtendedSNR-Modell. 63

5.8 Maximale Anzahl gleichzeitig sendender Knoten, beim Flood-Szenario und beim Hello-World-Szenario. Es wurden 10 Simulationen auf einer quadratischen Simulationsfläche mit steigender Kantenlänge und Knotendichte = 3 durchgeführt. 64

5.9 Entwicklung der Kollisionswahrscheinlichkeit bei steigender Knotenzahl auf wachsender quadratischer Fläche mit Dichte = 3. Die Simulation mit dem SimpleSNR wurde beim Flood-Szenario mit einem Rauschradius von $8cr_{max}$ und beim Hello-World-Szenario mit einem Rauschradius von $17cr_{max}$ durchgeführt. Der Referenzwert wurde durch eine Simulation mit dem SimpleSNR-Modell mit unendlichem Rauschradius ermittelt. 66

5.10 Entwicklung der Kollisionswahrscheinlichkeit bei steigender Knotenzahl auf wachsender quadratischer Fläche mit Dichte = 3. Die Simulation mit dem SimpleSNR wurde beim Flood- Szenario mit einem Rauschradius von $8cr_{max}$ und beim Hello-World-Szenario mit einem Rauschradius von $17cr_{max}$ durchgeführt. Der Referenzwert wurde durch eine Simulation mit dem SimpleSNR-Modell mit unendlichem Rauschradius ermittelt. 69

Abbildungsverzeichnis

Literaturverzeichnis

- [1] Peter Rothenpieler, Daniela Krüger, Dennis Pfisterer, Stefan Fischer, Denise Dudek, Christian Haas, Andreas Kuntz, and Martina Zitterbart. FleGSens - secure area monitoring using wireless sensor networks. In *Proceedings of the 4th Safety and Security Systems in Europe*, 2009.
- [2] Martin Werner. *Nachrichtentechnik: Eine Einführung für alle Studiengänge*. Vieweg+Teubner, 2009.
- [3] Qizheng Gu. *RF system design of transceivers for wireless communications*. Springer, 2005.
- [4] IEEE Computer Society. Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs). specification, IEEE Computer Society, 2006.
- [5] University of Southern California, Information Sciences Institute (ISI). Ns-2: Network simulator-2. <http://www.isi.edu/nsnam/ns/>.
- [6] Alexander Kröller, Dennis Pfisterer, Carsten Buschmann, Sandor P. Fekete, and Stefan Fischer. Shawn: A new approach to simulating wireless sensor networks. In *Design, Analysis, and Simulation of Distributed Systems 2005 (DASD' 05)*, pages 117–124, April 2005.
- [7] Sandor P. Fekete, Alexander Kröller, Stefan Fischer, and Dennis Pfisterer. Shawn: The fast, highly customizable sensor network simulator. In *Proceedings of the Fourth International Conference on Networked Sensing Systems (INSS' 07)*, June 2007.
- [8] Torsten Teubler. Optimierung, Implementierung und Evaluation von Routing-Protokollen unter Verwendung globalen Wissens. Master's thesis, Universität zu Lübeck, Mai 2009.
- [9] Rolf Klein. *Algorithmische Geometrie - Grundlagen, Methoden, Anwendungen*. Springer-Verlag Berlin Heidelberg, 2 edition, 2005.
- [10] An introductory tutorial on kd trees. *Andrew W. Moore*. Carnegie Mellon University, 1991.