



Fakultät für Informatik

Funkstandardübergreifende Kommunikation in Mobilien Ad Hoc Netzwerken

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur
(Dr.-Ing.)

vorgelegt

der Fakultät für Informatik
Technische Universität Chemnitz

von: Dipl.-Inf.(Univ.) Matthias Vodel
geboren am: 22.03.1982
geboren in: Rodewisch

Gutachter: Prof. Dr. Wolfram Hardt
Prof. Dr. Thomas Bauschert
Prof. Dr. Martin Gaedke

Danksagung

Diese Arbeit stellt den Abschluss einer wichtigen Phase meines Lebens und meiner Arbeit an der TU Chemnitz dar. Ich möchte daher die Gelegenheit nutzen, einigen Menschen zu danken, die mich auf dem Weg durch diesen Lebensabschnitt begleitet und in unterschiedlichen Bereichen unterstützt haben.

An erster Stelle möchte ich mich bei Prof. Dr. Wolfram Hardt für die hervorragende, konstruktive Betreuung bedanken. Er nahm sich regelmäßig Zeit für wissenschaftliche Diskussionen, die wesentlich zum Erfolg dieser Arbeit beitrugen. Nur durch das von ihm entgegengebrachte Vertrauen war es mir möglich, neue Ideen offen und kreativ umzusetzen.

Bei Prof. Dr. Thomas Bauschert bedanke ich mich für die sehr gute Zusammenarbeit und die aufgebrauchte Zeit. Die konstruktiven Rückmeldungen sowie die vielen neuen Impulse bereicherten diese Arbeit.

Seit mehr als drei Jahren arbeite ich nun an der Professur Technische Informatik der TU Chemnitz. All meinen Kollegen möchte ich danken, da Sie mir für fachliche Fragen immer zur Verfügung standen.

Mein besonderer Dank gilt Mirko Caspar, der mir an unzähligen Tagen als kompetenter Diskussionspartner und Freund zur Seite stand und sich dabei zu keinem Zeitpunkt aus der Ruhe bringen ließ.

Ein großer Dank auch an Wolfgang Lang, Mirko Lippmann, Daniel Opp und Matthias Sauppe für die Unterstützung in den verschiedenen Teilaspekten der Arbeit.

Auch meinen Freunden möchte ich an dieser Stelle danken, da sie trotz einiger zeitlichen Einschränkungen immer ein offenes Ohr für meine Anliegen hatten.

Meinen Eltern danke ich für das Vertrauen und die Unterstützung über nunmehr unzählige Jahre hinweg. Sie ließen mich meinen Weg gehen, förderten mich an den richtigen Stellen und haben mir so den Weg in das Informatikstudium geebnet.

Mein größter Dank gilt der Liebe meines Lebens, Stefanie, die mir all die Jahre Rückhalt und Kraft gegeben hat. Sie schenkte mir die so wichtige Geborgenheit und hat mich durch viele Höhen und Tiefen hindurch begleitet. Nicht zuletzt war sie es, die zahlreiche Einschränkungen auf sich nahm, um mir das Schreiben und Vollenden dieser Dissertation zu ermöglichen.

... Mi amas vin ...

Kurzfassung

Der kombinierte Einsatz selbstorganisierender Kommunikationskonzepte in Verbindung mit klassischen, hierarchisch organisierten Topologien steht zunehmend im Fokus wissenschaftlicher Betrachtungen auf dem Gebiet hochdynamischer, drahtloser Netzwerkstrukturen. Ungeachtet dessen, dass ab einer gewissen Skalierungsstufe hierarchische Strukturen weiterhin unverzichtbar bleiben, eröffnen sich durch den partiellen Wegfall zentraler Verwaltungsinstanzen in den unteren Hierarchieebenen völlig neue Anwendungsgebiete. *Mobile Ad Hoc Netzwerke (MANET)* und deren Fähigkeit zur *spontanen Vernetzung* bieten dabei wesentliche Vorteile gegenüber komplett zentral organisierten Strukturen. Es ergibt sich eine gleichmäßigere Netzlastverteilung, woraus wiederum eine effiziente Nutzung der zur Verfügung stehenden Energieressourcen und ein robustes Verhalten gegenüber partiellen Ausfällen resultiert.

Aktuelle Forschungsarbeiten auf diesem Gebiet gehen dabei meist von einer homogenen Netzwerkstruktur auf Basis eines einheitlichen Funkstandards aus. Aufgrund von Inkompatibilitäten zwischen unterschiedlichen Technologien ist eine Kommunikation über die Grenzen des gewählten Funkstandards hinweg ausgeschlossen. In Folge dessen ergeben sich unter Umständen viele Einschränkungen in Hinsicht auf Sendereichweite, Energiebedarf oder die benötigte Datenrate.

Durch eine Integration unterschiedlicher Funktechnologien können die spezifischen Vorteile mehrerer Kommunikationsstandards genutzt werden. Basierend auf einer intelligenten Steuerung ist es nun möglich, je nach Anforderungen an den Kommunikationskanal einen jeweils optimalen Funkstandard zu wählen. Die sich daraus ergebende, heterogene Netzwerktopologie erhöht die Erreichbarkeit aller Knoten und kann deutlich flexibler genutzt werden.

Diese Arbeit untersucht geeignete Möglichkeiten einer solchen *funkstandardübergreifenden Kommunikation in mobilen Ad Hoc Netzwerken*. Rahmenbedingungen wie bspw. die Kompatibilität zu verfügbaren Technologien oder die modulare Erweiterbarkeit sollen dabei beachtet werden. Das vorgestellte Konzept umfasst drei wesentliche Problemstellungen:

- Verbindungsmanagement / Topologieaufbau
- Routing / Forwarding
- Protokollkonvertierung

Zur Evaluierung der entwickelten Lösungsansätze erfolgt die Entwicklung einer umfassenden Simulationsumgebung auf Basis von Java und MPI - *SimANet*. Durch die Analyse der Simulationsergebnisse ist es anschließend möglich, qualitative Aussagen in Hinblick auf eine mögliche Steigerung der Erreichbarkeit, Energieeinsparungen sowie Richtwerte bezüglich der Skalierbarkeit zu ermitteln.

Auf Grundlage der Simulationsergebnisse wurde die prototypische Umsetzung eines funkstandardübergreifenden Netzwerkknotens initiiert. Die Prototyp-Plattform *PLANet* zeigt die praktische Umsetzbarkeit des vorgestellten Konzeptes. Das System erlaubt zusätzlich auch eine Validierung der zuvor ermittelten Simulationsergebnisse.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	2
1.2. Zielsetzung dieser Arbeit	3
1.3. Aufbau dieser Arbeit	4
2. Grundlagen und Modell	7
2.1. Das ISO/OSI Referenzmodell	7
2.1.1. Bitübertragungsschicht	9
2.1.2. Sicherungsschicht	12
2.1.3. Vermittlungsschicht	14
2.1.4. Transportschicht	15
2.1.5. Kommunikationssteuerungs-/Darstellungs-/Anwendungsschicht	16
2.2. Funkstandards nach IEEE 802	16
2.2.1. Wireless PAN - IEEE 802.15	17
2.2.2. Wireless LAN - IEEE 802.11	22
2.2.3. Wireless MAN - IEEE 802.16	23
2.3. Selbstorganisation	23
2.4. Schnittstellensynthese	24
2.4.1. Interface Block (IFB)	26
2.4.2. Rekonfigurierung	28
2.5. Zusammenfassung	28
3. Stand der Forschung	29
3.1. Mobile Ad Hoc Netzwerke	29
3.1.1. Topologieoptimierung	32
3.1.2. Routing	34
3.1.3. Drahtlose Sensornetzwerke	44
3.2. Kommunikationstechnologien der nächsten Generation	46
3.2.1. Software Defined Radio	47
3.2.2. Cognitive Radio	50
3.2.3. Smart Antennas / Intelligent Antennas	53
3.2.4. Ultrabreitband	55
3.2.5. Ambient Networking	56
3.3. Zusammenfassung	59
4. Embedded Ambient Networking	61
4.1. Konzept	62
4.2. Einordnung in das ISO/OSI Modell	65

4.3.	Anforderungen	67
4.3.1.	Nutzerseitige Anforderungen	68
4.3.2.	Konzeptionelle Anforderungen	73
4.4.	Konzeptumsetzung	80
4.4.1.	IFB Protokollkonvertierung	80
4.4.2.	Extended Interface Block (xIFB)	83
4.4.3.	Link Controller Unit (LCU)	85
4.4.4.	Energy Balanced Cooperative Routing (EBCR)	88
4.5.	Ablauf des Verarbeitungsprozesses	93
4.5.1.	Protokollkonvertierungsprozess	94
4.5.2.	Verbindungsmanagement innerhalb der LCU	96
4.5.3.	EBCR-basierter Routingprozess	98
4.6.	Konzeptionelle Grenzen	101
4.6.1.	Leistungsfähigkeit EAN-Kern	101
4.6.2.	Komplexität und indirekte Störeinflüsse	102
4.6.3.	IP Adressierungsmodell und Routing	103
4.6.4.	Protokollkonvertierung und Paketlaufzeit	104
4.7.	Zusammenfassung	105
5.	Simulationsumgebung SimANet	107
5.1.	Simulationsumgebungen im Bereich mobiler Ad Hoc Netzwerke	108
5.2.	Theoretische Grundlagen	111
5.2.1.	Simulationsmodell	111
5.2.2.	Bewegungsmodelle	112
5.2.3.	Modellierung von Hindernissen	114
5.2.4.	Energiemodell	115
5.2.5.	Nachbarschaftsverwaltung	115
5.2.6.	Parallelisierung	122
5.3.	Funktionsumfang und Implementierung	127
5.3.1.	Architektur	128
5.3.2.	Simulationskern	129
5.3.3.	Skalierbarkeit und Parallelisierung	130
5.3.4.	Ein- und Ausgabe	131
5.4.	Einschränkungen	133
5.5.	Optimierungsansätze	134
5.5.1.	Übertragungsmodell	134
5.5.2.	Topologieoptimierung / Routing	135
5.5.3.	Sliding Time Window	135
5.5.4.	Parallelisierung	135
5.5.5.	Partitionierung	137
5.6.	Simulation und Analyse	137
5.6.1.	Leistungsfähigkeit des Simulationskern	137
5.6.2.	Testreihen EAN - Topologische Kenngrößen	140
5.6.3.	Testreihen EAN - Worst Case Pfadkosten	144

5.6.4. Testreihen EAN - Dynamisches Verhalten	150
5.7. Zusammenfassung	157
6. Prototyp-Plattform PLANet	159
6.1. Funktionsumfang und Einsatzbereiche	160
6.1.1. Einschränkungen	161
6.2. Aufbau / Design	162
6.2.1. Hardware	163
6.2.2. Funkmodulansteuerung / Verbindungsmanagement	167
6.2.3. Laufzeitsystem / Software	168
6.2.4. Erweiterbarkeit	172
6.3. Anwendungsszenarien	173
6.3.1. Testszenario I: Multi-Standard, Multi-Hop Kommunikation	173
6.3.2. Testszenario II: Stabilität	176
6.3.3. Testszenario III: Energiebedarf	178
6.4. Zusammenfassung und offene Punkte	182
7. Zusammenfassung und Ausblick	185
7.1. Ergebnisse dieser Arbeit	186
7.1.1. Konzept	186
7.1.2. SimANet	187
7.1.3. PLANet	188
7.2. Ausblick	188
A. Anhang 1 - SimANet Strukturbeschreibung	191
B. Anhang 2 - SimACon Skriptsprache	193
Eigene Veröffentlichungen	197
Studentische Arbeiten	199
Literaturverzeichnis	201
Nomenklatur	218
Thesen	219

Abbildungsverzeichnis

1.1.	Bsp. funkstandardübergreifende Kommunikation	1
1.2.	a) Aktuelle Funkstandards, b) Funkstandardintegration	2
2.1.	ISO/OSI und TCP/IP Referenzmodell	8
2.2.	Kommunikationskanal nach dem ISO/OSI Referenzmodell	9
2.3.	Multiplexverfahren	10
2.4.	Hidden- / Exposed Terminal Problem	12
2.5.	Multihop Kommunikation nach dem ISO/OSI Referenzmodell	14
2.6.	Einordnung der zentralen Prokollle des TCP/IP Stacks	15
2.7.	Bluetooth Verbindungsaufbau	18
2.8.	Bluetooth Protokollstack	19
2.9.	IEEE 802.15.4 / ZigBee Protokollstack	20
2.10.	IEEE 802.15.4 spezifizierte Topologien	21
2.11.	Kopplung zweier IPs	24
2.12.	IFB Makrostruktur / endlicher Automat	26
2.13.	Datenfluss innerhalb des IFB's	27
3.1.	Topologieoptimierung	32
3.2.	Schrittweise Topologieoptimierung	33
3.3.	Modell eines Kommunikationsnetzes	36
3.4.	Bewertung Routingalgorithmen	41
3.5.	Impllosion und Overlap Problem	43
3.6.	Sensorknoten	44
3.7.	WSN: Abhängigkeit von Abtastrate, Energiebedarf und Latenz	46
3.8.	a) OSI Referenzmodell, b) ideales SDR-System	48
3.9.	SDR-Plattform	49
3.10.	"Cognitive Radio"	51
3.11.	Schematische Darstellung eines <i>Cognitive Radio Szenarios</i>	52
3.12.	Funktionsweise adaptiver Antennensysteme.	53
3.13.	Adaptive Antennensysteme - Beispiele	54
3.14.	Kommunikationsstandards	55
3.15.	Multipath-Ausbreitungsszenario	56
3.16.	Beispielszenario eines Ambient Network	57
3.17.	schematischer Aufbau eines <i>Ambient Network</i>	58
4.1.	Einordnung des Konzeptes in die aktuellen Forschungsthemen	61
4.2.	Netzwerkknoten - schematische Darstellung	62
4.3.	Konzept Embedded Ambient Networking - praktische Umsetzung	63
4.4.	Integration unterschiedlicher Funkstandards	64
4.5.	EAN: Einordnung in das ISO/OSI Modell	66

4.6.	IP basierter EAN Verarbeitungsprozess	66
4.7.	EAN Funkstandardintegration	67
4.8.	Aufbau IPv4 Paket	69
4.9.	Aufbau IPv6 Paket	69
4.10.	EAN Beispielszenario	70
4.11.	Einordnung aktueller, drahtloser Kommunikationstechnologien	71
4.12.	Integration der Funkmodule: Hardwarespezifische Ansteuerung	73
4.13.	Aufbau Routing Information Protocol (RIP)	76
4.14.	Routingkonzept: Unterschiede	77
4.15.	Protokolloverhead durch den Konvertierungsprozess	79
4.16.	funktionaler Ablauf innerhalb eines IFBs	80
4.17.	IFB-basierte Funkstandardintegration	81
4.18.	IFB Anwendung auf drahtlose Kommunikationsschnittstellen	83
4.19.	IFB Funktionalität und xIFB Erweiterung	84
4.20.	Schema des xIFB Gesamtsystems	85
4.21.	Integration der LCU Einheit in das Gesamtsystem	86
4.22.	Routen Cache Container	87
4.23.	Link Management Cache	87
4.24.	IP basiertes Routing: allgemeiner Ablauf als UML Zustandsdiagramm	88
4.25.	Pseudocode-Darstellung EBCR	92
4.26.	Netzwerkknoten - Verarbeitungsprozess	94
4.27.	xIFB Konzepterweiterung	95
4.28.	Anpassungsprozesse während der Protokollkonvertierung	96
4.29.	Interner Verarbeitungsprozess der LCU	97
4.30.	EBCR - Beispiel Pfadfindung	98
4.31.	EBCR Worst Case Szenario	100
5.1.	SimANet	107
5.2.	Bearbeitungszyklus der Event-Queue	111
5.3.	Bewegungsmodelle I	112
5.4.	Bewegungsmodelle II	113
5.5.	Modellierung von Hindernissen	114
5.6.	Quadtree und dessen geometrische Repräsentation	116
5.7.	Quadtree und Compressed Quadtree im Vergleich	117
5.8.	Compressed Quadtree: Worst Case Szenario	118
5.9.	Skip Quadtree	119
5.10.	Unschärfe Bereichssuche	120
5.11.	Pseudocode-Darstellung Bereichsanfrage	121
5.12.	Pseudocode: Sliding Time Window - Teil 1	123
5.13.	Pseudocode: Sliding Time Window - Teil 2	124
5.14.	Parallele Simulation: Dynamische Grenzverschiebung	125
5.15.	Pseudocode: Partitionierung - Dynamische Lastverteilung	126
5.16.	SimANet Version 1 und 2 im Vergleich	127
5.17.	Model-View-Controller Entwurfsmuster	128

5.18. Unterschiedliche Ansätze für MPI Umsetzungen in Java	130
5.19. Funktionale Aufteilung der SimANet GUI	132
5.20. Optimierungspotential durch MPI2Java	136
5.21. Parallelisierung von SimANet - MPI Umsetzungen im Vergleich	136
5.22. SimANet - Messreihen I und II - Datenstrukturen	138
5.23. SimANet - Messreihen III und IV - Ressourcenverwaltung und Parallelisierung	139
5.24. Funkstandardintegration - Minimalbeispiel	140
5.25. Screenshot - Steigerung der Erreichbarkeit	141
5.26. SimANet - Messreihen V: Steigerung der Erreichbarkeit	143
5.27. SimANet - Messreihen VI und VII: Steigerung der Erreichbarkeit	143
5.28. SimANet - Messreihen VIII und IX: Steigerung der Erreichbarkeit	144
5.29. EAN Messungen - Optimierungsverlauf I	144
5.30. Uniforme und zufällige Knotenverteilung	147
5.31. SimANet - Messreihen X: Kommunikationskosten	148
5.32. SimANet - Messreihe XI: Kommunikationskosten	149
5.33. SimANet - Messreihe XII: Kommunikationskosten	149
5.34. SimANet - Messreihe XIII: Kommunikationskosten	149
5.35. EAN Messungen - Optimierungsverlauf II	150
5.36. Knotentopologie - <i>Hot Spot</i>	152
5.37. Simulationsergebnisse Routing: Beispieldiagramm	153
5.38. SimANet - Messreihe XIV und XV: Routing	155
5.39. SimANet - Messreihe XVI und XVII: Routing	156
5.40. SimANet - Messreihe XVIII und XIX: Routing	156
6.1. Platform for Ambient Networking	159
6.2. Technische Umsetzung von PLANet	160
6.3. Erster Prototyp	162
6.4. PLANet - Platinen-Layout	163
6.5. Komponentendarstellung von PLANet	164
6.6. Verfügbare Funkmodule für PLANet	165
6.7. Integration der Funkmodule: Angepasster Lösungsansatz	167
6.8. PLANet - Abstraktionsebenen des Systems	169
6.9. Vergleich verfügbarer Betriebssysteme für eingebettete Systeme	170
6.10. PLANet Testaufbau	173
6.11. PLANet - Messreihe I: Paketlaufzeit	174
6.12. PLANet - Messreihe II: Paketlaufzeit	175
6.13. PLANet - Messreihe III: Signallaufzeiten	175
6.14. PLANet - Messreihe IV: Paketsegmentierung	176
6.15. PLANet - Messreihe Va: Schwankungen Konvertierungszeiten	177
6.16. PLANet - Messreihe Vb: Schwankungen Konvertierungszeiten	177
6.17. PLANet - Messreihe VI: Detaildarstellung ausgewählter Datenraten	178
6.18. PLANet - Messreihe VIIa: Leistungsbedarf Betriebsmodi	179
6.19. PLANet - Messreihe VIIb: Leistungsbedarf Einzelkomponenten	181
A.1. SimANet - Strukturbeschreibung	191

B.1. SimACon Skript 195

Tabellenverzeichnis

2.1. Aufgabenverteilung im ISO/OSI-Referenzmodell	8
2.2. Übersicht zentraler Funkstandards	17
2.3. Bluetooth: Übersicht Klassen und Reichweiten	20
3.1. Übersicht Routingverfahren	40
4.1. Bsp. EBCR - Definition des Kostenvektors und Gewichtungsszenario	98
5.1. SimANet - Rahmenbedingungen I	142
5.2. Definition der Kostenvektoren	145
5.3. Gewichtung innerhalb der Kostenvektoren	146
5.4. SimANet - Rahmenbedingungen II	147
5.5. SimANet - Rahmenbedingungen III	153
6.1. PLANet - Messungen des Energiebedarfs	179
6.2. Vergleichsrelation SimANet und PLANet I	180
6.3. Vergleichsrelation SimANet und PLANet II	180

1. Einleitung

Der Bereich *MANETs*¹ unterteilt sich in eine Vielzahl einzelner Forschungsschwerpunkte in der Informations- und Kommunikationstechnik. Viele dieser Teilgebiete entwickelten sich aus den Problemstellungen der ersten Generation drahtgebundener Netzwerke. In Hinblick auf die neuen, drahtlosen Kommunikationstechnologien und den Wandel zur *Ad Hoc Vernetzung* können viele klassische Lösungen nicht mehr effizient genutzt werden. Ein Großteil der verfügbaren Netzwerkprotokolle muss dementsprechend überarbeitet und angepasst werden. Durch die gewonnene Mobilität und den Wegfall zentraler Verwaltungsstrukturen entstehen nun hochdynamische Netzwerktopologien. Grundlegende Forderungen nach einer robusten Netzinfrastruktur, welche die Erreichbarkeit trotz partieller Störungen oder Ausfälle sichert, verlangen völlig neue Konzepte. Dies resultiert vor allem auch aus der Tatsache, dass in *MANETs* per Definition jede *Funkinheit*, im weiteren Verlauf als *Knoten* bezeichnet, als eine mögliche Vermittlungsstelle fungieren kann, um ankommende Datenpakete zielgerichtet weiterzuleiten.

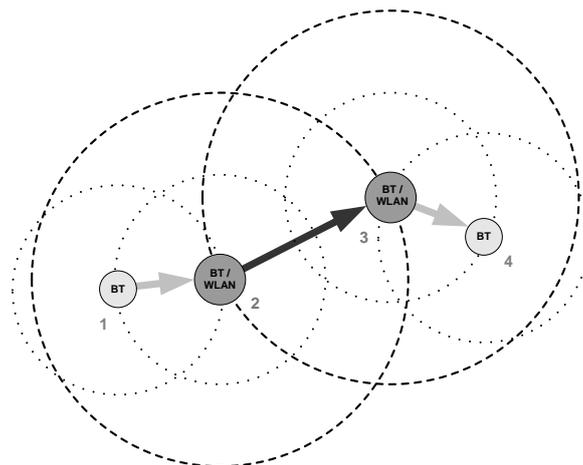
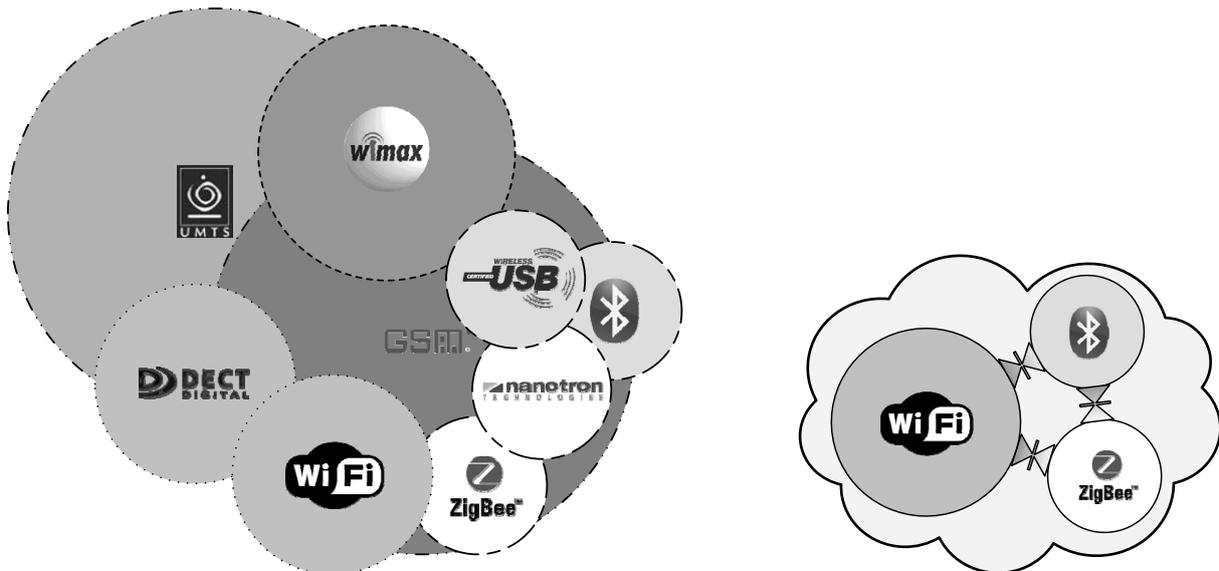


Abbildung 1.1.: Beispielhafte Darstellung einer funkstandardübergreifenden Kommunikation. Knoten 1 und 4 können aufgrund begrenzter Sendereichweite nicht auf direktem Wege miteinander kommunizieren. Die zusätzlichen Knoten 2 und 3 ermöglichen die Weiterleitung der Daten über einen alternativen Funkstandard mit höherer Sendereichweite.

Aufgrund der geringen Sendereichweite mobiler Endgeräte kann ein Großteil der Knoten nicht auf direktem Wege miteinander kommunizieren. Unter der Voraussetzung, dass gültige Pfade zwischen den Kommunikationspartnern existieren, muss versucht werden, die zu sendenden Informationen über einen möglichst effizienten Weg zum jeweiligen Empfänger weiterzuleiten (Abbildung 1.1). Während einer Kommunikation können existierende Kommunikationspfade aufgrund von Hindernissen oder begrenzter Energieressourcen jederzeit unterbrochen werden. Dies stellt hohe Anforderungen an die eingesetzten Algorithmen zur Wegfindung und Topologiekontrolle, da möglichst schnell auf Änderungen in der Umwelt reagiert werden muss.

¹Mobile Ad Hoc Netzwerke

1. Einleitung



(a) Aktueller Stand der Technik. Eine Vielzahl (teilweise konkurrierender) drahtloser Kommunikationsstandards auf Basis verschiedener Technologien und mit unterschiedlichen Optimierungsansätzen.

(b) Überwindung von Inkompatibilitäten - Die Integration unterschiedlicher Funkstandards in eine transparente Netzstruktur.

Abbildung 1.2.: a) Aktuelle Funkstandards, b) Funkstandardintegration

1.1. Motivation

MANETs nutzen für die Kommunikation das Medium Luft, basierend auf unterschiedlichen, elektromagnetischen Funktechnologien. Die verfügbaren Lösungen sind dabei teils hochkomplex und für die spezifischen Einsatzgebiete optimiert. Die spezifischen Übertragungseigenschaften sind somit nur für eine kleine Teilmenge aller denkbaren Anwendungsszenarien ideal. Soll der Kommunikationsstandard für andere Aufgaben genutzt werden, müssen unter Umständen Einschränkungen in Kauf genommen werden. Wesentliche Parameter sind hierbei die maximale Datenrate, mögliche Sendereichweite, durchschnittlicher Energiebedarf und die charakteristischen Eigenschaften des genutzten Frequenzbandes.

Resultierend aus den stetig steigenden Anforderungen der Anwender und dem Konkurrenzkampf großer Unternehmen werden zunehmend neue, anwendungsspezifische Kommunikationslösungen entwickelt. In Folge dessen existiert eine große Vielfalt aktuell verfügbarer, teils konkurrierender Funkstandards (Abbildung 1.2a). Aufgrund technologischer Einschränkungen besteht zwischen den einzelnen Technologien nahezu keine Kompatibilität und eine Kommunikation über die Grenzen des jeweiligen Funkstandards hinweg ist ausgeschlossen.

Die Gründe für eine Einführung immer neuer Standards sind dabei in vielen Fällen nicht global koordiniert. Für den Großteil aktueller Anwendungen erfüllen verfügbare Funkstandards nahezu alle Anforderungen. Eine mögliche Alternative zur Entwicklung immer neuer drahtloser Kommunikationsstandards ist dabei denkbar einfach. Durch die Integration unterschiedlicher Funktechnologien in eine heterogene Netzstruktur können funkstandardspezifische Vortei-

le mehrerer Kommunikationsstandards genutzt werden. Auf Basis einer intelligenten Kontrolleinheit ist es somit möglich, je nach Anforderungen an den Kommunikationskanal den jeweils optimalen Funkstandard zu wählen. Gleichzeitig reduzieren sich durch eine solche Kopplung sowohl die Entwicklungskosten als auch die *Time-to-Market*² erheblich.

Bisherige Forschungsansätze im Bereich mobiler Ad Hoc Netzwerke beschränken sich konzeptionell auf den Einsatz eines einheitlichen Funkstandards und gehen somit von einer technologisch homogenen Netzwerktopologie aus. Vorausschauend auf kommende Generationen drahtloser Kommunikationstechnologien müssen Lösungen für die Integration verfügbarer Funkstandards in eine heterogene Netzinfrastruktur gefunden werden. Auf dieser Basis entstehen deutlich effizientere und flexiblere Kommunikationsstrukturen. Ein Primäres Ziel ist somit die Überwindung von Inkompatibilitäten unterschiedlicher Funkstandards (Abbildung 1.2b). Neben einer deutlichen Steigerung der Erreichbarkeit ergibt sich daraus gleichzeitig eine effizientere Nutzung der verfügbaren Ressourcen.

1.2. Zielsetzung dieser Arbeit

Im Rahmen dieser Arbeit wird ein Konzept für die funkstandardübergreifende Kommunikation in mobilen Ad Hoc Netzwerken entwickelt. Basierend auf der hardwarenahen Kopplung unterschiedlicher Funkmodule und einer zentralen Kontrolleinheit kann eine Integration mehrerer homogener Kommunikationsstrukturen in eine heterogene Netztopologie erfolgen. Dabei stehen die folgenden Zielstellungen im Fokus der Entwicklungen:

- Steigerung der Erreichbarkeit aller Knoten
- Optimierung des topologieweiten Energiebedarfs
- Nutzung effizienter und adaptiver Kommunikationspfade
- Skalierbarkeit hinsichtlich einer zuverlässigen Nutzung auch in großen Topologien
- Kompatibilität bezüglich der Einbindung handelsüblicher Endgeräte mit standardisierten Funkschnittstellen

Für die Realisierung dieser Zielstellungen werden drei konzeptionelle Kernbereiche betrachtet:

Verbindungsmanagement / Topologieaufbau: Grundlage für jede Kommunikation ist eine hinreichend stabile und adaptive Netzwerktopologie, welche auf Änderungen in der Umwelt selbstständig reagiert. Um dies in hochdynamischen Ad Hoc Netzwerken gewährleisten zu können, kommen Optimierungsmethoden aus dem Bereich der Selbstorganisation zum Einsatz, welche die Komplexität des Systementwurfs in einem beherrschbaren Rahmen halten.

²TTM

1. Einleitung

Routing / Forwarding: Um Daten in MANETs effizient und ressourcenschonend zu übertragen, ist die Wahl geeigneter Routingverfahren von entscheidender Wichtigkeit. Die Problematik wird durch die Integration unterschiedlicher Kommunikationsstandards noch verstärkt. Ein Schwerpunkt dieser Arbeit ist daher die Analyse und Bewertung unterschiedlicher Routingstrategien. Auf dieser Grundlage wird ein Routingkonzept für die Anwendung in heterogenen Netzstrukturen mit hoher Dynamik entwickelt.

Protokollkonvertierung: Die Konvertierung von Datenpaketen unterschiedlicher Funkstandards ist ebenfalls eine zwingende Voraussetzung für die störungsfreie Kommunikation in heterogenen Netzstrukturen. Basierend auf dem Konzept der Schnittstellensynthese übernimmt ein dedizierter Hardwareblock diese Aufgabe.

Der vorgestellte Ansatz soll auf Basis einer geeigneten Simulationsumgebung evaluiert werden. Der zu entwickelnde Simulator muss dabei Multi-Standard, Multi-Interface Netzwerktopologien unterschiedlicher Größenordnungen berechnen können und die Möglichkeit für verteilte Simulationen auf dem Großrechner der TU Chemnitz bieten. Folglich stellt die Konzeption und Umsetzung geeigneter Datenstrukturen und Parallelisierungsverfahren einen weiteren Schwerpunkt dieser Arbeit dar.

Die darauf aufbauenden Untersuchungen befassen sich grundlegend mit der Energieeffizienz und entsprechenden Zusammenhängen hinsichtlich der Skalierbarkeit, da diese für den praktischen Einsatz wesentliche Kriterien darstellen. Ausgewählte Routingverfahren sowie Methoden zur Topologieoptimierung sollen in konkreten Anwendungsszenarien angewendet und miteinander verglichen werden. Aussagen in Bezug auf mögliche Steigerungen der Erreichbarkeit werden im Rahmen dieser Simulationsergebnisse konkretisiert und diskutiert.

Die gewonnenen Informationen werden auf Grundlage einer parallel entwickelten, prototypischen Testplattform validiert.

1.3. Aufbau dieser Arbeit

Zunächst werden im 2. Kapitel Grundlagen vermittelt, die zum besseren Verständnis dieser Arbeit beitragen sollen. Konkret betrifft dies den Bereich Netzwerkkommunikation, in welchem neben dem *ISO/OSI Referenzmodell* unterschiedliche *drahtlose Kommunikationsstandards* vorgestellt werden. Weitere Themenkomplexe befassen sich mit der *Selbstorganisation* und der *Schnittstellensynthese*.

Im folgenden 3. Kapitel richtet sich der Fokus auf aktuelle Forschungsarbeiten im Bereich mobiler Ad Hoc und Sensornetzwerke. Zentrale Themen befassen sich hierbei mit *Routingverfahren* und *Algorithmen zur Topologieoptimierung*, da diese mit wesentlichen Problemstellungen dieser Arbeit korrelieren. Auf dem Gebiet *drahtloser Sensornetzwerke* werden kooperative Verfahren für den energieeffizienten Betrieb sowie Lösungsansätze auf dem Gebiet der *Zeitsynchronisation* und der *Lokalisierung* vorgestellt. Der zweite Teil des Kapitels befasst sich mit *drahtlosen Kommunikationstechnologien der nächsten Generation* und den thematischen

Zusammenhängen mit dieser Arbeit. Dabei werden speziell die Bereiche *Software Defined Radio*, *Cognitive Radio* sowie *Ultrabreitband* und *Ambient Networking* angesprochen. Der Fokus liegt dabei auf den konzeptionellen Überschneidungen mit den Zielstellungen dieser Arbeit. Lösungsansätze auf dem Gebiet intelligenter Antennensysteme (*Smart Antennas*) zur optimierten Nutzung des Übertragungsmediums runden das Kapitel ab.

Anschließend folgt in Kapitel 4 die Einführung eines Konzeptes für die funkstandardübergreifende Kommunikation in mobilen Ad Hoc Netzwerken. Die drei wesentlichen Problemstellungen werden an dieser Stelle konkretisiert und Anforderungen sowohl auf Hardware- als auch auf Softwareseite analysiert. Zu den jeweiligen Themengebieten *Verbindungsmanagement / Topologieaufbau*, *Routing / Forwarding* sowie *Protokollkonvertierung* wird ein entsprechender Lösungsansatz präsentiert und anschließend bewertet. Das Kapitel stellt dabei alle konzeptionellen Kernkomponenten detailliert vor und verdeutlicht deren funktionale Zusammenhänge.

Kapitel 5 evaluiert im Anschluss das vorgestellte Konzept anhand der Simulationsumgebung *SimANet*³ und präsentiert die Ergebnisse der Testreihen. Der Simulator deckt dabei durch mehrere Modelle für Bewegung, Energiebedarf, Kommunikation, Routing und Hindernisse alle aufgeführten Problemstellungen ab und ermöglicht die Durchführung und Analyse vielfältiger Anwendungsszenarien. Neben Einzelheiten zu Aufbau und Implementierung wird außerdem die Möglichkeit zur Parallelisierung des Simulators auf ein Cluster-System, wie bspw. den *CHiC*⁴, erläutert. Es folgen Einschränkungen der Simulationsplattform in Hinsicht auf den Funktionsumfang und die Skalierbarkeit. Vervollständigt wird das Kapitel durch Vergleiche mit anderen verfügbaren Simulationsumgebungen im Bereich mobiler Ad Hoc und Sensornetzwerke.

Um die weitreichenden Simulationsergebnisse zu verifizieren, wurde im Rahmen dieser Arbeit ein Prototyp auf Basis des vorgestellten Konzeptes realisiert. Kapitel 6 stellt den Prototyp *PLANet*⁵ vor und erläutert den Aufbau sowie Funktionsumfang. Dies beinhaltet auch Themen wie die modulare Erweiterbarkeit oder primäre Einsatzgebiete des Prototypen. Abschließend werden erste praktische Versuche ausgewertet und mit den Ergebnissen der Simulation verglichen.

Kapitel 7 fasst die Ergebnisse dieser Arbeit noch einmal abschließend zusammen und gibt einen Ausblick auf mögliche Erweiterungen und weiterführende Arbeiten in diesem Forschungsthema.

³Simulation Platform for Ambient Networking

⁴Chemnitz High Performance Linux Cluster

⁵Platform for Ambient Networking

2. Grundlagen und Modell

Das folgende Kapitel stellt grundlegende Themenkomplexe aus dem Bereich der Kommunikation in Rechnernetzen und verteilten Systemen vor. Dabei wird ein *verteilt System* im Rahmen dieser Arbeit wie folgt nach [79] definiert:

”Ein verteiltes System ist ein System, in dem sich Hardware- und Softwarekomponenten auf vernetzten Computern befinden und miteinander über den Austausch von Nachrichten kommunizieren.”

Die hier ausgewählten Themen sollen das Verständnis für diese Arbeit erleichtern. Im Vordergrund stehen dabei Methoden und Verfahrensweisen zur drahtlosen Kommunikation in komplexen Systemen auf Basis unterschiedlichster Technologien. Neben dem ISO/OSI Referenzmodell werden im weiteren Verlauf des Kapitels einzelne, standardisierte Funkstandards sowie Methoden der Selbstorganisation und der Schnittstellensynthese näher erläutert.

2.1. Das ISO/OSI Referenzmodell

Das *ISO/OSI Schichtenmodell* bildet die Designgrundlage für eine Vielzahl moderner Kommunikationstechnologien. Es wurde von der *ISO*¹ zertifiziert und steht für *Open Systems Interconnection*. Es kapselt die unterschiedlichen Aufgaben einer Kommunikation in sieben hierarchisch organisierten Schichten (Abbildung 2.1). Innerhalb dieser Schichten werden die jeweiligen Aufgaben durch entsprechende Kommunikationsprotokolle umgesetzt. Ein *Protokoll* wird im Rahmen dieser Arbeit wie folgt definiert:

Protokoll eine Vereinbarung, nach der die Verbindung, Kommunikation und Datenübertragung zwischen zwei Parteien abläuft. In seiner einfachsten Form kann ein Protokoll definiert werden als die Regeln, die Syntax, Semantik und Synchronisation der Kommunikation bestimmen.

”Ein Protokoll (→ Kommunikation / Datenübertragung) entspricht einer Menge von Regeln für den direkten oder indirekten Informationsaustausch zwischen einem Sender und einem Empfänger innerhalb derselben Schicht. Die Regeln definieren dabei Syntax, Semantik und Synchronisation der Kommunikation.”

Ein Protokoll legt somit fest, zu welchem Zeitpunkt oder in welcher Reihenfolge ein Knoten im Netzwerk eine Operation auszuführen hat.

¹Internationalen Standardisierungs Organisation

2. Grundlagen und Modell

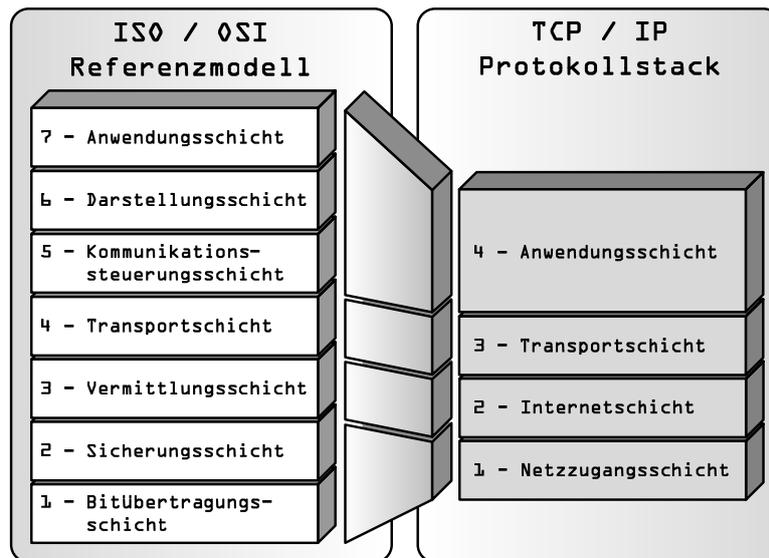


Abbildung 2.1.: Links das ISO/OSI Referenzmodell und rechts die Abbildung auf den vereinfachten Aufbau des TCP/IP Protokollstacks.

Hauptaufgabe jeder Schicht ist es, der darüberliegenden Schicht Dienste anzubieten. Diese Dienste setzen sich aus Funktionalitäten der eigenen Schicht sowie aus der Summe der Funktionalitäten aller darunterliegenden Schichten zusammen. Durch die Definition spezifischer Schnittstellen zwischen den einzelnen Schichten werden Fehlerquellen innerhalb eines hierarchischen Schichtenmodells minimiert. Zusätzlich gewährleistet die konsequente Verteilung der Aufgaben ein hohes Maß an Flexibilität und Modularität bei der Entwicklung.

Die Verteilung der Aufgaben auf die einzelnen Ebenen kann wie folgt aufgegliedert werden:

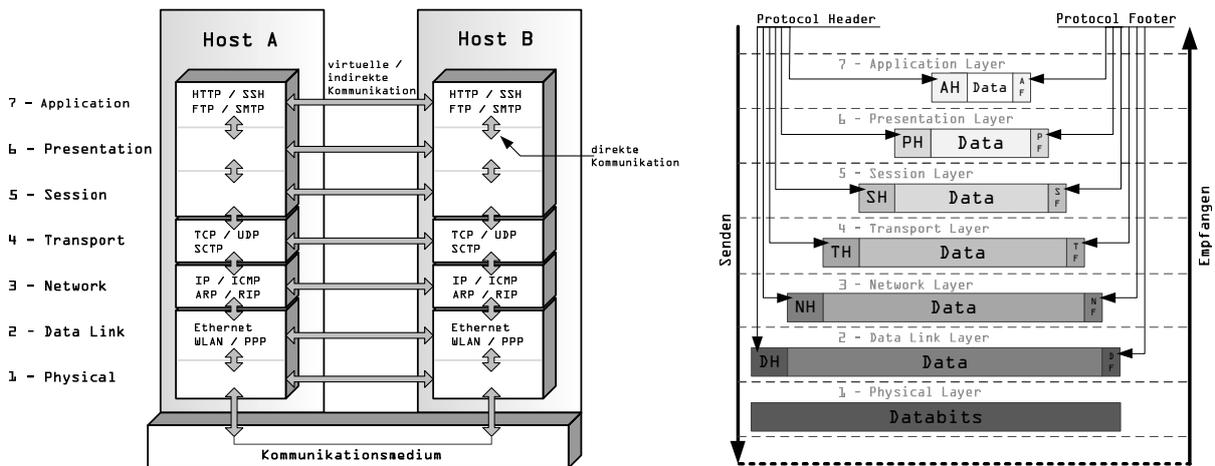
ISO/OSI Schicht	Funktion	Zuordnung	
7	Anwendungsschicht	Anwendungsspezifische Protokolle	
6	Darstellungsschicht	Datenformatierung und -darstellung	Software oder Betriebssystem
5	Sitzungsschicht	Dialogsteuerung	
4	Transportschicht	Ende-zu-Ende-Datenaustausch	
3	Vermittlungsschicht	Ende-zu-Ende-Vermittlung	
2	Sicherungsschicht	Punkt-zu-Punkt Verbindung zwischen direkt vernetzten Systemen	Firm- oder Hardware
1	Physikalische Schicht	Physikalische Bitübertragung	

Tabelle 2.1.: Übersicht der Aufgabenverteilung im ISO/OSI-Referenzmodell

Ein Datenaustausch zwischen zwei Stationen findet physisch nur auf der untersten Ebene des ISO/OSI Modells über ein geeignetes Medium statt. Da alle höher gelegenen Schichten auf den Diensten der tiefer gelegenen Ebenen aufsetzen, existiert auf jeder Ebene eine abstrakte, virtuelle Verbindung mit dem jeweiligen Kommunikationspartner. Abbildung 2.2(a) verdeutlicht diesen Vorgang. Einhergehend mit der Funktionskapselung in den einzelnen Ebenen werden die

zu übertragenden Informationen (*Nutzdaten/Payload*) durch die Protokolle mit zusätzlichen Informationen versehen und in entsprechende Datenpakete verpackt. Durch diese *Protokollheader* (Abbildung 2.2(b)) entsteht ein Mehraufwand bei der Kommunikation in Form eines erhöhten Datenvolumens, welcher als *Protokoll-Overhead* bezeichnet wird.

Der *TCP/IP Protokollstack* ist eine der bekanntesten Vertreter auf Grundlage des ISO/OSI Referenzmodells und wird in nahezu allen aktuellen Technologiebereichen angewendet. Abbildung 2.1(b) verdeutlicht den Bezug des Protokollstacks zum ISO/OSI Modell. Der vereinfachte TCP/IP Stack verfügt über nur vier Schichten und kapselt die drei obersten und die beiden untersten ISO/OSI Ebenen zu jeweils einer verallgemeinerten Protokollschicht.



(a) Tatsächliche und virtuelle Kommunikationspfade einer Verbindung zwischen Host A und B nach dem ISO/OSI Referenzmodell.

(b) Verpacken der Nutzdaten (Payload) in die Protokollheader der jeweiligen ISO/OSI-Ebenen.

Abbildung 2.2.: Schematische Darstellung einer Punkt-zu-Punkt Kommunikation nach dem ISO/OSI Schichtenmodell.

In Hinblick auf das Thema dieser Arbeit erscheint der Aufbau und die Funktionsweise des ISO/OSI Referenzmodells und des daraus abgeleiteten TCP/IP Protokollstacks wesentlich. Aus diesem Grund wird auf die einzelnen Ebenen des ISO/OSI Modells näher eingegangen.

2.1.1. Bitübertragungsschicht

Die unterste Ebene des ISO/OSI Referenzmodells repräsentiert die Bitübertragungsschicht (*Physical Layer / Layer 1*), welche Informationen über ein physikalisches Medium transportiert. Der Begriff *Medium* lässt sich hierbei als ein physikalisches Phänomen beschreiben, welches für den Transport von Informationen genutzt werden kann. Phänomene, die sich örtlich entfernt ausdrücken und praktisch nutzbar sind, können wie folgt aufgeschlüsselt werden:

2. Grundlagen und Modell

- Metallische Leiter auf Basis elektrischen Ladungsphänomenen
- Glas bzw. Glasfasern auf Basis von Licht
- Äther (bspw. Luft) auf Basis elektromagnetischer Wellen oder Schall

Der Begriff *Medium* umfasst folglich sowohl drahtgebundene als auch drahtlose Technologien zur Informationsübermittlung.

In Bezug auf das ISO/OSI Modell definiert die Bitübertragungsschicht mechanische, elektrische und funktionale Spezifikationen, um eine physikalische Verbindung aufzubauen, aufrecht zu erhalten und abzubauen. Die Spezifikationen [80, 160] umfassen dabei:

- *mechanische Eigenschaften*: Dimensionierung von Steckern, Pinbelegungen
- *elektrische Eigenschaften*: Signalpegel und Frequenzen
- *funktionale Eigenschaften*: Übertragungsverfahren

Die Definition entspricht damit der Beschreibung einer Schnittstelle, auf deren Bedeutung im Abschnitt 2.4 zum Thema *Schnittstellensynthese* näher eingegangen wird.

Für die drahtlose Übertragung auf dem Basisband können drei wesentliche *Modulationsverfahren* aus dem Bereich der Nachrichtentechnik zum Einsatz kommen. Der Datenstrom kann mittels einer Amplituden-, Frequenz- oder Phasenmodulation über das Medium übertragen werden [67]. Dabei sind die spezifischen Eigenschaften des Übertragungsmediums ausschlaggebend für die Zuverlässigkeit des Kommunikationssystems.

Um die Übertragung mehrerer paralleler Datenströme auf einem Medium zu gewährleisten, kommen unterschiedliche Übertragungsverfahren zum Einsatz. Einige Vertreter sogenannter *Multiplexverfahren* werden nun kurz vorgestellt.

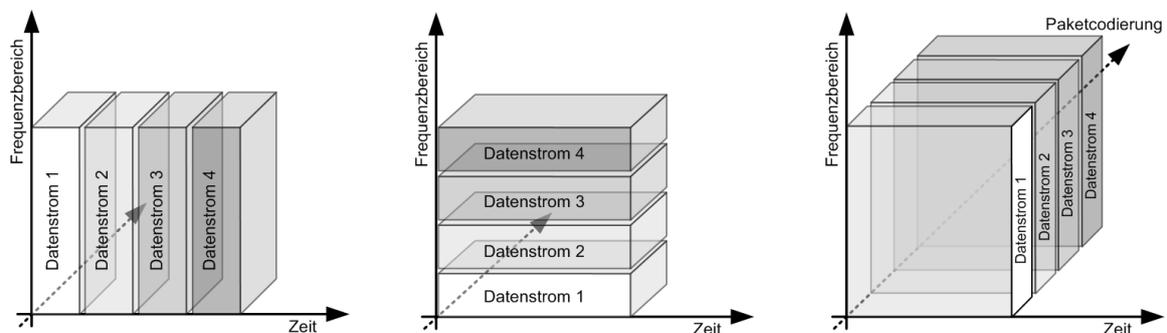


Abbildung 2.3.: Funktionsweise unterschiedlicher Multiplexverfahren. Zeitmultiplex- (links), Frequenzmultiplex- (mitte) und Codemultiplexverfahren (rechts).

Zeitmultiplexverfahren - TDMA²:

Die einzelnen Datenströme der jeweiligen Sender werden hier zeitlich versetzt in sogenannten *Time Slots* (Zeitscheiben/Zeitfenster) auf einem Kanal übertragen (Abbildung 2.3-links). Diese Zeitfenster können sowohl dynamisch vergeben oder nach einem vordefinierten Muster synchronisiert zugeteilt werden. Voraussetzung hierfür ist eine einheitliche, synchronisierte Zeitbasis der jeweiligen Sender und Empfänger.

Codemultiplexverfahren - CDMA³:

Bei Codemultiplexverfahren nutzen alle Nachrichten den gleichen Übertragungskanal und werden anschließend empfängerseitig anhand ihrer Codierung zugeordnet. Im Gegensatz zu TDMA entfällt bei CDMA die Aufteilung und Verwaltung einzelner Zeitfenster. In Abbildung 2.3-rechts wird dieses Verfahren verdeutlicht.

Raummultiplexverfahren - SDMA⁴:

Hierbei handelt es sich um ein sehr einfaches Multiplexverfahren, bei dem mehrere Übertragungssysteme parallele zur Verfügung stehen, welche somit mehrere zeitgleiche Übertragungen unterschiedlicher Sender ermöglichen. Im drahtgebundenen Fall können dies bspw. mehrere Leitungspaare sein. Äquivalent dazu repräsentieren bei der drahtlosen Kommunikation unterschiedliche Richtfunkstrecken ein solches Raummultiplexverfahren.

Frequenzmultiplexverfahren - FDMA⁵:

FDMA Verfahren nutzen unterschiedliche Frequenzbänder innerhalb des verfügbaren Frequenzbereichs um mehrere, parallele Datenströme übertragen zu können (Abbildung 2.3-mitte). FDMA kommt sowohl bei der drahtgebundenen als auch bei der drahtlosen Datenübertragung zum Einsatz.

Unter dem Namen *OFDM*⁶ wurden das klassische Frequenzmultiplexverfahren weiterentwickelt. Ähnlich wie bei FDMA werden auch hier verschiedene Datenströme auf mehrere schmale Frequenzbänder moduliert. Die Besonderheit von OFDM Verfahren besteht in der speziellen Aufteilung der Signale auf die Frequenzbänder. Die jeweiligen Trägerfrequenzen stehen zueinander orthogonal, wodurch Interferenzen auf dem Medium minimiert werden können. OFDM kommt in vielen modernen Übertragungstechnologien wie bspw. WLAN (IEEE 802.11a/g), WiMAX, *DSL*⁷ oder *DVB-T*⁸ zum Einsatz.

²Time Division Multiple Access

³Code Division Multiple Access

⁴Space Division Multiple Access

⁵Frequency Division Multiple Access

⁶Orthogonal Frequency Division Multiplex

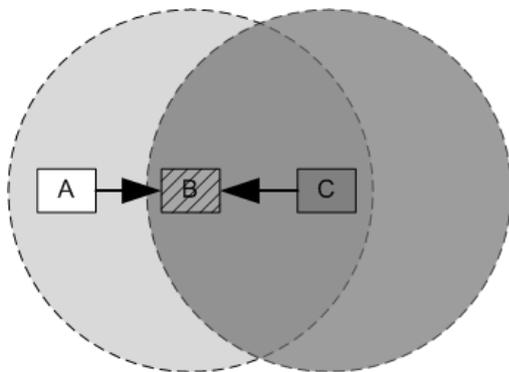
⁷Digital Subscriber Line

⁸Digital Video Broadcasting-Terrestrial

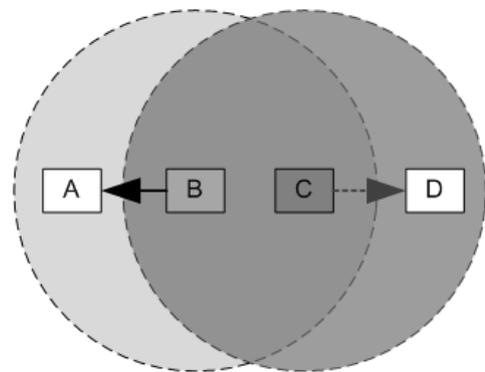
2.1.2. Sicherungsschicht

Auf der zweiten Ebene des ISO/OSI Referenzmodells detektieren und korrigieren Mechanismen der *Sicherungsschicht (Data Link Layer / Layer 2)* Übertragungsfehler. Nur durch eine solche Fehlerkorrektur kann eine stabile, fehlertolerante Datenübertragung gewährleistet werden. Spezielle Prüfsummen, wie bspw. *CRC*⁹, oder einfache Paritätsbits ermöglichen dabei die Erkennung fehlerhafter Pakete. Eine Flusssteuerung regelt zusätzlich je nach aktueller Systemlast dynamisch die Datenrate während der Übertragung.

Die Sicherungsschicht teilt sich in zwei Teilschichten auf. Die *LLC*¹⁰ übernimmt die zuvor genannten Funktionalitäten und ist somit für die Datensicherung auf Verbindungsebene verantwortlich.



(a) *Hidden Terminal Problem*: Knoten B liegt in der Sendereichweite von A; C jedoch nicht. Somit erhält C keine Informationen über die laufende Kommunikation zwischen A und B. Unter der Annahme, dass das Kommunikationsmedium nicht belegt ist, beginnt C mit dem Senden an B. In Knoten B tritt daraufhin eine Kollision auf und sowohl die Nachrichten von A als auch von C gehen verloren. A ist somit für C nicht sichtbar (*hidden*).



(b) *Exposed Terminal Problem*: Bei der dargestellten Ausgangssituation könnte während der Kommunikation zwischen B und A eine parallele Übertragung zwischen C und D erfolgen, da empfängerseitig keine Datenkollisionen erfolgen würden. Durch die Trägerprüfung (*carrier sense*) von C wird eine Übertragung jedoch verhindert, da ein belegtes Kommunikationsmedium durch Knoten B detektiert wird.

Abbildung 2.4.: Hidden- / Exposed Terminal Problem.

Nutzen mehrere Kommunikationspartner das gleiche Übertragungsmedium, muss diese Resource durch einzelnen Knoten exklusiv reservierbar sein. Ohne ein entsprechendes Zugangsverfahren, auch als *Arbitrierung* bezeichnet, könnten mehrere Knoten gleichzeitig mit der Übertragung von Daten starten, wodurch Kollisionen auf dem Medium entstehen (*Interferenzen*). Um die Überlagerung von Signalen und den daraus folgenden Informationsverlust zu vermeiden, kommt die *Medienzugangssteuerung (MAC)*¹¹ zum Einsatz. Diese kann auf der Basis von drei wesentlichen Prinzipien den fehlerfreien Datenfluss gewährleisten:

⁹Cyclic Redundancy Check - Zyklische Redundanzprüfung

¹⁰Logical Link Control

¹¹Media Access Control

Kollisionsentdeckung (Collision Detection)

Hier sind Kollisionen auf dem Medium jederzeit möglich. Sie werden erkannt und die Übertragung wird anschließend wiederholt. Das *CSMA/CD*¹² Verfahren [102] stellt hier einen wichtigen Vertreter dar, welcher bspw. bei Ethernet zum Einsatz kommt. Vor jeder Übertragung prüft die sendebereite Station zunächst das Übertragungsmedium ab (*Trägerprüfung - carrier sense*). Detektiert die Prüfung einen freien Kommunikationskanal, wird sofort mit der Übertragung begonnen. Im Falle eines belegten Mediums wird der Sendevorgang zunächst abgebrochen und zurückgestellt. Nachdem der Sender eine solche Kollision erkannt hat, wartet er eine zufällig gewählte Dauer innerhalb eines fest definierten Zeitintervalls. Anschließend erfolgt ein erneuter Sendeversuch mit initialer Trägerprüfung. Im ungünstigsten Fall wird bei mehreren aufeinander folgenden Sendeversuchen eine Kollision detektiert. Verfahren wie *ALOHA* [26] und *R-ALOHA* [101] basieren ebenfalls auf der Kollisionsentdeckung. Im Gegensatz zu *CSMA/CD* wird hier auf eine Prüfung des Mediums vor dem Senden verzichtet und der Übertragungsvorgang beginnt somit ohne Zeitverzögerung. Neben dem reinen *ALOHA* existiert noch ein *Slotted ALOHA*-Verfahren, bei dem die einzelnen Stationen nur zu definierten Zeiten (*Slots*) senden dürfen. Dazu ist allerdings ein synchrones Zeitsignal in allen Stationen notwendig, was im Bereich mobiler Ad Hoc Netzwerke zu neuen Problemen führt.

Kollisionsvermeidung (Collision Avoidance)

Kollisionen sind hier ebenfalls möglich, werden aber durch heuristische Verfahren minimiert. Das *CSMA/CA*¹³ Verfahren kommt bei *ISDN*¹⁴ oder auch bei IEEE 802.11 zum Einsatz. Durch eine zusätzlich eingeführte, zufällige Wartezeit nach der Trägerprüfung wird die Wahrscheinlichkeit für eine Datenkollision deutlich verringert.

Trotz der Einführung einer Medienzugriffssteuerung treten speziell bei der drahtlosen Kommunikation unterschiedliche Probleme auf. Zwei wesentliche Phänomene sind hierbei das *Hidden Terminal* und das *Exposed Terminal* Problem, welche in Abbildung 2.4 vereinfacht dargestellt sind. Zusätzliche Mechanismen wie das *RTS/CTS Verfahren*¹⁵ verbessern die Medienzugriffsverfahren nochmals und sorgen für die Minimierung von Kollisionen auf dem Übertragungsmedium. Probleme wie das *Hidden Terminal* können hierdurch vermieden werden.

Kollisionsauflösung (Collision Resolution)

Bei der Kollisionsauflösung werden Konkurrenzsituationen mittels *Prioritätsanalyse* oder einer festgelegten Reihenfolge geregelt. Umgesetzt wird ein solches Verfahren bspw. durch die Bit-Arbitrierung im *CAN-Bus*¹⁶.

Neben den genannten Verfahren existieren noch zahlreiche spezifische Lösungsansätze im Bereich drahtloser Sensornetzwerke oder *Topologieoptimierung* in mobilen Ad Hoc Netzwerken, auf die im späteren Verlauf dieser Arbeit noch genauer eingegangen wird.

¹²Carrier Sense Multiple Access / Collision Detection

¹³Carrier Sense Multiple Access / Collision Avoidance

¹⁴Integrated Services Digital Network

¹⁵Ready To Send / Clear To Send

¹⁶Controller Area Network

2. Grundlagen und Modell

Bezugnehmend auf den vereinfachten TCP/IP Protokollstack vereinen sich die Funktionalitäten der *Sicherungsschicht* und der *Bitübertragungsschicht* in der *Netzzugangsschicht* (auch *Host-to-Network Layer* genannt), welche die unterste Ebene in diesem Modell bildet.

2.1.3. Vermittlungsschicht

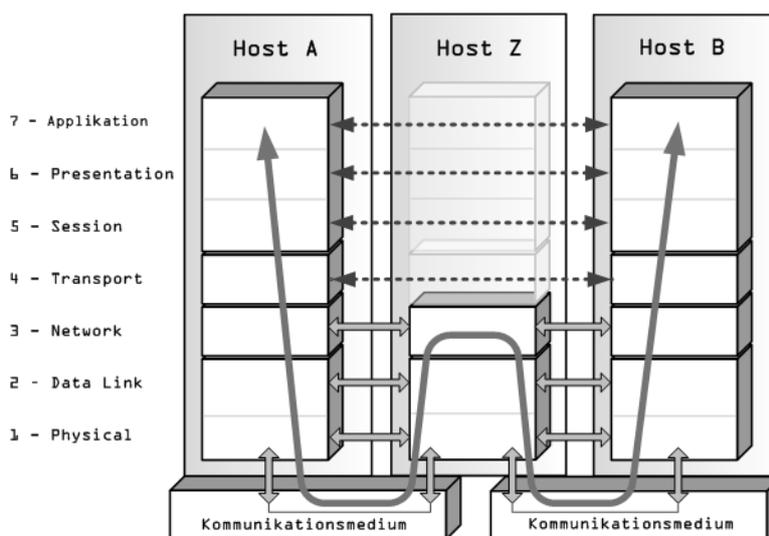


Abbildung 2.5.: Schematische Darstellung der Kommunikationspfade zwischen Host A und B über Host Z (*Multihop-Szenario*). Dabei erfolgt die Entscheidung für die Wegewahl auf Vermittlungsebene (Layer 3).

Besonders interessant für das im Rahmen dieser Arbeit vorgestellte Konzept ist die *Vermittlungsschicht* (*Network Layer / Layer 3*). Die primäre Aufgabe dieser Schicht besteht in der Suche nach möglichst optimalen Kommunikationspfaden vom Sender zum Empfänger und damit einhergehend die evtl. Weiterleitung der Daten über Zwischenknoten. In Bezug auf die Abstraktion vom ISO/OSI-Referenzmodells zum TCP/IP Protokollstack ist auf dieser Ebene *IP*¹⁷ [47, 137, 178] einzuordnen. Neben IP beinhaltet diese *Internetschicht* auch den Austausch von Statusinformationen und Fehlermeldungen mittels IP (*ICMP*¹⁸ [138]) sowie die Auflösung/Zuordnung von IP-Adressen zu eindeutigen Hardwareadresse der Netzwerkknoten mittels *ARP*¹⁹ [135].

Im Fokus weiterer Betrachtungen stehen auf dieser Ebene jedoch IP-basierte Protokolle für die Wegewahl (*Routing*). Die Weiterleitung von Datenpaketen auf IP-Ebene über einen Zwischenknoten wird in Abbildung 2.5 verdeutlicht. Da sich der Abschnitt 3.1.2 ausführlich mit dem Thema Routing/Forwarding im Bereich mobiler Ad Hoc Netzwerke befasst, wird an dieser Stelle auf weitere Details verzichtet.

¹⁷Internet Protocol

¹⁸Internet Control Message Protocol

¹⁹Address Resolution Protocol

2.1.4. Transportschicht

Auf dieser Ebene wird erstmals ein Übertragungskanal zwischen Sender und Empfänger über einen gewählten Multihop-Pfad durch das Netzwerk zur Verfügung gestellt. Zusätzlich werden die Datenpakete auf dieser Ebene segmentiert. Mit Hilfe einer *Congestion Avoidance* sowie einer Flusssteuerung auf der Basis eines *Sliding Window Verfahrens* wird die Kontrolle des Datenflusses im Netzwerk ermöglicht. Überlastsituationen durch zu viele Datenpakete in den Warteschlangen einzelner Knoten können verhindert werden. Anwendungen der darüberliegenden Schichten nutzen die durch die Transportschicht zur Verfügung gestellten, abstrakten Kommunikationskanäle und benötigen somit keine Informationen über die darunterliegenden Kommunikationsnetze. Im TCP/IP Protokoll Stack ist hier das *TCP*²⁰ [139, 174, 175] als ein *verbindungsorientiertes Transportprotokoll* angesiedelt, welches auf IP aufsetzt und virtuelle Kanäle zwischen zwei Netzwerkknoten zur Verfügung stellt. Diese Verbindungskanäle können dann auf Anwendungsebene in Form von sogenannten *Sockets* genutzt werden. Ein Großteil aktueller Netzwerkdienste der Anwendungsschicht verwenden TCP aufgrund mehrerer wesentlicher Vorteile. Zum einen werden aufgetretene Datenverluste automatisch detektiert und die fehlenden Segmente erneut angefordert, was eine vollständige und fehlerfreie Übertragung der Informationen garantiert. Zum anderen wird durch TCP eine bidirektionale Kommunikation ermöglicht. Mechanismen zur Vermeidung von Überlastsituationen erhöhen die Stabilität und die Zuverlässigkeit zusätzlich. Abbildung 2.6 verdeutlicht die Einordnung wesentlicher Protokolle in die vorgestellten Ebenen.

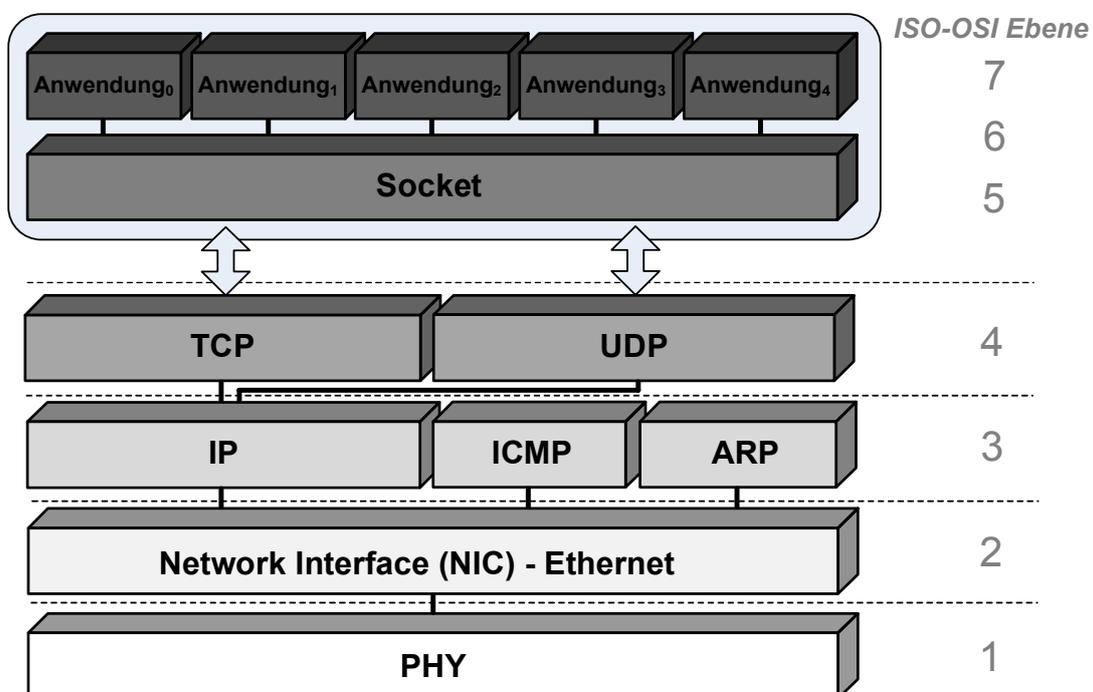


Abbildung 2.6.: Einordnung der zentralen Protokolle des TCP/IP Stacks in das ISO/OSI Referenzmodell.

²⁰Transmission Control Protocol

2. Grundlagen und Modell

Alternativ zu TCP kommt *UDP*²¹ [136] als *verbindungsloses Transportprotokoll* zum Einsatz. Speziell multimediale und zeitkritische Kommunikationsdienste (*VoIP*²², IPTV) nutzen die Vorteile von UDP. Da im Gegensatz zu TCP bei UDP die einzelnen Datenpakete vom Empfänger nicht bestätigt werden müssen, treten nur geringe Schwankungen der Verzögerungszeiten während der Kommunikation auf. Allerdings kann somit auch nicht garantiert werden, dass jedes Datenpaket tatsächlich den Empfänger erreicht. In Bezug auf die Übertragung von Audio- und/oder Video-Streams ist der Verlust einzelner Pakete allerdings weniger kritisch, was diesen Nachteil relativiert.

2.1.5. Kommunikationssteuerungs-/Darstellungs-/Anwendungsschicht

Die Aufgaben der drei obersten Schichten im ISO/OSI Modell werden auf die *Anwendungsschicht* (*Application Layer / Layer 4*) des TCP/IP Protokollstacks abgebildet. Diese vierte Ebene vereint die Kommunikationssteuerungs-, Darstellungs- und Anwendungsschicht nach dem ISO/OSI Referenzmodell. Die Aufgaben werden wie folgt gegliedert:

Die *Kommunikationssteuerungsschicht*, auch Sitzungsschicht genannt, koordiniert die logischen Verbindungen mehrerer, miteinander kommunizierender Anwendungsprozesse. Bei der Kommunikation zweier Knoten synchronisiert diese den Datenaustausch und stellt für eventuelle Unterbrechungen während der Datenübertragung Wiedereinstiegspunkte zur Verfügung.

Die *Darstellungsschicht* ist für die Codierung aller übermittelten Daten verantwortlich. Durch die beteiligten Kommunikationspartner müssen sowohl Dateiformat als auch eine etwaige Verschlüsselung und/oder Kompression der Daten interpretierbar sein.

Die *Anwendungsschicht* stellt nun für beide vorgestellten Modelle die oberste hierarchische Abstraktionsebene dar und bietet Softwareanwendungen Möglichkeiten zur Kommunikation mit anderen, im Netzwerk befindlichem Systemen. Die Fülle verfügbarer Dienste reicht von relativ einfach aufgebauten Protokollen wie *FTP*²³ über Funktionalitäten für *E-Mail* (*IMAP*²⁴, *SMTP*²⁵) bis hin zu Protokollen für die Übermittlung sicherheitsrelevanter Daten *SSH*²⁶.

2.2. Funkstandards nach IEEE 802

Die heutige Gesellschaft wird von einem hohen Grad an Mobilität, Flexibilität und Konnektivität geprägt. Dieser Fokus spiegelt sich auch in den technologischen Entwicklungen wieder. In Folge der stetig sinkenden Preise und einfachen Handhabung nimmt die Anzahl drahtlos

²¹User Datagram Protocol

²²Voice over IP

²³File Transfer Protocol

²⁴Internet Message Access Protocol

²⁵Simple Mail Transfer Protocol

²⁶Secure Shell

kommunizierender Geräte rasant zu. Die verwendeten, drahtlosen Kommunikationstechnologien sind dabei sehr spezifisch an ihre jeweiligen Anwendungsgebiete gebunden. Parameter wie bspw. die maximale Übertragungsrate, die effektive Sendereichweite oder der durchschnittliche Energieverbrauch müssen entsprechend angepasst werden. Weiterhin spielen das nutzbare Frequenzband und die verwendeten Kollisionsmechanismen eine wichtige Rolle für die Qualität und Zuverlässigkeit des Kommunikationssystems.

Zur Bearbeitung dieser Forschungs- und Entwicklungsthemen wurde im Februar 1980 das *Projekt 802 des IEEE*²⁷ [48] gegründet, welches sich primär mit lokalen Netzwerken befasst. Aufgabe von *IEEE 802* ist die Standardisierung verschiedener Technologien auf den beiden unteren Ebenen des ISO/OSI-Referenzmodells, wodurch eine herstellerübergreifende Kompatibilität der Endgeräte gewährleistet werden kann. Zusätzlich wirkt diese Standardisierung der Entstehung zu vieler gleichartiger, aber dennoch inkompatibler Technologien entgegen. Die folgenden Abschnitte sowie Tabelle 2.2 geben einen Überblick über die, in Bezug auf diese Arbeit, relevanten Technologien.

	IEEE 802.15.4	Bluetooth	IEEE 802.11 b/g/n
max. Datenrate	250 KBit/s	1 MBit/s	300 MBit/s
Reichweite (m)	10 - 75	1, 10, 100	50 - 150
Energiebedarf	niedrig	mittel/hoch	hoch
Sendeleistung (mW)	1, 100	1, 100	50 - 100 (1000 in USA)
Frequenz (Mhz)	868, 915, 2400	2400	2400, 5000

Tabelle 2.2.: Übersicht der wesentlichen, im Rahmen dieser Arbeit betrachteten Funkstandards.

2.2.1. Wireless PAN - IEEE 802.15

Auf dem Gebiet der drahtlosen *Nahbereichskommunikation* spricht man von *WPAN*²⁸. In diesem Zusammenhang wurde durch das IEEE eine spezielle Arbeitsgruppe *802.15* definiert. Innerhalb dieser Gruppe existieren sechs Unterabteilungen. Besonderer Fokus soll aber auf den beiden Normen *802.15.1* und *802.15.4* liegen. *802.15.1* entspricht dabei der *Bluetooth Technologie* in Version 1.1 bzw. 1.2 [35, 168]. Der Funkstandard *ZigBee* basiert auf der *802.15.4* Spezifizierung und fokussiert somit einen optimierten Energieverbrauch [169, 196].

Bluetooth IEEE 802.15.1

In den 1990er Jahren entwickelte die Firma *Ericsson* den Industriestandard *Bluetooth* zur Überbrückung kurzer Distanzen zwischen mobilen Endgeräten. Bluetooth ersetzt dabei die veraltete Infrarot-Technologie, welche eine direkte Sichtverbindung zwischen Sender und Empfänger er-

²⁷Institute of Electrical and Electronics Engineers

²⁸Wireless Personal Area Networks

2. Grundlagen und Modell

fordert. Für die Übertragung mit *Bluetooth* wurde dazu das lizenzfreie *ISM-Band*²⁹ im 2,4GHz Bereich verwendet. In der ursprünglichen Versionen 1.0, 1.1 sowie 1.2 wurde dabei eine Übertragungsrate von maximal 732,2kBit/s erreicht. Diese konnte in der Version 2.0 auf 2,1Mbit/s erhöht werden. Bei allen Versionen kommt dabei ein *Frequenzhopping*-Verfahren (*FHSS*³⁰) für die Datenübertragung zum Einsatz, bei dem die Trägerfrequenz innerhalb von Sekundenbruchteilen nach einem definierten Schema gewechselt wird. Die Wechselrate beträgt dabei 1600 Kanäle pro Sekunde. In Folge dessen können Interferenzen während der Datenübertragung deutlich reduziert werden. Im Gegensatz zu vielen anderen Funkstandards basiert *Bluetooth* auf einer Vielzahl unterschiedlicher, anwendungsorientierter *Kommunikationsprofile*. Einige dieser Profile sind bspw.:

HID (Human Interface Device Profile) drahtlose Eingabegeräte

PAN (Personal Area Networking Profile) Netzwerkverbindungen

SPP (Serial Port Profile) Serielle Datenübertragung (Emulierung einer RS232 Verbindung)

A2DP (Advanced Audio Distribution Profile) Übermittlung von Audiodaten

OBEX (Object Exchange) Datenübertragung zwischen zwei Geräten

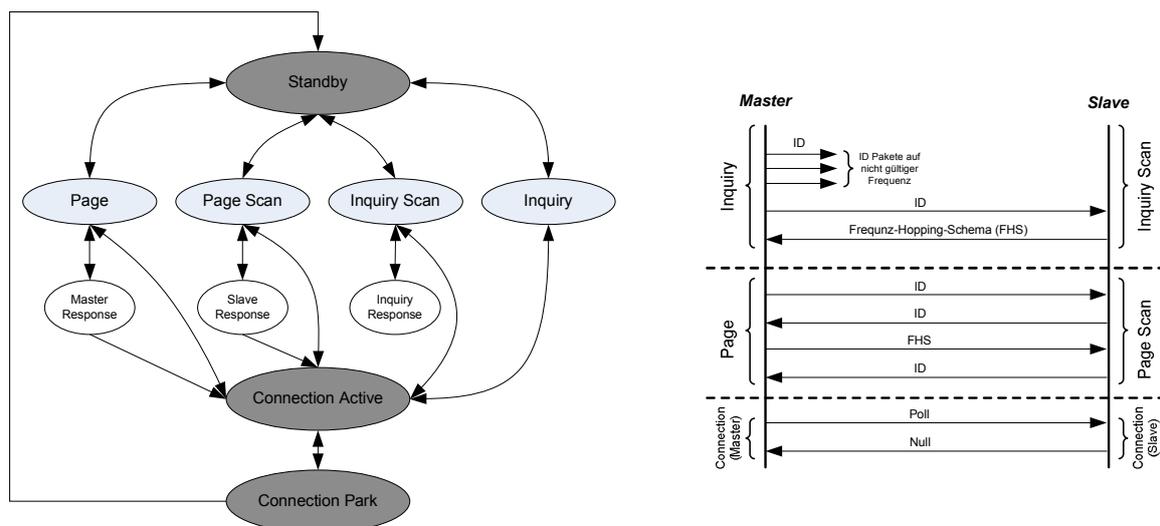


Abbildung 2.7.: Bluetooth Verbindungsmanagement für den Aufbau eines gültigen Kommunikationskanals. Dazu links die Darstellung der einzelnen Zustandsmenge und rechts der Ablauf des Verbindungsaufbaus in Form eines Ablaufdiagramms.

Dieses dynamisch erweiterbare, profilbasierte System verleiht *Bluetooth* im direkten Vergleich zu vergleichbaren Kommunikationsstandards eine deutlich höhere Flexibilität und ermöglicht den vielseitigen Einsatz in mobilen Endgeräten. Dazu tauschen die Endgeräte nach einem initialen Verbindungsaufbau Informationen über die verfügbaren Dienste aus und wählen dann ein geeignetes Schnittstellenprofil aus. Der grundlegende Verbindungsaufbau soll anhand der nachfolgenden Abbildung 2.7 verdeutlicht werden.

²⁹Industrial, Scientific and Medical Band

³⁰Frequency Hopping Spread Spectrum

Aufbauend auf diesem Verbindungskanal erfolgt nun die Aushandlung eines geeigneten Verbindungsprofils. Die abstrahierte Verbindung kann im Anschluss durch darüberliegende Protokollebenen bzw. Softwareapplikationen genutzt werden. Die folgende Abbildung 2.8 verdeutlicht den Aufbau des vollständigen Protokollstacks.

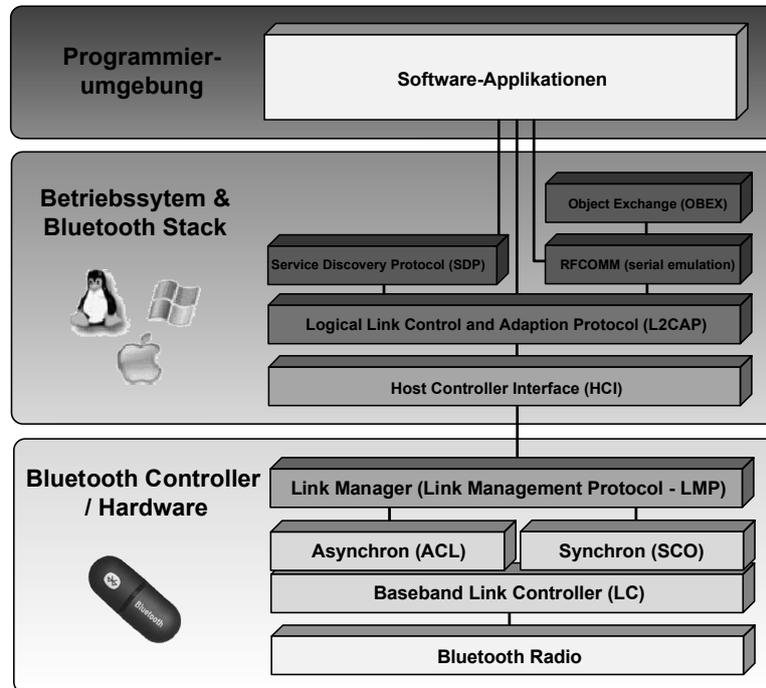


Abbildung 2.8.: Bluetooth Protokollstack mit den Einteilungen in Hardware-, Treiber- und Softwarefunktionalität.

Die Initiierung eines Kommunikationskanals mit einem handelsüblichen Mobiltelefon und einer entsprechenden Gegenstelle auf Basis von Bluetooth könnte beispielhaft wie folgt ablaufen:

1. Gerätesuche per *Inquiry Scan* (Verbindungsaufbau über *LMP*³¹)
2. Verbindungsanfrage auf Basis der ermittelten Hardwareadresse (*LMP*)
3. Aufbau eines *L2CAP*³² Kanals
4. Datenbankabfrage (selektiv oder komplett) von Dienstinformationen (Profile) über *SDP*³³
5. Auswahl eines geeigneten Profils
6. Aufbau eines Kanal auf *RFCOMM Ebene* (Emulation einer seriellen Schnittstelle)
7. Kommunikation über den *RFCOMM Kanal* nach Profilspezifikation mit entsprechenden Protokollen

³¹Link Manager Protocol

³²Logical Link Control and Adaption Protocol

³³Service Discovery Protocol

2. Grundlagen und Modell

Den genannten Vorteilen von Bluetooth steht ein relativ hoher Energiebedarf in Relation zur effektiven Sendereichweite und Datenrate gegenüber. Die hohe Komplexität des *Bluetooth Stacks* erschwert zudem die Einarbeitung im Rahmen eigener Entwicklungen auf Softwareebene. Zur maximal zulässigen Sendeleistung von 100mW (Klassifizierung Tabelle 2.3) addiert sich ein erhöhter Energiebedarf durch einen entsprechend dimensionierten Mikrocontroller, welcher die Protokollverarbeitung steuert. In der Praxis nutzen mobile Endgeräte meist Sendemodule mit geringerer Leistung, wodurch sich eine effektive Sendereichweite von ca. 10m ergibt.

BT-Klasse	Maximale Sendeleistung	theoretische Reichweite	Dämpfung Antenne
Klasse 1	100 mW	20 dBm	ca. 100 m
Klasse 2	2,5 mW	4 dBm	ca. 50 m
Klasse 3	1 mW	0 dBm	ca. 10 m

Tabelle 2.3.: Bluetooth: Übersicht Klassen und Reichweiten nach Spezifikation

Um auch in Zukunft den Anforderungen moderner, multimedialer Anwendungen gerecht zu werden, sieht die Weiterentwicklung in der Nahbereichskommunikation den Einsatz von Ultrabreitband-Technologien vor [22], wodurch die maximale Datenrate um ein Vielfaches erhöht und die Störanfälligkeit gleichzeitig minimiert werden kann.

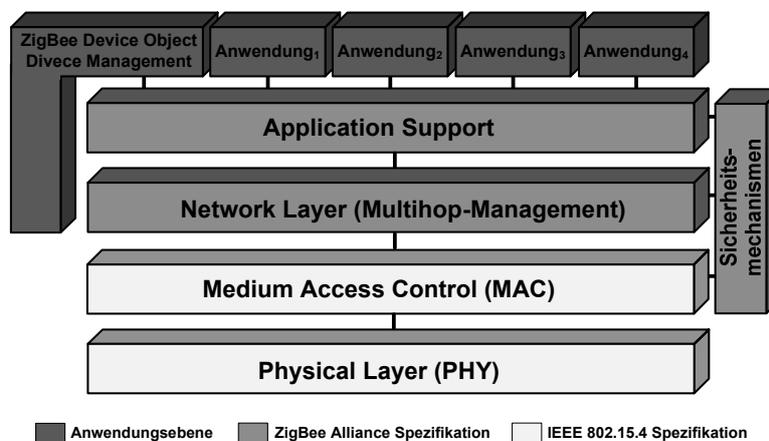


Abbildung 2.9.: IEEE 802.15.4 / ZigBee Protokollstack mit der Aufteilung in die jeweilige Spezifikation.

IEEE 802.15.4 und ZigBee

Im Gegensatz zu 802.15.1 fokussiert 802.15.4 Kommunikationssysteme mit einer sehr niedrigen Datenrate und möglichst geringem Energieverbrauch. Anwendungsgebiete sind hierbei vor allem *drahtlose Sensornetzwerke*, welche meist über einen weniger komplexen Protokollstack verfügen. Basierend auf den beiden nach 802.15.4 spezifizierten unteren Protokollebenen setzt der *ZigBee* Funkstandard auf (Abbildung 2.9). Die effektive Sendereichweite liegt hier, ähnlich dem *Bluetooth*-Standard, bei 10-30m (maximal 100m) und die maximale Datenrate

beträgt nur 250KBit/s bei einer zulässigen Sendeleistung von 10mW. Für die vorgesehenen Anwendungsszenarien, wie bspw. die Heim- und Gebäudeautomatisierung bzw. die Industrie- und Automatisierungstechnik reichen diese Leistungsmerkmale aber vollkommen aus. Zur Vermeidung von Interferenzen stehen 16 vordefinierte Kanäle für die Kommunikation zur Verfügung.

Die IEEE 802.15.4 Spezifikation erlaubt dabei zwei unterschiedliche Topologien, dargestellt in Abbildung 2.10. Einerseits eine klassische, zentralisierte Sterntopologie mit einer koordinierenden Knoteninstanz, dem *PAN-Koordinator*. Über ihn erfolgt die Anmeldung am Netzwerk sowie die Übertragung aller Daten. Folglich sollte der gewählte Knoten über entsprechende Leistungs- und Energieressourcen verfügen. Alternativ steht ein dezentrales *Peer-to-Peer* Kommunikationsszenario zur Verfügung, bei dem zwar ebenfalls ein Koordinator definiert wird, dieser jedoch keine Relevanz für den Kommunikationsablauf hat. Der Verbindungsaufbau erfolgt direkt zwischen Quelle und Senke, jedoch ausschließlich per *Single-Hop*, also auf direktem Weg ohne die Datenweiterleitung durch einen Zwischenknoten. Eine *Multi-Hop*-Funktionalität, sprich die Routenberechnung und anschließende Datenvermittlung über mehrere Relaisstationen sieht die Spezifikation nicht vor und muss bei Bedarf auf höheren Ebenen manuell implementiert werden.

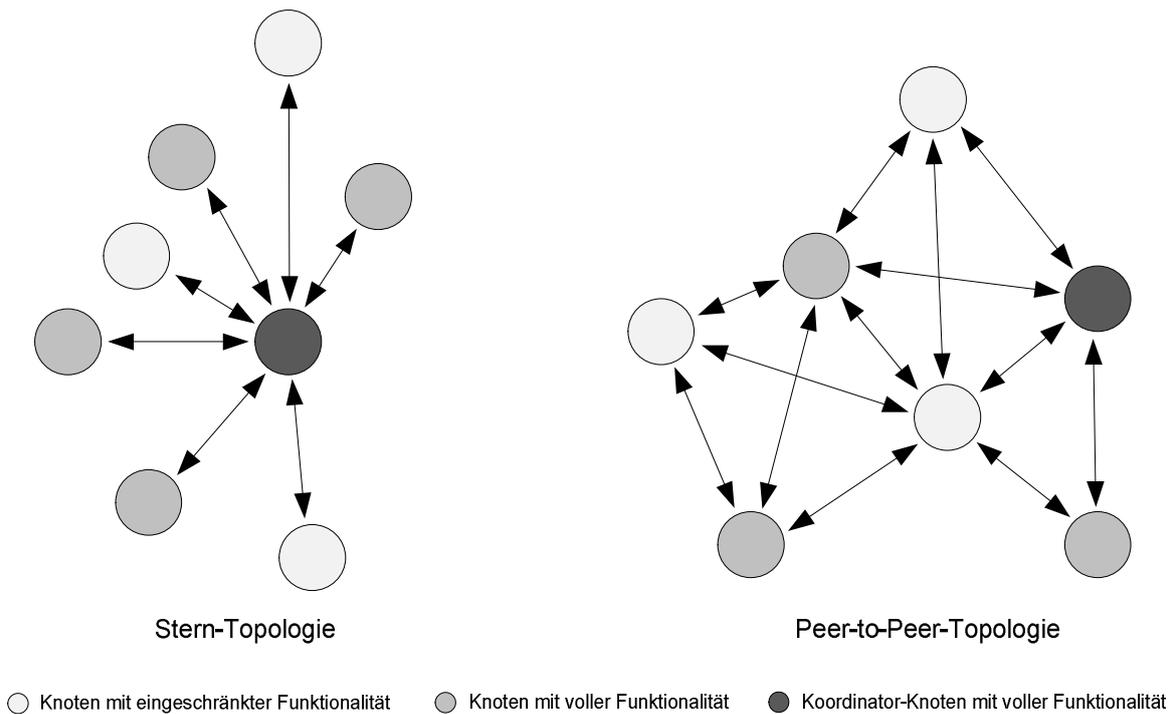


Abbildung 2.10.: Die nach IEEE 802.15.4 spezifizierten Topologien erlauben sowohl *Stern-* (links) als auch *Peer-to-Peer (P2P)* Topologien (rechts). Während der *PAN-Koordinator* in zentralisierten Netzstrukturen die gesamte Topologie verwaltet, spielt er in der P2P Struktur nur eine untergeordnete Rolle.

ZigBee soll im Wesentlichen dazu dienen, eine Vielzahl einfacher Geräte um eine zentrale Kontrollinstanz drahtlos zu vernetzen. Der Standard legt dabei besonderen Wert auf Stabilität und geringe Latenzzeiten. In der Praxis ergeben sich allerdings Probleme bei einer Koexistenz mit *Wireless LAN* und *Bluetooth*, da auch *ZigBee* das freie ISM-Frequenzband bei 2,4GHz für die

2. Grundlagen und Modell

Übertragung nutzt. Zusätzlich sind zwei weitere Frequenzbänder von 868-870MHz (Europa) bzw. 902-928MHz (Nordamerika) vorgesehen.

2.2.2. Wireless LAN - IEEE 802.11

Neben *Bluetooth* stellt *IEEE 802.11*, oder auch *WLAN*³⁴ genannt, eine der am häufigsten eingesetzten und bekanntesten drahtlosen Kommunikationstechnologien der heutigen Zeit dar [148, 167]. Im Gegensatz zu den eben vorgestellten Nahbereichstechnologien bietet WLAN sowohl höhere Datenraten als auch eine erhöhte Reichweite. Im Laufe der vergangenen Jahre wurden mehrere WLAN Kommunikationsstandards durch das IEEE spezifiziert. Nachdem die erste Version *802.11b* Datenraten von nur 11MBit/s erreichte, wurde der Datendurchsatz in der *802.11g* Spezifizierung auf 54MBit/s erhöht. Während diese Standards das 2,4GHz ISM-Band nutzen, erlaubt die *802.11a* Spezifikation das Senden in einem breiteren Frequenzbereich des 5,1GHz Bandes. Um eine effektive Sendereichweite von 100-300m außerhalb und 30m innerhalb geschlossener Gebäude zu gewährleisten, ist die maximale Sendeleistung auf 100mW (2,4GHz ISM-Band) bzw. 500mW (5,1GHz Band) begrenzt. Je nach WLAN Spezifikation verfügen alle Kommunikationsstandards nach 802.11 über mehrere, vordefinierte Funkkanäle innerhalb des jeweiligen Frequenzbandes. Sie ermöglichen eine weitestgehend störungsfreie Kommunikation auch bei einem Betrieb mehrerer paralleler Funknetze. Die verfügbare Anzahl hängt dabei stark von gesetzlichen Beschränkungen ab.

WLAN hat sich in den vergangenen Jahren zur einer Standardschnittstelle für mobile Endgeräte wie Laptops oder PDAs entwickelt und ermöglicht primär den einfachen, kabellosen Zugang zu privaten Netzwerken und dem Internet. Nachdem anfängliche Sicherheitsprobleme in den Verschlüsselungsmechanismen (*WEP*)³⁵ beseitigt wurden, hält WLAN durch zuverlässige Verschlüsselungsverfahren, wie bspw. *WPA/WPA2*³⁶, mehr und mehr Einzug sowohl im privaten als auch im industriellen Umfeld [48, 148].

Mit Einführung der neuen *802.11n* Spezifikation erhöht sich die maximale Datenrate durch den Einsatz von *MIMO-Technologien*³⁷, also der gleichzeitigen Nutzung mehrerer Funkkanäle, nochmals erheblich auf bis zu 300MBit/s. Somit kann auch der Leistungsunterschied zu herkömmlichen, drahtgebundenen Netzwerktopologien vernachlässigt werden und es ist damit zu rechnen, dass die WLAN Technologie in naher Zukunft die drahtgebundene Netzwerkkonnektivität im privaten Sektor verdrängen wird.

³⁴Wireless Lokal Area Network

³⁵Wired Equivalent Privacy

³⁶Wi-Fi Protected Access

³⁷Multiple Input Multiple Output

2.2.3. Wireless MAN - IEEE 802.16

Um auch über größere Distanzen eine breitbandige Kommunikation zu gewährleisten, wurde die *WiMAX*-Technologie³⁸ in Form des *IEEE 802.16* Standards spezifiziert [48, 166]. Innerhalb von *IEEE 802.16* existieren unterschiedliche Teilspezifikationen, welche sich sowohl mit stationären als auch mit mobilen Kommunikationsnetzen auseinandersetzen. Im Gegensatz zu *Wireless LAN* definiert bei *WiMax* die Basisstation das Scheduling für die einzelnen Übertragungen der Klienten und vergibt dazu entsprechende Zeitslots.

In zukünftigen *MANs*³⁹ stellt *WiMAX* somit hohe Datenraten bei geringen Übertragungslatenzen zur Verfügung. Dabei soll auf Basis dieser Technologie eine effektive Flächenabdeckung bis ca. 30km realisiert werden und dabei eine drahtlose Alternative zu bereits verfügbaren drahtgebundenen *DSL* Anschlüssen darstellen.

2.3. Selbstorganisation

Mit der Entwicklung immer komplexerer, verteilter Systeme in der Kommunikations- und Informationstechnik, treten zunehmend Probleme in Bezug auf den Entwurf sowie die Kontrolle in den Vordergrund. Um die Beherrschbarkeit in Zeiten immer schneller ansteigender Komplexität und Funktionalität auch in Zukunft gewährleisten zu können, stehen *selbstorganisierende Systeme* mit ihren spezifischen Eigenschaften im Fokus wissenschaftlicher Forschungen. Die Verfahrensweisen der *Selbstorganisation* stellen dabei eine grundlegend neue Idee im technischen Umfeld dar [66].

Da der Begriff in der Literatur sehr vielseitig und für unterschiedlichste Bereiche verwendet wird, fehlt eine einheitliche Definition. Stellvertretend für *Selbstorganisation* im technischen Bereich wird der Begriff in [192] treffend umschrieben:

”Self-organisation is a dynamical and adaptive process where systems acquire and maintain structure themselves, without external control.”

Ausgehend von der eigentlichen Selbstorganisation in biologischen Systemen, in denen gleichartige oder hierarchisch geordnete Elemente durch kollektives und kooperatives Verhalten komplexe Abläufe vollziehen können, wurden die wesentlichen, sogenannten *Self-X* Eigenschaften zunächst auf Bereiche der *künstlichen Intelligenz* (*künstliche neuronale Netze*, *Multiagentensysteme*) und der *Systemtheorie* übertragen. Neben der eigentlichen Selbstorganisation umfassen die *Self-X* Eigenschaften unter anderem eine *Selbst-Konfiguration* und *-Adaption*, *Selbst-Heilung* oder *Selbst-Schutz*. Da in der Literatur der Umfang sogenannter *Self-X* Eigenschaften stark diskutiert wird, können konkrete Definitionen und Abgrenzungen bis zum jetzigen Zeitpunkt nicht getroffen werden. Ein zentraler Aspekt ist die fehlende, zentrale Kontroll- bzw. Steuerinstanz, wodurch keine oder nur eine begrenzte Sicht auf das Gesamtsystem verfügbar

³⁸Worldwide Interoperability for Microwave Access

³⁹Metropolitan Area Network

2. Grundlagen und Modell

ist. Jedes Einzelement besitzt einen vordefinierten, durch Lernen erweiterbaren Regelsatz, welcher das Agieren und Reagieren festlegt. Durch ein Analysieren der Umwelt und das entsprechende Zusammenwirken der einzelnen Elemente wird eine Anpassung des Gesamtsystems erreicht. Selbstorganisation erlaubt somit einen hohen Grad an Flexibilität und Robustheit gegenüber äußeren Störeinflüssen. Einhergehend mit der Lernfähigkeit solcher Systeme können die Entwicklungszeiten unter Umständen erheblich gesenkt werden.

Im technischen Umfeld der Informatik und Informationstechnik stellt Selbstorganisation ein Schlüsselkonzept für die Beherrschbarkeit zukünftiger hochkomplexer, intelligenter Systeme dar [33, 103, 170]. Hier ermöglichen die Eigenschaften eine besonders gute Abbildung auf Anwendungsgebiete moderner Kommunikationstechnologien in verteilten Systemen. Auch auf diesem Gebiet kann eine immer stärkere Vermaschung unterschiedlicher, heterogener Netzwerke beobachtet werden. Besonders im Bereich *mobiler Ad Hoc* bzw. *Mesh-Netzwerke* [56] bietet die Selbstorganisation viele Einsatzmöglichkeiten. Ziel hierbei ist bspw. eine ressourcensparende Topologieverwaltung oder ein skalierbares Routing mit einer möglichst effizienten Lastverteilung.

Selbstorganisierende Aspekte können dabei auch auf das in dieser Arbeit vorgestellte Kommunikationskonzept angewendet werden, bei dem die *funktstandardübergreifenden Netzwerkknoten* autonom in einer Ad Hoc Topologie agieren. Dies umfasst auch Möglichkeiten einer erweiterten *Topologiekontrolle*, mit deren Hilfe aus dem Pool verfügbarer Kommunikationspartner sinnvolle Verbindungskanäle selektiert und dynamisch verwaltet werden.

2.4. Schnittstellensynthese

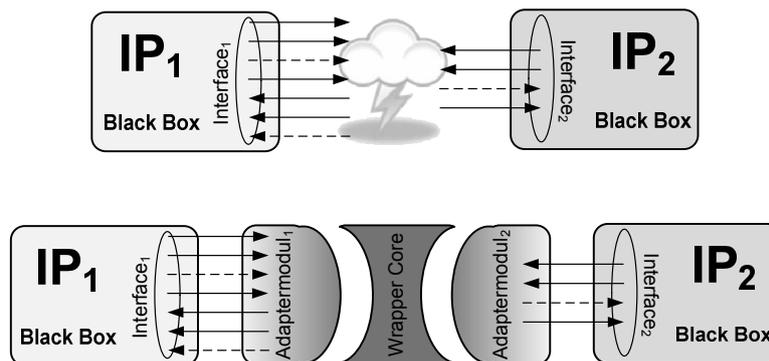


Abbildung 2.11.: Möglichkeiten der Kopplung zweier nicht kompatibler IPs, welche als Black Box Komponente vorliegen. Die Blöcke stellen vordefinierte, nicht modifizierbare Schnittstellen für die Anbindung an ihre Umwelt zur Verfügung. Ohne entsprechende Wrapper-Module inkl. der nötigen, logischen Konverter-Funktionalität ist eine Kommunikation nicht möglich.

Im Bereich eingebetteter Systemen treten zunehmend Probleme in Hinblick auf die Beherrschbarkeit der ansteigenden Komplexität in den Fokus wissenschaftlicher Betrachtungen. Um den Entwicklungsaufwand zu reduzieren versuchen die Entwickler, bereits verfügbare Komponen-

ten wiederzuverwenden oder zu erweitern. Die Hersteller wiederum schützen ihre Komponenten vor Zugriffen und Modifikationen durch Dritte. Die entwickelten und getesteten Komponenten - sogenannte *IPs*⁴⁰ - stehen in Folge dessen nur mit einer begrenzten Sicht auf die Funktionalität zur Verfügung. Diese Sicht wird auch als *Black Box* bezeichnet. In den meisten Fällen beschränkt sich die Dokumentation auf den Funktionsumfang, die Rahmenbedingungen für den störungsfreien Betrieb sowie die Interaktion über externe Schnittstellen. Interne Details der Komponenten werden nicht offen gelegt.

Für die Entwicklung neuer Systeme können diese fertigen Komponenten also nur nach der entsprechenden Spezifikation und über die vordefinierten Schnittstellen genutzt werden. Für die Realisierung einer komplexen Gesamtfunktionalität muss eine Vielzahl solcher Teilkomponenten möglichst effizient gekoppelt werden (Abbildung 2.11). Diese Kopplung erweist sich aufgrund der Einschränkungen in vielen Fällen als schwierig und zeitintensiv [81]. Ein Großteil der bestehenden Inkompatibilitäten lässt sich dabei auf funktionale Unterschiede der jeweiligen Schnittstellen zurückführen. Eine *Schnittstelle* sei hierbei nach [160] wie folgt definiert:

”Eine Schnittstelle ist eine Trennstelle zweier (Teil-) Systeme. An ihr können die Systeme getrennt werden. Obwohl die Schnittstelle die Grenze zwischen zwei Systemen darstellt, ist sie für ein System, d.h. aus dessen Sicht, definiert.”

Während einer Datenübertragung ist das Zusammenwirken der Signale auf unterschiedlichen Ebenen in Protokollen definiert. Ein Protokoll sei dabei nach [159] wie folgt definiert:

”Ein Protokoll ist eine Menge von Regeln, die das Verhalten von Instanzen oder Prozessen bei der Kommunikation festlegt. Die Regeln betreffen u.a. den Aufbau und Abbau von Verbindungen, die Formate der auszutauschenden Nachrichten, den benutzten Code und Absprachen sowie eine Fehlererkennung und Fehlerkorrektur. Protokolle werden größtenteils durch Software, teilweise aber auch durch Hardware realisiert.”

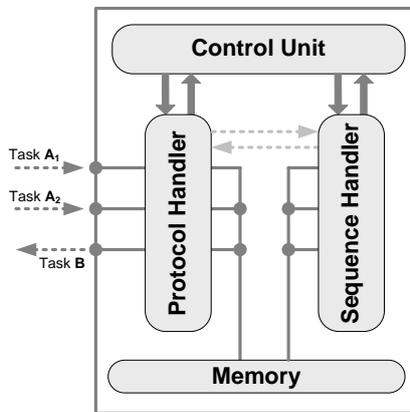
Um nun eine Kopplung zweier inkompatibler Schnittstellen mit evtl. nicht konformen Protokollen zu ermöglichen, wurden auf dem Forschungsgebiet der Schnittstellensynthese Konzepte entwickelt, mit denen diese Hindernisse überwunden werden können. Im Rahmen der Forschungsarbeiten zur *automatisierten Interface Synthese (IFS)*⁴¹ [81, 82] wurde der *Interface Synthesis Design Flow (IFS-Flow)* als eine Methodik zur Modellierung und automatisierten Generierung rekonfigurierbarer Schnittstellen entwickelt. Basierend auf diesem *Design-Flow* entsteht ein *Adapter Modul* [90], welches die direkte Konvertierung zwischen den Schnittstellen mehrerer Kommunikationskomponenten innerhalb eines eingebetteten Systems ermöglicht. Das Konzept soll an dieser Stelle kurz vorgestellt werden.

⁴⁰Intellectual Property - geistiges Eigentum

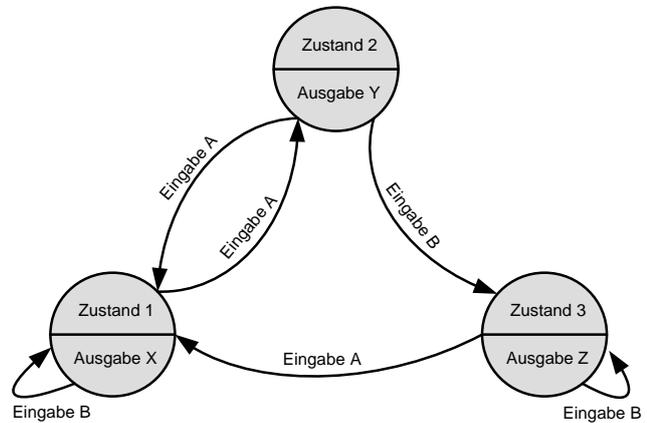
⁴¹Interface Synthesis

2.4.1. Interface Block (IFB)

Der *IFB* resultiert aus einem erfolgreichen *IFS-Flow* und realisiert das Bindeglied der jeweiligen Kommunikationskomponenten. Die Struktur des IFB (Abbildung 2.12(a)) besteht aus drei zentralen Komponenten.



(a) IFB Makrostruktur mit den drei zentralen Komponenten: *Control Unit* (CU) sowie *Protocol Handler* (PH) und *Sequence Handler* (SH), welche die Daten über gemeinsamen Speicher und einen entsprechenden Bus austauschen.



(b) *Endlicher Automat (FSM)* als eine endlichen Menge von Zuständen mit Eingabe-, Ausgabe- und Zustandsüberföhrungs-funktion.)

Abbildung 2.12.: IFB Makrostruktur / endlicher Automat.

Control Unit - CU

Die *Kontroll-Einheit* (*Control Unit - CU*) steuert alle Aktionen des Adapter Moduls und regelt dadurch den kompletten Datenfluss für die angeschlossenen Kommunikationskomponenten. Für die Steuerung der einzelnen IFB-internen Module stehen mehrere Daten- und Kontrollsignale zur Verfügung. Die CU hat keine dedizierte Schnittstelle zur Außenwelt und kann somit während der Laufzeit nicht durch den Nutzer modifiziert werden.

Protocol Handler - PH

Die *Protocol Handler* repräsentieren die externen Schnittstellen des IFB's mit der Umwelt. Sie verwalten den zeitlichen Protokollablauf ein- und ausgehender Datenpakete. Dabei ist der IFB in der Lage, mehrere Schnittstellen angeschlossener Komponenten durch unabhängige Modi im PH zu verwalten. Jeder Modus behandelt dabei jeweils ein Protokoll einer Schnittstelle. Im Falle eingehender Datenpakete entfernt der PH die Headerinformationen, speichert diese für die spätere Weiterverarbeitung und übergibt die extrahierten Nutzdaten an den *Sequence Handler*. Nach dem Anpassen der Nutzdaten durch den SH übernimmt ein weiterer Modus des PH's das

Verpacken der Nutzdaten für das jeweilige Zielprotokoll. Für diese Funktionalitäten werden sowohl die Protocol Handler als auch die Sequence Handler durch *endliche Automaten (FSM⁴²)* repräsentiert.

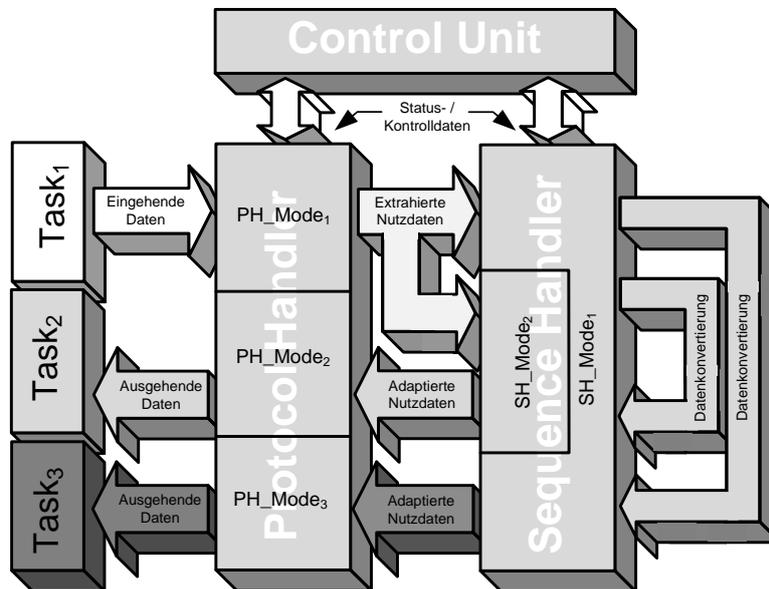


Abbildung 2.13.: Beispielhafter Datenfluss innerhalb des IFB's. Eingehende Daten von $Task_1$ werden vom entsprechenden *Protocol Handler Modus* bearbeitet. Die extrahierten Nutzdaten gelangen anschließend in Abhängigkeit von der Zielkomponente in den *Sequence Handler Modus* 1 oder 2. Nach der Adaption der Nutzdaten werden diese im *Protocol Handler Modus* 2 oder 3 in die jeweiligen *Protokollheader* verpackt und gemäß den Protokollspezifikationen an das Zielsystem ($Task_2$ bzw. $Task_3$) weitergeleitet.

Sequence Handler - SH

Während der Protokollkonvertierung passt der SH das Datenformat entsprechend den Konventionen des Zielprotokolls an und legt die adaptierten Nutzdaten für die anschließende Weiterleitung im Speicher ab. Dies beinhaltet bspw. die Anpassung der Byteorder. Nachdem die CU entsprechende Statusinformationen erhalten hat, werden diese Daten dem verantwortlichen PH zur Verfügung gestellt und somit an das Zielsystem versendet. Auch die SH verfügen über mehrere Modi zur Behandlung unterschiedlicher Datenkonventionen.

Die modulare Struktur des IFB's ermöglicht einen hohen Grad an Flexibilität und soll die Wiederverwendung in Hinblick auf zukünftige Projekte gewährleisten.

Bezugnehmend auf diese Arbeit werden die konzeptionellen Grundlagen des *IFB*-Modells auf den entwickelten *PLANet* Prototypen (Kapitel 6) angewendet. Die Protokollkonvertierung ermöglicht dieser Kommunikationsplattform, während der Laufzeit zwischen unterschiedlichen Funkstandards dynamisch zu wechseln. Einzelheiten bezüglich der Protokollkonvertierung und des *PLANet*-Prototypen werden in Kapitel 4 bzw. 6 näher erläutert.

⁴²Final State Machines

2.4.2. Rekonfigurierung

Um die Flexibilität und Leistungsfähigkeit eingebetteter Systeme trotz begrenzter Ressourcen zu steigern, können Methoden der *Rekonfigurierung* zum Einsatz kommen. Der Begriff Rekonfigurierung findet grundsätzlich sowohl im Bereich Software als auch Hardware Anwendung. Der hier fokussierte Forschungsbereich der Hardware-Rekonfigurierung wurde im Rahmen des *DFG-Schwerpunktprogrammes 1148 - Rekonfigurierbare Rechensysteme* näher untersucht [55, 117].

Aufgrund der begrenzten Chipfläche heutiger, hochkomplexer Rechensysteme, können aufgrund von Platzmangel unter Umständen nicht alle Funktionen gleichzeitig in das System integriert werden. In diesem Zusammenhang eröffnen sich durch eine *dynamische Rekonfigurierung* völlig neue Lösungsmöglichkeiten und Anwendungsszenarien. Ziel ist es dabei, Systemen die Fähigkeit zu verleihen, während der Laufzeit die Funktionalität sowie die hardwareseitige Struktur partiell oder vollständig zu verändern. Auf dieser Basis wird eine selbstständige Adaption des Systems an die Anforderungen seiner Umwelt ermöglicht. Auch im Falle etwaiger Fehlfunktionen können somit automatisch alternative Konfiguration geladen werden.

2.5. Zusammenfassung

In diesem Kapitel wurden grundlegende Kenntnisse auf dem Gebiet der drahtlosen Kommunikation, der Selbstorganisation sowie der Schnittstellensynthese vermittelt, welche zum allgemeinen Verständnis dieser Arbeit beitragen sollen.

Dazu wurde im ersten Teil das *ISO/OSI Schichtenmodell* vorgestellt und die Unterteilung der Funktionalitäten in den jeweiligen Ebenen näher erläutert. Das Modell dient dabei als Designgrundlage für eine Vielzahl aktueller Kommunikationsprotokolle, wie bspw. den vorgestellten TCP/IP-Protokollstack.

Darauf aufbauend wurden unterschiedliche, nach *IEEE* genormte *Kommunikationsstandards* präsentiert und deren spezifische Eigenschaften sowie primäre Anwendungsfelder gegenübergestellt.

Anschließend wurden aktuelle Themen aus dem Bereich der *Selbstorganisation* erläutert und denkbare Einsatzmöglichkeiten in Hinblick auf das im Rahmen dieser Arbeit vorgestellte Konzept analysiert. Die Forschungsansätze bieten hier ein breites Spektrum für mögliche Ansatzpunkte.

Abschließend wurde auf den Forschungsbereich der *Schnittstellensynthese* näher eingegangen und die konkrete Anwendung im entwickelten *PLANet* Prototypen erläutert. Gleichzeitig wurde das Thema *Rekonfigurierung* kurz aufgegriffen und dessen Vorteile aufgeführt.

Im folgenden Kapitel werden nun aktuelle Forschungen im Bereich mobiler Ad Hoc Netzwerke sowie Entwicklungen auf dem Gebiet drahtloser Kommunikationstechnologien vorgestellt.

3. Stand der Forschung

Aufbauend auf den in Kapitel 2 vorgestellten Grundlagen sollen nun aktuelle Forschungen verschiedener Schwerpunktbereiche vorgestellt werden, welche in direktem oder indirektem Zusammenhang mit dieser Arbeit stehen. Im Fokus steht dabei die Vermittlung aktueller Technologien, welche aus wissenschaftlicher Sicht den derzeitigen Stand der Entwicklungen widerspiegeln. Die Ziele einzelner Forschungsbereiche korrelieren dabei teilweise mit Anforderungen und Zielstellungen dieser Arbeit. Auf diese partiellen Überschneidungen sowie die sich daraus ergebenden Vor- und Nachteile wird in den einzelnen Abschnitten näher eingegangen.

3.1. Mobile Ad Hoc Netzwerke

Das zentrale Forschungsthema, welches auch in dieser Arbeit einen wesentlichen Schwerpunkt bildet, befasst mit sogenannten *Mobilen Ad Hoc Netzwerken - MANETs*. MANETs repräsentieren einen Teilbereich auf dem Gebiet der Funknetze. Funknetze können wie folgt klassifiziert werden.

- Zellulare Netze
- Nahbereichs-Kommunikationsnetze
- Mobile Ad Hoc Netze mit drahtlosen Sensornetzwerken als eine anwendungsspezifische Teilmenge

Bevor im weiteren Verlauf dieses Kapitels auf ausgewählte Themenkomplexe im Bereich mobiler Ad Hoc Netzwerke näher eingegangen wird, sollen die verschiedenen Rubriken kurz erläutert werden.

Zellulare Netze

Zellulare Netze kommen vorwiegend im Mobilfunkbereich zum Einsatz und binden mobile Endgeräte an verfügbare Basis-/Relaisstationen, welche untereinander ein großflächiges, drahtgebundenes Netz aus Zugangspunkten zur Verfügung stellen. Aufgrund der strengen Netzwerkhierarchie ist die Konfiguration der jeweiligen Knoten sowie die Wegwahl auf einfache Art und Weise realisierbar. Da die Basisstationen über eine konstante Stromversorgung verfügen, können rechenintensive Prozesse von den mobilen Endgeräten auf diese Stationen ausgelagert werden, wodurch die Komplexität der Endgeräte deutlich reduziert werden kann. In Folge dessen koordiniert die Basisstation den gesamten Funkverkehr aller Knoten innerhalb der jeweiligen Zelle.

Die sternförmigen Topologien zellularer Netze wirken sich aber auch nachteilig auf die Kommunikationswege aus. Der gesamte Informationsfluss muss über die jeweilig verfügbaren Zu-

3. Stand der Forschung

gangspunkte abgewickelt werden, welche die eingehenden Daten zum entsprechenden Zielknoten weiterleiten. Eine direkte Kommunikation zweier Endgeräte ist nicht möglich, auch wenn sich diese in unmittelbarer Nachbarschaft befinden. Dies wirkt sich negativ auf den Energieverbrauch der einzelnen Knoten aus. Weiterhin ergibt sich hierdurch ein Flaschenhals in den Zugangspunkten, welche zum einen eine enorme Anzahl mobiler Endgeräte verwalten müssen als auch jedem Knoten eine definierte Datenrate zur Verfügung stellen sollten.

Aktuelle Technologien, welche auf zellularen Netzen basieren, sind bspw. die Mobilfunkstandards *GSM*¹ sowie die Nachfolger *UMTS*² und *WiMAX*. Auch *WLAN* Technologien nach *IEEE 802.11*, welche im *Infrastruktur / Access Point Modus* betrieben werden, realisieren ein zellulares Netz.

Nahbereichs-Kommunikation

Die Nahfeld- oder auch Nahbereich-Kommunikation stellt eine weitere Rubrik im Bereich der Funknetze dar. Hier werden im wesentlichen Punkt-zu-Punkt Verbindungen zwischen zwei Kommunikationspartnern betrachtet. Einen aktueller Vertreter stellt dabei *RFID*³ dar, welcher es ermöglicht, Objekte auf einfache Art und Weise zu identifizieren. Aber auch für die drahtlose Übertragung von sicherheitskritischen Informationen kommen Technologien aus der Nahbereich-Kommunikation zum Einsatz. Hier nutzt bspw. *Bluetooth* in der Version 2.1 *NFC*⁴ um einen einfachen, sicheren Schlüsselaustausch zweier Kommunikationspartner zu realisieren.

Mobile Ad Hoc Netze / Drahtlose Sensornetzwerke

Ein *mobiles Ad Hoc Netzwerk (MANET)*, auch *Mesh Netzwerk* genannt, ist eine Ansammlung von mindestens zwei mobilen Geräten mit der Fähigkeit, spontan auf drahtlosem Weg miteinander zu kommunizieren und somit ein Netzwerk aufzubauen. Die Geräte kommunizieren innerhalb ihrer Sendereichweite auf direktem Wege miteinander. Ist die direkte Kommunikation aufgrund dieser begrenzten Reichweite nicht möglich, müssen dazwischenliegende Geräte für einen Kommunikationsweg sorgen. Die Geräte fungieren in diesem Fall als Zwischenstation und übernehmen die Routingfunktionen im Netzwerk. Dieses entscheidende Merkmal ist charakteristisch für mobile Ad Hoc Netzwerke.

Im Gegensatz zu herkömmlichen Netzwerken kommen MANETs ohne stationäre Infrastruktur aus. Sie sind an keine fest definierte Infrastruktur gebunden, wodurch die Topologie anders als bei zellularen oder klassischen Netzwerken variiert. Somit folgt das vorliegende Kommunikationsparadigma einem reinen *Peer-to-Peer Ansatz (P2P)*, bei dem ein dezentraler, gleichberechtigter Rechner-Rechner-Verbund angestrebt wird und keine zentrale Verwaltungsinstanz existiert [114]. Alle Kommunikationsteilnehmer stehen hierbei auf der gleichen Hierarchieebene und stellen anderen Netzteilnehmern ihre Ressourcen in Form von Diensten zur Verfügung. Neben den reinen P2P Ansätzen existieren in der Literatur auch zentralisierte, hybride oder hierarchische P2P Systeme, auf die aber nicht näher eingegangen werden soll.

¹Global System for Mobile Communications

²Universal Mobile Telecommunications System

³Radio Frequency Identification

⁴Nearfield Communication

Durch den hohen Grad an Mobilität der Knoten in MANETs ändert sich die verfügbare Infrastruktur stetig. Mobile Ad Hoc Netzwerke verfügen über die in Kapitel 2.3 vorgestellten Self-X Eigenschaften und sind somit selbstorganisierend und selbstkonfigurierend. Dabei liegt der Fokus meist auf der Kopplung einer Vielzahl von drahtlosen Endgeräten mit mittlerer oder kurzer Kommunikationsreichweite (IEEE 802.11 und IEEE 802.15). Basierend auf einem gemeinsam definierten Kommunikationsstandard kommt in realen Anwendungsszenarien meist eine Vielzahl unterschiedlicher Netzwerkknoten zum Einsatz, welche sich in Größe, Rechenleistung, Laufzeit und Ausstattung unterscheiden. In Folge dessen impliziert der Begriff MANET im Allgemeinen eine heterogene Netzwerktopologie. Ein heterogenes Netz wird durch den Autor im Rahmen dieser Arbeit wie folgt definiert:

”Der Begriff heterogen leitet sich aus dem Griechischen von heterogenis ab und drückt eine Verschiedenartigkeit aus. Unter Heterogenität versteht man die Uneinheitlichkeit von Elementen einer Menge hinsichtlich eines oder mehrerer Merkmale. Heterogene Netzwerkstrukturen besitzen somit ungleichartige Knoten. Die Objekte eines Netzwerkes können dabei nach [190] auf verschiedenen Ebenen heterogen sein.”

MANETs bieten eine Vielzahl wesentlicher Vorteile gegenüber klassischen Netzwerken. Einhergehend mit der rasant steigenden Anzahl mobiler Kommunikationsgeräte bieten MANETs die Möglichkeit, die aufkommende Netzlast gleichmäßig zu verteilen, da keine zentralen Instanzen für die Verwaltung der Topologie benötigt werden. Somit *skalieren* MANETs besser als jede Form zellularer oder hierarchischer Funknetze. Aufgrund der *selbstorganisierenden Eigenschaften* sind mobile Ad Hoc Netzwerke äußerst vielseitig und flexibel einsetzbar. Auch die Robustheit gegenüber Störungen oder partiellen Netzausfällen macht den Einsatz von MANETs für ein breites Anwendungsspektrum interessant.

Der Forschungsbereich drahtloser Sensornetzwerke (engl. WSN⁵) bildet eine Teilmenge der Rubrik mobiler Ad Hoc Netzwerke. Die Knoten organisieren sich analog zu mobilen Ad Hoc Netzwerken selbständig und ohne Kontrollinstanz. Aufgrund des stark eingeschränkten Anwendungsgebietes ist die Leistungsfähigkeit der einzelnen Knoten gering. Diese reduziert sich auf das Sammeln von Daten mit Hilfe der verfügbaren Sensorik sowie einer optionalen Vorverarbeitung der gewonnenen Informationen. Im Gegensatz zu klassischen MANETs werden die gesammelten Daten anschließend über die verfügbare Infrastruktur zu einer zentrale Instanz übertragen, welche auf Basis der verschiedenen lokalen Informationen Rückschlüsse auf globale/systemweite Ereignisse ableitet. Weiterhin beschränkt sich der überwiegende Teil drahtloser Sensornetzwerke auf eine homogene Knotenlandschaft, in der jeder Sensorknoten über die gleiche Kommunikationshardware verfügt.

Die Realisierung solcher Ad Hoc Netzwerkstrukturen bringen jedoch einige Probleme mit sich, welche neben der Selbstorganisation bspw. auch aus den begrenzten Energieressourcen oder dem hohen Grad an Mobilität resultieren. Die nun folgenden Abschnitte gehen auf verschiedene Problemstellungen innerhalb dieses Forschungsbereiches näher ein.

⁵Wireless Sensor Networks

3.1.1. Topologieoptimierung

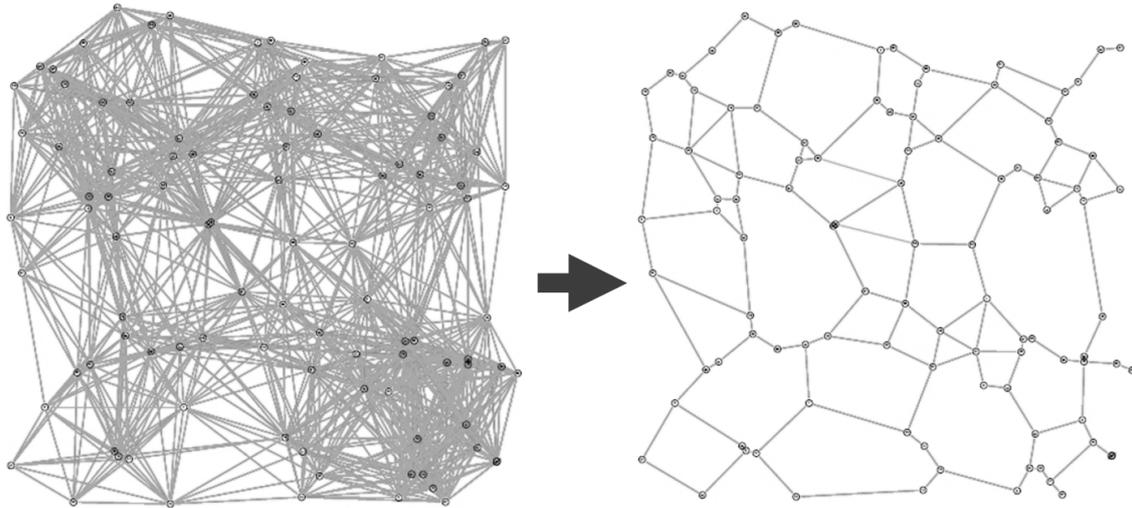


Abbildung 3.1.: Optimierung einer Topologie durch die Reduktion von Redundanzen. Überflüssige Kommunikationskanäle werden durch die Knoten nicht mehr genutzt.

Sollen die zur Verfügung stehenden Energieressourcen mobiler Knoten innerhalb eines Ad Hoc Netzwerkes möglichst effizient genutzt werden, stehen unterschiedliche Ansatzpunkte zur Verfügung. Ist dabei eine Modifikation auf physikalischer Ebene nicht möglich, stellt die *Topologieoptimierung* den ersten Ansatzpunkt dar. Auf Basis applikationsseitiger, anwendungsspezifischer Informationen, welche den unteren Protokollschichten zur Verfügung gestellt werden, findet eine Mediengriffskontrolle statt und es wird versucht, aus einer Vielzahl verfügbarer Kommunikationspartner möglichst sinnvolle Stationen herauszufiltern (Abbildung 3.1). Im Bereich mobiler Ad Hoc Netzwerke können sich daraus mehrere wesentliche Vorteile ergeben [36, 108–110, 172]. Zum einen sinkt in Folge der eingeschränkten Topologie die Netzlast durch die verringerte Menge an Kontrolldaten der verbleibenden Kommunikationspartner. Dies hat besonders bei sehr hoch skalierten Topologien Auswirkungen, da sich hier durch das Broadcasten von Information erhebliche Lastspitzen innerhalb der Topologie ergeben. Zum anderen sinkt durch eine Reduzierung des Vernetzungsgrades die mathematische Wahrscheinlichkeit für Datenkollision auf dem Übertragungsmedium erheblich. In Folge eines verringerten Interferenzlevels sinkt die Paketverlustrate, was die gesamte Kommunikation im Netzwerk positiv beeinflusst und auch das Auftreten kritischer, partieller Netzausfälle reduziert. Somit kommt eine gezielte Selektion unter den verfügbaren Kommunikationspartnern allen höher liegenden Protokollen, speziell den Routingverfahren, entgegen.

Obwohl einige wissenschaftliche Veröffentlichungen dem Aspekt der Interferenzminimierung widersprechen [41], existieren noch einige Vorteile, welche den Einsatz von Algorithmen zur Topologieoptimierung rechtfertigen. Besonders in mobilen Ad Hoc bzw. Sensornetzwerken ist der verringerte administrative Aufwand in den jeweiligen Knoten nicht zu unterschätzen. Durch die hohe Dynamik, vor allem in dichten Topologien, steigt der Aufwand für die Verwaltung erreichbarer Nachbarn und die Bearbeitung der entsprechenden Kommunikationskanäle schnell

auf ein kritisches Niveau an. Dem gegenüber stehen stark begrenzte Hardware-Ressourcen der Knoten, welche sich aus den Anforderungen einer möglichst langen Akku-Laufzeit ergeben.

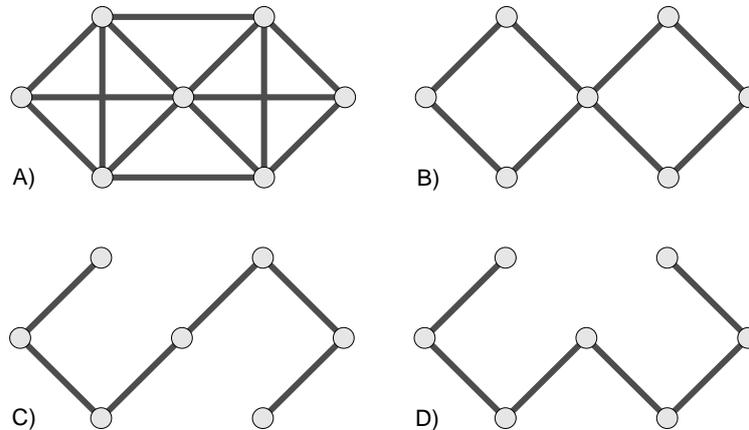


Abbildung 3.2.: Schrittweise Topologieoptimierung beginnend mit der maximal möglichen Vernetzung (A) bis zu zwei möglichen minimalen Vernetzungen (C/D).

Aus diesem Grund wurden unterschiedliche, teils sehr komplexe Verfahren entwickelt, welche neben der eigentlichen *Topologiekontrolle* eine *Topologieoptimierung* ermöglichen. Viele Veröffentlichungen gehen dabei speziell auf die Anforderungen und Problemstellungen im Bereich mobiler Ad Hoc und Sensornetzwerke ein [157, 180]. Einige grundlegende Mechanismen werden im Folgenden kurz erläutert.

Dynamischer Adaption der Sendeleistung

Mit Hilfe dieses einfachen Verfahrens kann die Sendeleistung des Funkmoduls je nach Situation angepasst werden. Die Netzwerkknoten sind damit in der Lage, die Anzahl erreichbar Kommunikationspartner aktiv zu beeinflussen [146]. Abbildung 3.2 verdeutlicht das Ergebnis einer solchen schrittweisen Adaption. Wird eine definierte Untergrenze unterschritten, wird die Sendeleistung entsprechend erhöht. Auf diese Weise kann der Energiebedarf für die Datenübertragung deutlich reduziert werden. In der Praxis zeigt sich aber, dass aktuell verfügbare Funkmodule teilweise erheblich mehr Energie im Empfangsmodus benötigen, was somit das Optimierungspotential relativiert. Weiterhin wird die Auswahl der Kommunikationspartner ausschließlich auf die Knoten mit der kürzesten Distanz beschränkt. Unter Umständen ergibt diese Wahl eine suboptimale Lösung oder führt zur Isolation einzelner Netzbereiche.

Lokale minimale Spannbäume

Das *LMST-Verfahren*⁶ [108] nutzt minimale Spannbäume für die Optimierung der Topologie. Jeder Knoten betrachtet dazu seine n -Hop Nachbarschaft und bildet über diese Knotenmenge den *lokalen Spannbaum*. Die Berechnung ist mit bekannten Algorithmen, bspw. nach *Kruskal* oder *Prim* auf einfache Art und Weise möglich. Bei letzterem ergibt sich dabei eine Worst-Case Laufzeit von:

⁶Local Minimum Spanning Tree

3. Stand der Forschung

$$O(|V| \cdot \log(|V|) + |E|)$$

$V \Rightarrow$ Anzahl Knoten

$E \Rightarrow$ Anzahl Kanten

Die Spannbaum-Berechnung beschränkt sich somit jeweils auf eine stark begrenzte Region, wodurch das Verfahren durch die geringe Knotenmenge effizient sehr gute Ergebnisse erzielt [1].

Selektive Vernetzung

Bei der gezielten Auswahl einzelner Kommunikationspartner spricht man von einer *selektiven Vernetzung*. Hier können je nach Anwendungsgebiet unterschiedlichste Parameter für die Auswahl einbezogen werden. In [37, 38] wird ein solches Verfahren vorgestellt. Da hier für eine vordefinierte Zahl von Nachbarschaftsknoten eine optimale Belegung errechnet wird, steigt der Aufwand entsprechend an. Aus diesem Grund wurde auf Basis evolutionärer Algorithmen das *EOSC-Verfahren*⁷ entwickelt [1–3]. Mit Hilfe drei biologischer Prinzipien (Selektion, Rekombination und Mutation) wird hier versucht, sich an die optimale Topologie anzunähern. Dabei wird die Möglichkeit einer suboptimalen Lösung in Kauf genommen. Im Gegenzug kann die Komplexität des Verfahrens deutlich reduziert werden.

In Hinblick auf das Thema dieser Arbeit bieten diese Mechanismen viele Ansatzpunkte für eine Optimierung. Im Bereich MANETs kann durch Methoden der Topologieoptimierung die Skalierbarkeit erhöht und der Energiebedarf gesenkt werden. Besonders bei dem Übergang von einer homogenen Netzstruktur hin zu einer integrierten, heterogenen Topologie sind deutliche Verbesserungen zu erwarten, da angepasste Verfahren hier die Möglichkeit bieten, zwischen unterschiedlichen Kommunikationskanälen zweier benachbarter Knoten auf Basis verschiedener Standards zu wählen.

3.1.2. Routing

Nachdem in Abschnitt 2.1 der TCP/IP Protokollstack mit einzelnen Ebenen strukturiert beschrieben wurde, soll im Folgenden die Funktionalität der Vermittlungsschicht nochmals aufgegriffen werden. Im Kontext mobiler Ad Hoc Netzwerke nehmen Verfahren für die Wegfindung in dynamischen Netzstrukturen eine zentrale Rolle ein [73, 153]. Die folgenden Abschnitte beschreiben daher wesentliche Verfahrensweisen im Bereich *Routing* und erläutern diese anhand konkreter Beispiele. Zuvor muss jedoch der Begriff *Routing* näher definiert werden.

Routing beschreibt die Art der Wegfindung durch ein Netzwerk. Im Rahmen des *Request for Comments (RFC) 1983* [115] wird der Begriff wie folgt umschrieben:

⁷Evolutionary Optimization Selective Connectivity

”Routing ist der Prozess der Auswahl der richtigen Schnittstelle und des nächsten Hops für Pakete, die weitergeleitet werden sollen.”

Gemeint ist dabei die Weiterleitung einzelner Informationseinheiten von einer Quelle zur Senke. Auf diesem Pfad müssen diese unter Umständen zielgerichtet, mehrere Zwischenstationen passieren. Eine durchlaufene Zwischenstation wird dabei als ein *Hop* bezeichnet und durch den Autor im Rahmen dieser Arbeit wie folgt definiert:

”Im Kontext der Rechnernetzkommunikation bezeichnet ein Hop den Weg einer Informationseinheit von einem Netzwerkknoten zum Folgenden auf dem Pfad von der Datenquelle zur Senke.”

Wird auf einem berechneten Routenpfad mindestens ein Zwischen-/Relaisstation genutzt, wird dies als *Multi-Hop-Routing* bezeichnet. Folglich unterscheidet man zwischen *Multi-Hop* und *Single-Hop-Kommunikation*.

”Bei einer Single-Hop-Kommunikation liegt zwischen Datenquelle und Senke genau ein Hop. Im Gegensatz dazu besitzt der Verbindungspfad einer Multi-Hop-Kommunikation n Hops zwischen den beiden Kommunikationspartnern, wobei $n \in \mathbb{N}, n > 1$. Die einzelnen Informationseinheiten werden somit über mehrere Zwischenstationen [zielgerichtet] weitergeleitet.”

In diesem Zusammenhang liegt der Fokus in dieser Arbeit auf einer *paket-orientierten* Kommunikation, wodurch die elementare *Informationseinheit* durch ein *Datenpaket* repräsentiert wird.

Bei kritischer Betrachtung erfolgt weiterhin eine Differenzierung der Begriffe *Routing* und *Forwarding*, da diesbezüglich in der Literatur viele Unklarheiten existieren. Während Routing den gesamten Pfad von der Quelle zur Senke und dessen Berechnung umschreibt und damit die Summe aller Zwischenstationen beinhaltet, definiert Forwarding die Weiterleitung von Informationen über ein verfügbares Netzwerkinterface zum nächsten Knoten (*Single-Hop*) innerhalb eines definierten Pfades.

Zum besseren Verständnis wird im weiteren Verlauf dieser Arbeit der Begriff Routing sowohl für die Berechnung eines Pfades durch ein Kommunikationsnetz als auch für die Weiterleitung von Datenpaketen über ein definiertes Interface verwendet. Weiterhin befasst sich dieser Abschnitt ausschließlich mit adaptiven Routingverfahren, die in der Lage sind, auf Veränderungen der Topologie zu reagieren.

Abbildung 3.3 zeigt beispielhaft ein Kommunikationsnetz mit gewichteten Verbindungskanten zwischen den einzelnen Netzwerkknoten. Für den Aufbau eines Kommunikationskanals zwischen zwei Knoten (Quelle und Senke) muss mit Hilfe eines gewählten Routingverfahrens ein optimaler Pfad durch das zugrundeliegende Kommunikationsnetz gefunden werden. Dabei fließen Nebenbedingungen, wie bspw. geringe Pfadkosten, wenig Zwischenstationen oder eine möglichst gleichmäßige Verteilung der Netzlast mit in die Berechnung des Routenpfades ein.

3. Stand der Forschung

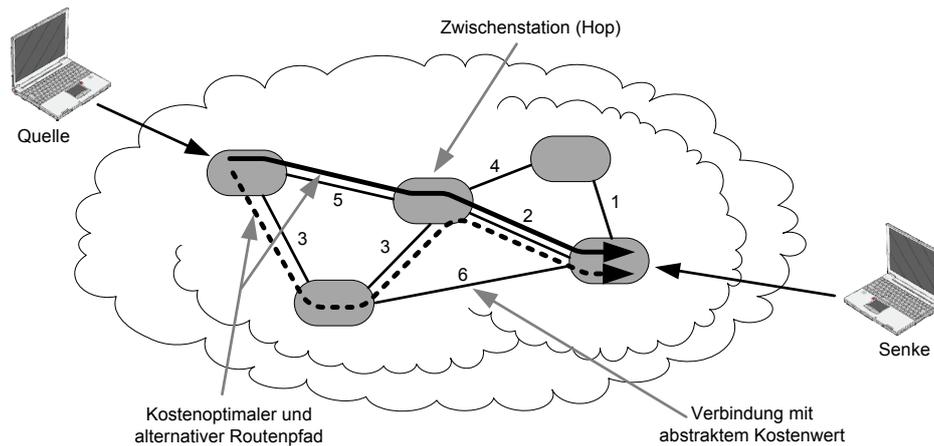


Abbildung 3.3.: Beispielhaftes Kommunikationsnetz mit Quell- und Zielknoten sowie einer Netzwerktopologie mit abstrakten Kosten.

Ein idealer Routing Algorithmus erfüllt dabei folgende Zielstellungen:

- *geringe Belastung* in Hinsicht auf die benötigten Ressourcen Rechenzeit und Speicher innerhalb aller beteiligter Knoten.
- *Optimierung* der Routenpfade in Abhängigkeit der Metrik und der aktuellen Rahmenbedingungen in der Topologie.
- *Leistungsfähigkeit* bzw. Verzögerungen welche aus der Entscheidungsfindung resultieren (*Response-Zeit*).
- *Skalierbarkeit* in direktem Zusammenhang mit dem Ressourcenverbrauch und der Leistungsfähigkeit. Auch bei großen Topologien muss ein reibungsloser und schneller Ablauf gewährleistet sein.
- *Stabilität* oder Robustheit gegen äußere und innere Störeinflüsse, was sich direkt auf die anderen Zielstellungen auswirkt.
- *Flexibilität* und die damit verbundene, möglichst schnelle Anpassungsfähigkeit an Änderungen im Netzwerk.
- [*Konvergenz*] und somit die Abgleichung der Wissensbasis aller Router-Elemente im Netzwerk. Wie schnell werden Informationen über Ereignisse im Netz verteilt. Dies trifft auf die betrachteten Routingverfahren nur teilweise zu, da die Mechanismen teils nur über eine lokale Sicht auf die Netzstruktur verfügen und *reaktiv* handeln.

Im Kontext des jeweiligen Anwendungsfeldes stellt der Routing Algorithmus stets einen optimalen Pfad innerhalb des Kommunikationsnetzes bereit. Dabei kann nach unterschiedlichen *Metriken* unterschieden werden, welche per Definition ein Bewertungskriterium repräsentieren:

- *Latenz / Verzögerung* - Dauer für die Übertragung eines Datenpaketes von der Quelle zur Senke.

- *Hop Count* - Anzahl der Zwischenstationen auf dem Weg von der Quelle zur Senke.
- *Zuverlässigkeit* - Stabilität des Routenpfades (Paketverlust- / Fehlerrate, Schwankungen in den Latenzen, etc.).
- *Datenrate / Bandbreite* - Maximaler und durchschnittlicher Datendurchsatz über einen Routenpfad.
- *Last* - derzeitiger Grad der Auslastung auf dem Kommunikationskanal
- *Kosten* - Anfallende abstrakte/reale Kosten bei einer Nutzung des Pfades - relevant bei gebührenpflichtigen Netzen (UMTS, GPRS, o.ä.)

Im Gegensatz zu traditionellen Routing-Verfahren, bspw. *RIP*⁸ [84], welches in statischen Netzwerktopologien Anwendung fanden, ist das Routing in mobilen Ad Hoc Netzwerken mit vielen neuen Problemen verbunden. In Folge der hohen Dynamik einzelner Knoten arbeiten diese klassischen Verfahren ineffizient oder versagen teilweise komplett. Die primäre Anforderung an Ad Hoc Routing-Protokolle sind somit, neben einer geringen Latenz und hohem Datendurchsatz, eine Optimierung für dynamisch wechselnde Netzwerktopologien. Die Berechnung stabiler Routenpfade und eine schnelle Reaktion auf sich ändernde Netzstrukturen steht im Vordergrund der Forschungen [70, 181].

Die Vielzahl unterschiedlicher Routingprotokolle lassen sich nach unterschiedlichen Eigenschaften klassifizieren:

Informationsgewinnung: *Reaktives* ↔ *Proaktives Routing*

Proaktive Routingverfahren pflegen zu jedem Zeitpunkt eine aktuelle Routingtabelle (*table-driven*), welche für jeden bekannten Knoten im Netzwerk einen gültigen Routenpfad bereitstellt. Folglich ist eine ständige Aktualisierung der Informationen erforderlich, was speziell bei dynamischen Netzstrukturen zu großen Problemen führt, da ein großer Protokolloverhead in Form von Statuspaketen erzeugt wird. Aus diesem Grund skalieren proaktive Verfahren relativ schlecht und arbeiten daher nur bei Netztopologien kleiner bis mittlerer Größe effizient. Dem hohen Aktualisierungsaufwand steht eine geringe Latenz bei der Kommunikation gegenüber, da jederzeit ein gültiger Pfad zur Verfügung steht.

Im Gegensatz dazu suchen *Reaktive Routingverfahren* gültige Routen zum Zielknoten erst im Zuge eines anstehenden Datentransfer, also erst "bei Bedarf" (*on demand*). Die Pfadsuche wird dabei oft durch *Flooding* Mechanismen realisiert, bei denen die Routinganfrage per *Broadcast* über das gesamte Netzwerk verbreitet wird. Auf die dabei entstehenden Probleme wird im weiteren Verlauf dieses Abschnittes näher eingegangen. Zusätzlich kommen häufig *Caching-Verfahren* zum Einsatz, welche gefundene Routen für einen bestimmten Zeitraum zwischenspeichern. Auf diese Art und Weise kann zum einen der Protokolloverhead reduziert werden und gleichzeitig sinkt die durchschnittliche Latenz für die Kommunikation in Netzwerk. Im allgemeinen skalieren reaktive Verfahren deutlich besser und sind somit auch in größeren Topologien einsetzbar. Nachteilig wirken

⁸Routing Information Protocol

3. Stand der Forschung

sich bei der Kommunikation primär die Latenzen aus, welche im Vergleich zu proaktiven Verfahren höher ausfallen.

Informationsverteilung: *Distanzvektor- ↔ Link-State-Verfahren*

Bei Link-State Routingverfahren verteilen die einzelnen Knoten aktuelle Verbindungsinformation ihrer direkten Nachbarschaft im Netzwerk. Auf dieser Basis ist es jedem Knoten möglich, die gesamte Netzwerktopologie zu rekonstruieren und somit optimale Routen für alle verfügbaren Kommunikationspartner zu berechnen. Die Anwendbarkeit auf mobile Ad Hoc Netzwerke ist nur bedingt möglich, da die Pflege der Routingtabelle viele Ressourcen benötigt und der Informationsaustausch eine hohe Netzlast verursacht.

Distanzvektor-Verfahren sind im Gegensatz dazu deutlich weniger komplex und basieren auf dem Konzept des lokalen Informationsaustausches. Die einzelnen Knoten speichern hierbei ausschließlich Informationen, über welchen Kommunikationspartner in der direkten Nachbarschaft die jeweiligen Zielknoten erreichbar sind und wieviele Hops benötigt werden um diesen zu erreichen. Diese *Distanzen* repräsentieren die *Güte* der jeweiligen Verbindung, die wiederum als Parameter in die Berechnung abstrakten Kostenwerte mit einfließt. Neben der Anzahl benötigter Hops sind auch andere Parameter für die Güte möglich, wie bspw. die gemessenen Verzögerungszeiten oder der Datendurchsatz, der die Länge der Paketwarteschlange als Messgröße nutzt. Die Informationen werden in Form einer Distanz-/Kostenmatrix abgelegt und an die direkte Nachbarschaft verteilt. Werden Distanzinformationen anderer Knoten empfangen, werden diese einfach auf die Distanzmatrix eingepflegt. Die aktualisierten Informationen verteilen die Knoten in regelmäßigen Abständen. Im Vergleich zu Link-State-Verfahren sind Distanzvektor-Algorithmen einfacher zu realisieren und nahezu wartungsfrei.

Zielauswahl: *Source-Routing ↔ Hop-by-Hop-Routing*

Bei der Nutzung von Source-Routing Protokollen wählt der Senderknoten einen vollständigen Routenpfad, welcher in die Header der Datenpakete gespeichert wird. Auf dem Weg des Datenpaketes von der Quelle zur Senke wird die gespeicherte Route durch schrittweise Auswertung in den einzelnen Hops abgearbeitet. Dabei muss sichergestellt werden, dass der im Header hinterlegte Routenpfad während der Übertragung nicht manipuliert wird.

Im Gegensatz dazu wird bei einem Hop-by-Hop Routing jeweils nur das nächste Ziel in Form des nächsten Hops ausgehandelt bzw. berechnet. Somit erfolgt auf jeder Zwischenstation zwischen Sender und Empfänger eine erneute Prüfung hinsichtlich eines optimalen Routenpfades. Dies ermöglicht mehr Flexibilität, da auf kurzfristige Störungen schneller reagiert werden kann. Andererseits erhöhen sich durch ein solches Verfahren die Latenzzeiten und der Protokolloverhead erheblich.

Hierarchie: *Flaches ↔ Hierarchisches Routing*

Hierarchisches Routing kann im Kontext mobiler Ad Hoc Netzwerke nur eingeschränkt angewendet werden, da zentrale Verwaltungsinstanzen auf den jeweiligen Hierarchie-

Ebenen vorausgesetzt werden. Nahezu alle klassischen Routingverfahren auf Basis drahtgebundener Topologien sind hier einzuordnen.

Betrachtet man Ad Hoc Strukturen in kleinen bis mittelgroßen Strukturen, existieren per Definition keine höheren Verwaltungsebenen. Somit befinden sich alle Knoten auf einer Hierarchie-Ebene und es kommen ausschließlich flache Routingverfahren zum Einsatz.

Routenselektion: *Single-Path* ↔ *Multi-Path-Routing*

Bei Single-Path Verfahren wird ausschließlich ein Pfad von der Quelle zur Senke ermittelt und für die Übertragung genutzt. Multi-Path-Protokolle ermöglichen das Speichern mehrerer, verschiedener Routenpfade zwischen zwei gewählten Kommunikationspartnern [69, 122, 124, 188]. Die Datenlast auf mehrere Routenpfade verteilt und im Zielknoten wieder zusammengefügt. So ist einerseits durch die geschaffene Redundanz eine erhöhte Ausfallsicherheit gegenüber Störungen und partiellen Netzausfällen gegeben. Bei Ausfall eines Pfades können die Daten ohne Verzögerungen auf einen alternativen Pfad umgeleitet werden. Zum anderen sinkt die durchschnittliche Übertragungszeit erheblich. Dem gegenüber steht ein erhöhter Aufwand für die Routenverwaltung und die Verarbeitung der Datenströme.

Adressierung: *Unicast-, Multicast-Routing, Broadcast-* sowie *Anycast-Mechanismen*

Diese Klassifizierung bezieht sich auf die Anzahl der Empfänger einer gesendeten Nachricht und die damit verbundenen Adressierungsmechanismen. Unicast-Verfahren adressieren als Ziel ausschließlich einen im Netzwerk eindeutigen Empfänger. Bei Multicast-Adressierungsmechanismen ist die Generierung von Gruppen möglich und folglich können mit einem ausgehenden Datenpaket mehrere Empfänger angesprochen werden. Broadcast-Pakete werden von allen Stationen empfangen und verarbeitet die sich in Sendereichweite befinden. Auf Anycast-Pakete antwortet aus einer Gruppe von potentiellen Empfängern derjenige, der die schnellstmögliche Antwort senden kann. Anycast wird daher häufig im Zuge einer Lastverteilung eingesetzt, die für den Nutzer völlig transparent erscheint. Um eine exklusive Kommunikation mit jedem Knoten einer Gruppe zu gewährleisten, verfügen diese meist über eine zusätzliche Unicast-Adresse.

Positiondaten: *topologiebasiertes Routing* ↔ *geographisches Routing*

Mit dem Fokus auf dynamischen Netzstrukturen sind *geobasierte Routingverfahren* [72] von besonderem Interesse. Grundlage hierfür ist die absolute Positionsbestimmung des aktuellen Standorts mit Hilfe von Trackingsystemen. Basierend auf diesen Positionsangaben können zusätzliche Informationen wie Geschwindigkeit und Beschleunigung errechnet werden und stehen anschließend den Routingalgorithmen für die Routenberechnung zur Verfügung.

Topologiebasierte Routingverfahren ermöglichen keine absolute Positionsbestimmung. Auf der Basis unterschiedlicher Lokalisierungsverfahren ist es jedoch möglich, zusätzliche relative Positionsinformationen zu generieren, welche Aufschluss über Entfernung und Richtung benachbarter Knoten geben. Mit Hilfe der so geschaffenen Wissensbasis

3. Stand der Forschung

kann die Wegfindung auch ohne absolute Geodaten deutlich verbessert werden.

Ausbreitung / Sicherheit: *Interne* ↔ *Externe* Routingprotokolle

Wichtig für das im Rahmen dieser Arbeit vorgestellte Konzept ist auch die Klassifizierung in *IGP*⁹ und *EGP*¹⁰ Verfahren. Die beiden Kategorien unterscheiden sich grundlegend beim Umgang mit autonomen Systemen, welche in diesem Zusammenhang einen Netzwerkverbund unter administrativer Kontrolle einer Instanz repräsentieren. Folglich leiten IGP innerhalb eines autonomen Systems / Netzwerktopologie und EGP-Verfahren zwischen unterschiedlichen autonomen Systemen.

Wissensbasis *Level-3* ↔ *Cross-Layer* Routingprotokolle

Herkömmliche Routingverfahren arbeiten ausschließlich auf Netzwerk-Ebene (Level-3) und verwenden die verfügbaren Netzwerkinformationen zur Routenberechnung / -optimierung. Im Gegensatz dazu nutzen Cross-Layer Ansätze vielfältige Informationen des jeweiligen Netzwerkknotens, um so die Wissensbasis zu erweitern [89, 193, 195]. Die Integration von Systeminformationen, wie bspw. der aktuelle Ladezustand, die Verbindungsqualität oder die Paketverlustrate, ermöglicht eine hochgradig angepasste Optimierung der Routenpfade. Die Metrik kann dabei gezielt Schwachstellen in der Topologie umgehen, welche auf Netzwerkebene nicht identifizierbar wären.

	Proaktiv	Reaktiv	Hybrid
Distanzvektor	<i>DSDV</i> (<i>Destination-Seq<u>u</u>enced Distance-<u>V</u>ector</i>)	<i>AODV</i> (<i>Ad-hoc On-Demand Distance <u>V</u>ector</i>)	-
Link State	<i>OLSR</i> (<i>Optimized <u>L</u>ink <u>S</u>tate <u>R</u>outing</i>)	-	-
andere	<i>LRR</i> (<i>Link <u>R</u>eversal <u>R</u>outing</i>)	<i>DSR</i> (<i>Dyn<u>a</u>mic <u>S</u>ource <u>R</u>outing</i>)	<i>ZRP</i> (<i>Zone <u>R</u>outing <u>P</u>rotocol</i>)

Tabelle 3.1.: Übersicht aktueller Routingverfahren

Wichtige Vertreter im Bereich proaktiver Routingverfahren sind *DSDV* [134], *OLSR* [46] oder *LRR* (Tabelle 3.1). Die bekanntesten reaktiven Routingverfahren im Bereich mobiler Ad Hoc Netzwerke sind dabei *DSR* [23] und *AODV* [42], welche in unterschiedlichsten, experimentellen Umgebungen zur Anwendung kamen. Abbildung 3.4 vergleicht einige bekannte Routingverfahren im Bereich von MANETs hinsichtlich verschiedener Parameter wie Skalierbarkeit, Komplexität oder Protokolloverhead.

Das *ZRP*¹¹ [78] nimmt bei dieser Einordnung eine Sonderstellung ein. Es stellt ein hybrides Routingverfahren dar, bei dem jeder Knoten eine definierte Zone verwaltet. Innerhalb dieser

⁹Interior Gateway Protocol

¹⁰Exterior Gateway Protocol

¹¹Zone Routing Protocol

n Hop Nachbarschaft wird ein proaktives Routing genutzt und somit eine aktuelle Routingtabelle gepflegt. Außerhalb der Zone wird ein besser skalierendes reaktives Routing angewendet, wodurch sowohl die Netzlast als auch der administrative Aufwand in den einzelnen Knoten reduziert wird.

*TORA*¹² [133], welcher zu den *LRR* Algorithmen zählt, nimmt in diesem Zusammenhang eine Sonderstellung ein. In der Literatur wird er sowohl im Bereich Routing als auch im Bereich der Topologiekontrolle/-optimierung eingeordnet. Um den Protokolloverhead zu minimieren, wird hier ein nicht optimaler Routenpfad als Ergebnis der Berechnungen einkalkuliert.

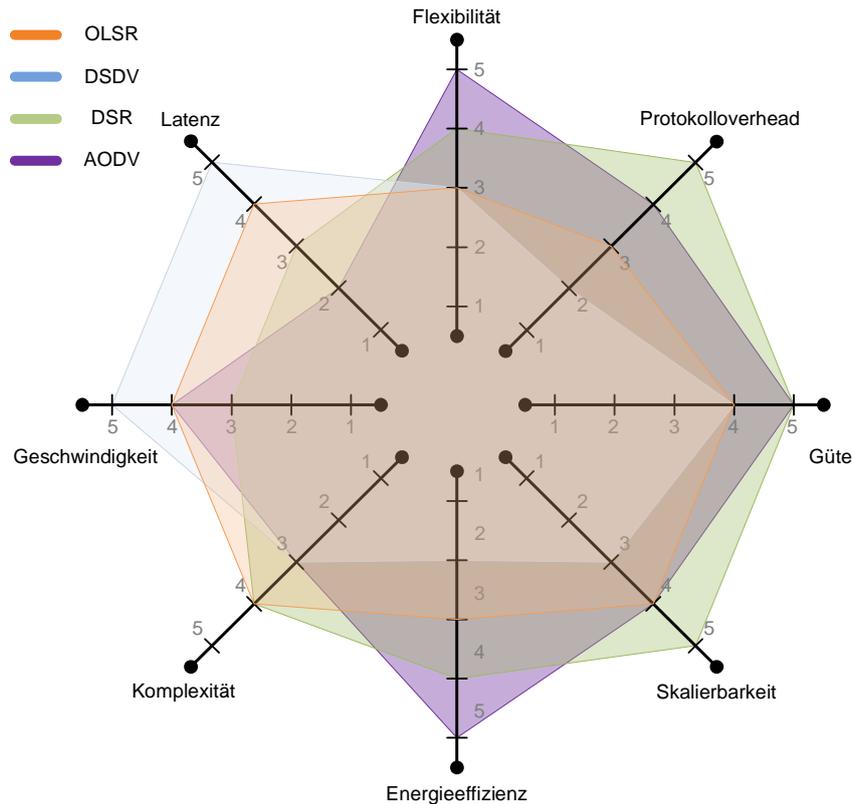


Abbildung 3.4.: Einige wichtige Routingverfahren im direkten Vergleich. Die anwendungsspezifischen Vor- und Nachteile hinsichtlich der verschiedenen Parameter sind deutlich erkennbar. Somit werden je nach Einsatzbereich unterschiedliche Verfahren präferiert.

Bekanntester Vertreter der externen Routing-Protokolle ist das *BGP*¹³ [149], welches sich stark an Distanzvektorverfahren anlehnt. Es gehört zu den Distanzvektor-Protokollen und gewährleistet eine schleifenfreie Pfadauswahl. Leider ist mit dem Verfahren keine Lastverteilung (*Load Balancing*) möglich und die max. Datenrate der einzelnen Verbindung spielt für die Pfadauswahl nur eine untergeordnete Rolle. Wie bei vielen anderen Vertretern werden auch bei BGP Sicherheitsaspekte vernachlässigt.

Einer der einfachsten und am häufigsten verwendeten IGP-Verfahren ist das bereits erwähnte

¹²Temporally Ordered Routing Algorithm

¹³Border Gateway Protocol

3. Stand der Forschung

Distanzvektorverfahren *RIP*, welches 1969 als Teil des dezentralen Forschungsnetzes *ARPA-NET*¹⁴ [52], dessen Nachfolger das heutige Internet repräsentiert, entwickelt wurde. Es nutzt die Anzahl der nötigen Hops von der Quelle zur Senke als alleinige, triviale Metrik. Größter Nachteil von *RIP* besteht in der Tatsache, dass eine schleifenfreie Pfadsuche nicht gewährleistet werden kann und es somit zu Problemen bei der Routenberechnung kommen kann. *IGRP*¹⁵ ist ein weiterer bekannter Vertreter der internen Routing-Protokolle und gehört ebenfalls zu den Distanzvektorverfahren. Es skaliert deutlich besser als *RIP* da es die Beschränkung von maximal 15 Hops umgeht, welche in der *RIP*-Metrik den beschränkenden Faktor darstellt. *IGRP* und auch die Weiterentwicklung *EIGRP (Enhanced IGRP)* wählen den optimalen Pfad basierend auf Latenz, Bandbreite, aktueller Last sowie der Zuverlässigkeit der gefundenen Route. Es sei dabei erwähnt, dass *EIGRP* als hybrides Routing-Protokoll zu betrachten ist, da dieses Distanzvektor-Verfahren auch Link-State Mechanismen in sich vereint. Auch *OSPF*¹⁶ gehört zu den schleifenfreien IGP mit guter Skalierbarkeit und geringem Ressourcenverbrauch.

Neben den zuvor genannten Klassifizierungen existieren eine Vielzahl weiterer Routingverfahren mit anwendungsspezifischen Vor- und Nachteilen. Bekannte Vertreter sind hier *GPSR*¹⁷ [99] und *LAR*¹⁸. Das *DREAM-Verfahren*¹⁹ nutzt diese Informationen auf eine besondere Art und Weise. Hier fließen die Distanzinformationen in das Abtastintervall für die Erreichbarkeit der einzelnen Knoten mit ein. Das Verfahren geht davon aus, dass Knoten, die sich mit hoher Geschwindigkeit fortbewegen, ihre Position häufiger mitteilen sollten als Knoten mit geringerer Dynamik. In Folge dessen prüft *DREAM* Knoten in unmittelbarer Nachbarschaft in einem kürzeren Intervall und kann dadurch auf Änderungen entsprechend schnell reagieren und verfügbare Routenpfade anpassen. Entfernte Knoten werden im Gegensatz dazu mit einer geringeren Abtastrate geprüft, was die Netzlast und den Energiebedarf relativiert.

Im Bereich drahtloser Sensornetze sollen zwei weitere Verfahren kurz erläutert werden. *SPIN*²⁰ [106] nutzt eine effiziente Modifikation des klassischen *Flooding*s, indem die eigenen Routinginformationen zunächst angeboten werden. Benötigen andere Knoten die Daten des Senders, so bestätigt er die Anfrage und fordert die Daten somit explizit an. Erst im Anschluss übermittelt der Sender daraufhin die verfügbaren Informationen. Durch dieses initiale, invertierte Handshake-Verfahren sinkt der Protokolloverhead und die Effizienz steigt. Desweiteren behebt *SPIN* dadurch zwei konzeptionelle Nachteile des *Flooding*s: Implosion und Overlap, welche in Abbildung 3.5 verdeutlicht werden. Das in der Literatur häufig genutzte *Directed Diffusion Verfahren* [94] basiert ebenfalls auf *SPIN*.

¹⁴Advanced Research Projects Agency Network

¹⁵Interior Gateway Routing Protocol

¹⁶Open Shortest Path First

¹⁷Greedy Perimeter Stateless Routing for Wireless Networks

¹⁸Location-Aided Routing

¹⁹Distance Routing Effect Algorithm for Mobility

²⁰Sensor Protocols for Information via Negotiation

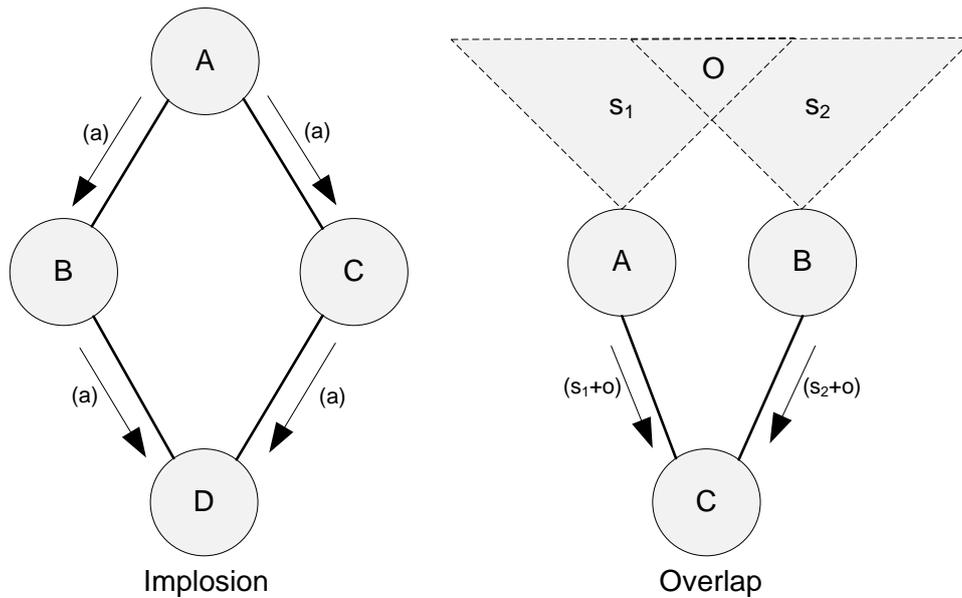


Abbildung 3.5.: Zwei wesentliche Probleme im Bereich Routing: links *Implosion* - die Informationen von A erreichen D mehrfach über B und C; rechts *Overlap* - bei der Exploration der Topologie überschneiden sich die durchsuchten Bereiche. Knoten C erhält Informationen über den Bereich O doppelt.

Das *LEACH-Verfahren*²¹ [85] strukturiert die Netzwerktopologie in hierarchische Ebenen und basiert somit auf der Clusterbildung. Dies ermöglicht eine gleichmäßige Lastverteilung und in Folge dessen eine optimierte Nutzung der verfügbaren Energieressourcen. Da in jedem Cluster ein zentraler Verwaltungsknoten benötigt wird, liegt hier auch ein zentraler Schwachpunkt clusterbasierter Verfahren. Bei einem Ausfall des *Cluster-Heads* kann es dabei zu erheblichen Störungen kommen.

Für das im Rahmen dieser Arbeit entworfene Konzept zur funkstandardübergreifenden Kommunikation in mobilen Ad Hoc Netzwerken wurde ein eigenständiges Routingverfahren auf Basis der vorgestellten Strategien entworfen. Da der Fokus auf einer guten Skalierbarkeit innerhalb hochdynamischer Ad Hoc Netzwerke liegt, wurde ein reaktives Verfahren konzipiert und implementiert. Auf das Verfahren wird im folgenden Kapitel (Abschnitt 4.4.4) detailliert eingegangen.

Broadcast Storms

Broadcast Storms resultieren vorrangig aus Flooding-Mechanismen unterschiedlicher Routingprotokolle und Topologieoptimierungsverfahren [127]. Dabei streuen die einzelnen Knoten ihre Anfragen mit voller Sendeleistung in das Netzwerk um möglichst viele benachbarte Stationen zu erreichen. Ohne zusätzliche Gegenmaßnahmen leitet jeder Empfänger die Broadcast-Pakete weiter. Die Informationen verteilen sich somit nach dem Schneeball-Prinzip und fluten in Folge

²¹Low-Energy Adaptive Clustering Hierarchy

3. Stand der Forschung

dessen das gesamte Netzwerk. Daraus folgt eine massive Netzlast sowie ein erhöhtes Interferenzniveau, welches die Kommunikation im Netzwerk negativ beeinflusst.

Um das Entstehen solcher Situation zu vermeiden, wurden unterschiedliche Strategien entwickelt. Diese realisieren im Wesentlichen effizientere Broadcasting-Mechanismen für die Informationsverteilung [88, 182]. Zwei einfache Vertreter sind dabei *Jitter / Random Assessment Delay* und ein *Counter-based Retransmission Scheme*, welche auch im *EBCR* Routingverfahren umgesetzt werden. *Random Delay* versucht dabei das Auftreten von Interferenzen durch die bewusste Verzögerung einzelner Datenpakete zu minimieren. Während der Bearbeitung und Weiterleitung von Broadcast-Paketen in den Knoten werden dazu künstliche, zufällige Wartezeiten erzeugt. Da somit die Anzahl paralleler Übertragungen im Netzwerk sinkt, wird das Auftreten von Interferenzen deutlich reduziert. Gleichzeitig resultieren daraus aber auch höhere Latenzen während der Kommunikation.

Counter-based retransmission Scheme zählt die Anzahl eingehender Broadcast-Anfragen und verweigert beim Überschreiten eines vordefinierten Wertes das "re-broadcasten" der Anfrage. Die Methode ist ebenfalls sehr effizient, muss allerdings an das jeweilige Anwendungsszenario angepasst werden.

Neben den aufgeführten Methoden kommen auch probabilistische Verfahren zur Anwendung. Diese verwerfen mit einer festgelegten Wahrscheinlichkeit eingehende Broadcast-Anfragen. Da wahrscheinlichkeitsbasierte Verfahren durch ihr nichtdeterministisches Verhalten schwer kontrollierbar sind und die Ergebnisse somit starke Schwankungen aufweisen, werden diese im Rahmen dieser Arbeit nicht betrachtet.

3.1.3. Drahtlose Sensornetzwerke

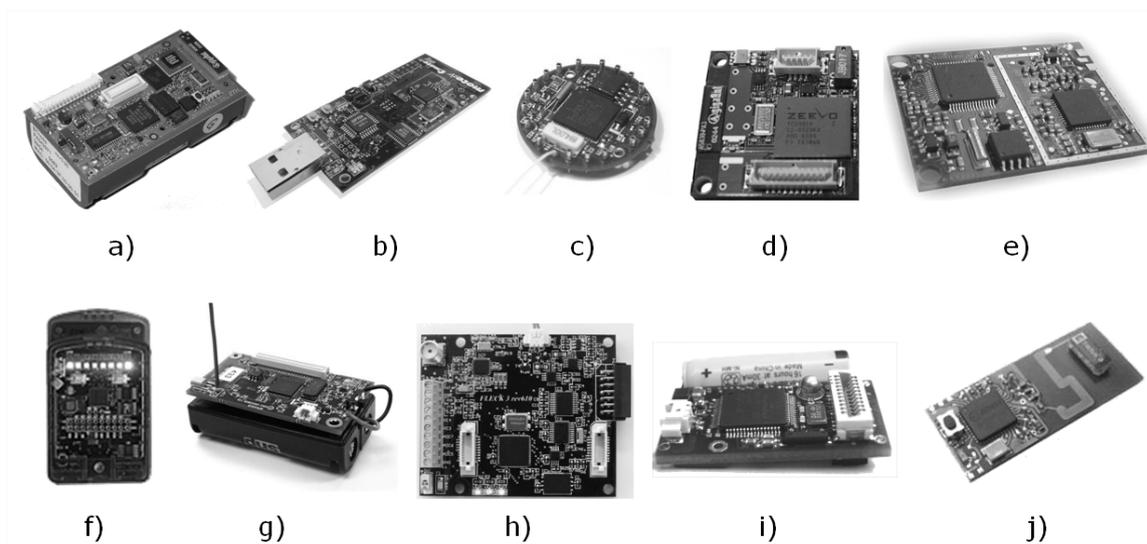


Abbildung 3.6.: Eine Auswahl verschiedener Sensor-Hardwareplattformen: a) BTNode, b) Tmote Sky, c) Mica2Dot, d) Imote, e) TinyNode 584, f) Sun SPOT, g) MicaZ, h) Fleck3, i) Particles, j) WeBee.

Drahtlose Sensornetzwerke (WSN²²) repräsentieren, wie bereits zu Beginn dieses Kapitels erwähnt, eine Teilmenge mobiler Ad Hoc Netzwerke. WSN besitzen meist eine Topologie mit sehr vielen Knoten und hoher Dichte. Analog zu MANETs soll sich die Topologie selbstständig mit den genannten *Self-X* Eigenschaften organisieren [165, 171, 187]. Da WSN mit den verfügbaren Energieressourcen meist sehr lange autark arbeiten müssen, sind hier die Anforderungen an den Energiebedarf besonders hoch priorisiert. Aufgrund dessen wird die Komplexität der Hardware auf ein Minimum reduziert und auf die verfügbare Sensorik sowie das Anwendungsfeld abgestimmt. Abbildung 3.6 zeigt eine Auswahl verfügbarer Hardwareplattformen für den experimentellen Einsatz. Die dargestellten Sensorknoten unterscheiden sich neben der verwendeten Funktechnologie primär in der Sensorik, der Rechenleistung und in den Abmessungen.

Anwendungsgebiete

Drahtlose Sensornetzwerke finden in vielen Bereichen Anwendung. Typische Szenarien sind hier bspw. die Gebäude- und Objektüberwachung oder der Katastrophenschutz. Hier kann mit Hilfe von WSN präventiv ein großflächiges Areal in Hinsicht auf bestimmte Ereignisse überwacht werden. Im Falle einer bereits bestehenden Katastrophe bieten sie die Möglichkeit, schnell und effizient Informationen verschiedenster Art aus dem betroffenen Gebiet zu sammeln.

Mit der zunehmenden Miniaturisierung elektronischer Geräte liegt die Vision in *Smart Dust* oder *Sensordust*, bei dem die einzelnen Sensorknoten die Größe von Staubkörnern besitzen. Damit wären die Möglichkeiten sowohl im militärischen als auch im zivilen Bereich nahezu unbegrenzt.

Methoden zur Energieoptimierung

Neben der Gelände und Objektüberwachung ist eine zentrale Aufgabe von WSNs die Ermittlung globaler Ereignisse auf Basis unterschiedlichster lokaler Informationen. In diesem Zusammenhang sind drei wesentliche Parameter für die Güte ausschlaggebend: Energiebedarf, Latenz und Robustheit [57, 126, 194]. Um die Qualität der Sensordaten zu erhöhen, kann eine *Datenaggregation* zur Anwendung kommen. Hierbei findet die Datenvorverarbeitung in der Nähe des Beobachtungsortes, also noch auf den Sensorknoten statt. Somit können bestimmte Parameter und Resultate gezielt gefiltert werden, um die Netzlast zu reduzieren. Für die Verfahren sind allerdings Ressourcen in Form von Rechenleistung und Speicher nötig, wodurch ein Einsatz nur bedingt möglich ist.

Da im Gegensatz zu MANETs die gesamten Informationen in WSNs meist zu einem zentralen Punkt geleitet werden müssen, wurden entsprechend angepasste Algorithmen für den Bereich Routing und Topologieoptimierung entwickelt [73]. In Hinsicht auf das Routing ist der

²²Wireless Sensor Networks

3. Stand der Forschung

kooperative Aspekt dabei besonders interessant. Ziel ist es, den Energieverbrauch für die Kommunikation gleichmäßig auf die gesamte Topologie zu verteilen. Dadurch wird die Laufzeit aller Knoten maximiert und die Erreichbarkeit sichergestellt. Gleichzeitig bleibt somit die Güte der gewonnenen Informationen aus dem Sensornetz konstant. Aufgrund der genannten Vorteile wurde ein solches Konzept auch im Rahmen dieser Arbeit innerhalb des *EBCR* Routingverfahren integriert.

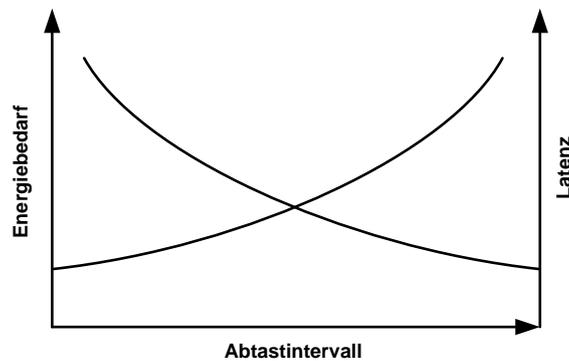


Abbildung 3.7.: Drahtlose Sensornetzwerke: Abhängigkeit von Abtastrate, Energiebedarf und Latenz.

Auf Medienzugriffsebene wurden Verfahren wie *S-MAC*²³, *T-MAC*²⁴ oder *TRAMA*²⁵ [126, 144] entwickelt. *S-MAC* und *T-MAC* unterscheiden dabei aktive und passive Betriebsphasen. Während in den aktiven Phasen die Nutzdaten übermittelt werden, nutzen die Knoten die passiven Phasen um Energie zu sparen und gehen dazu in einen Ruhezustand. Um dies möglichst effizient einsetzen zu können, müssen die Zeitpläne der einzelnen Knoten entsprechend synchronisiert werden. Ziel von *TRAMA* ist es, eine kollisionsfreie Kommunikation zu ermöglichen und dabei die Ressource Zeit dynamisch je nach Anforderung der einzelnen Knoten zu verteilen [126].

Die vorgestellten Verfahren bilden nur eine kleine Teilmenge im Bereich der Optimierung drahtloser Sensornetzwerke. Sowohl auf Applikations- als auch auf Vermittlungs- und Sicherungsebene existiert eine Vielzahl weiterer Ansätze mit dem Ziel, einen Kompromiss zwischen Energiebedarf, Latenz und Abtastrate zu finden (Abbildung 3.7).

3.2. Kommunikationstechnologien der nächsten Generation

Die folgenden Abschnitte befassen sich mit zukünftigen Kommunikationsnetzen und deren technische Grundlagen. Die Auswahl der hier angesprochenen Themen soll sich dabei auf für diese Arbeit relevante Themen beschränken. Unter dem Forschungsschwerpunkt *xG* (*Next*

²³Sensor-Media Access Control

²⁴Timeout-Media Access Control

²⁵Traffic-Adaptive Medium Access Protocol

Generation) entsteht eine Vielzahl wissenschaftlicher Arbeiten. Probleme in Zusammenhang mit dem Übertragungsmedium Luft treten in Folge der wenigen freien Frequenzbänder mehr und mehr in den Vordergrund. Gleichzeitig müssen die erhöhten Anforderungen moderner, multimedialer Anwendungen in Bezug auf die verfügbare Übertragungsrate umgesetzt werden. Primäre Anforderungen und mögliche Lösungskonzepte lassen sich wie folgt klassifizieren und werden in den angegebenen Abschnitten näher erörtert:

- *Konnektivität / Verfügbarkeit*
 - Software Defined Radio (Abschnitt 3.2.1)
 - Cognitive Radio (Abschnitt 3.2.2)
 - Ambient Networking (Abschnitt 3.2.5))
- *Energieeffizienz / Interferenzminimierung*
 - Smart Antennas (Abschnitt 3.2.3)
 - Ultrabreitband-Technologien (Abschnitt 3.2.4)
 - Ambient Networking (Abschnitt 3.2.5))
- *Steigerung der Übertragungsrate*
 - Smart Antennas (Abschnitt 3.2.3)
 - Ultrabreitband-Technologien (Abschnitt 3.2.4)
 - Cognitive Radio (Abschnitt 3.2.2))

Auf die genannten Lösungskonzepte soll nun näher eingegangen werden:

3.2.1. Software Defined Radio

*SDR*²⁶ beschreibt ein Konzept, in dem die komplette Signalverarbeitung einer drahtlosen Kommunikation über das Medium Luft durch *Softwaremodule und programmierbare Hardware* realisiert wird [96].

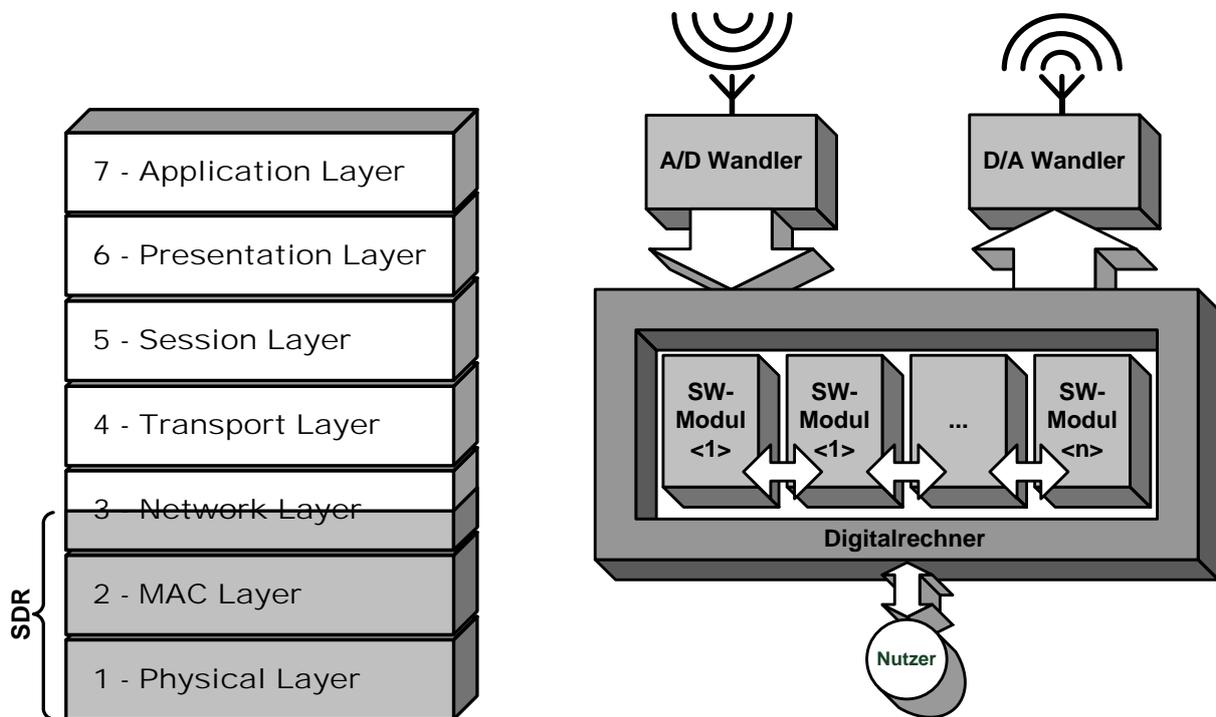
Funktionsweise

Die konzeptionelle Idee von *SDR* basiert auf der Portierung von hardwareseitigen Teilfunktionalitäten der Signalverarbeitung in modulare Softwarekomponenten [60, 96, 161]. Diese Portierung umfasst mehrere Ebenen der jeweiligen Protokollstacks (Abbildung 3.8a). Zunächst ist die *Bitübertragungsschicht (Physical Layer, Layer 1)* des ISO/OSI Modells relevant, welche die physikalische Übertragung der einzelnen Bits über das Transportmedium Luft realisiert. Weiterhin muss auch die *Sicherungsschicht (Data Link Layer, Layer 2)* betrachtet werden, in der unter anderem der Zugriff auf das Übertragungsmedium geregelt ist. In einigen Fällen kommen weitere Teilfunktionalitäten der *Vermittlungsschicht (Network Layer, Layer 3)* hinzu, die im Wesentlichen für das Management der Datenkanäle zwischen zwei Verbindungspartnern verantwortlich ist [154]. Auf *SDR* basierende Systeme können somit auf Änderungen in den

²⁶Software Defined Radio

3. Stand der Forschung

unteren Kommunikationsprotokollen, welche bisher als statische Hardwarebausteine realisiert wurden, flexibel und kosteneffizient reagieren, indem die relevanten Softwaremodule adaptiert werden.



(a) OSI Modell. Die für SDR relevanten Schichten sind grau hinterlegt.

(b) Die Abbildung beschreibt schematisch ein SDR-System. Eingehende Daten werden nach der Analog-Digital-Wandlung direkt in einem Digitalrechner weiterverarbeitet. Das Senden von Daten geschieht nach dem gleichen Schema.

Abbildung 3.8.: a) OSI Referenzmodell, b) schematische Darstellung eines SDR-Systems.

Für die Verarbeitung dieser Teilgebiete kommen unter anderem leistungsfähige digitale *Signalprozessoren*, moderne Allzweckrechner sowie spezielle *FPGAs*²⁷ zum Einsatz. Alle höheren Protokollebenen sind ebenfalls als modulare Softwarekomponenten realisiert, beziehen sich aber nicht auf das Forschungsgebiet von SDR. Ein solches System kann somit als eine rekonfigurierbare Kommunikationseinheit betrachtet werden, welche durch angepasste Softwaremodule an den jeweils verwendeten Funkstandard angepasst wird [75].

Die dafür zugrunde liegende Softwarearchitektur spielt eine zentrale Rolle für die Effektivität und Flexibilität des SDR-Systems [83, 95, 97]. Es sind hierbei zwei Softwaremodelle für SDR zu unterscheiden. Im *parametergesteuerten SDR* [191] werden die Algorithmen für die einzelnen Teilbereiche der Signalverarbeitung so generell wie möglich gehalten. Die jeweiligen Funkstandards werden dann lediglich durch einen entsprechenden Parametersatz in diesen Algorithmen realisiert. Im Gegensatz dazu wird beim *modularen SDR* [150] auf eine möglichst kleine Kapselung der Teilfunktionalitäten Wert gelegt. Diese sind spezifisch an die verschie-

²⁷Field Programmable Gate Array

denen Funkstandards angepasst und somit laufzeitoptimiert. Dadurch kann ein Baukastensystem realisiert werden, aus dem sich je nach Anwendung eine logische Reihenfolge optimierter Softwarekomponenten entnommen wird.

Anwendungsszenarien

Mit dem heutigen Stand der Technik sind SDR Lösungen nur teilweise realisierbar, da ein ideales SDR-System, wie es schematisch in Abbildung 3.8b dargestellt ist, noch nicht umsetzbar wurde. Grund hierfür ist die verfügbare Elektronik, deren Geschwindigkeit noch nicht ausreicht, um analoge Funksignale im Bereich von bis zu 2 GHz in hinreichend hoher Auflösung zu digitalisieren. Für eine korrekte Rekonstruktion des Datensignals müssen dabei die analoge Daten mit einer sehr hohen Frequenz abgetastet werden ($\text{minimale Abtastrate} \geq 2 * \text{Grenzfrequenz}$: siehe auch *Nyquist-Theorem* [45, 130]).

Aktuelle Projekte wie das *GNU Radio* [74] oder das *IBMS² - Projekt²⁸* des Forschungsschwerpunktes *Universelle Nutzung von Kommunikationsnetzen für künftige Mobilfunkgenerationen - HyperNET* - [39, 40] konnten auf der Basis von SDR Funktionalitäten aus dem Bereich der Radio- und Mobilfunktechnik umsetzen. Kernpunkt ist weiterhin die Adaption von SDR auf mobile, kompakte Endgeräte.

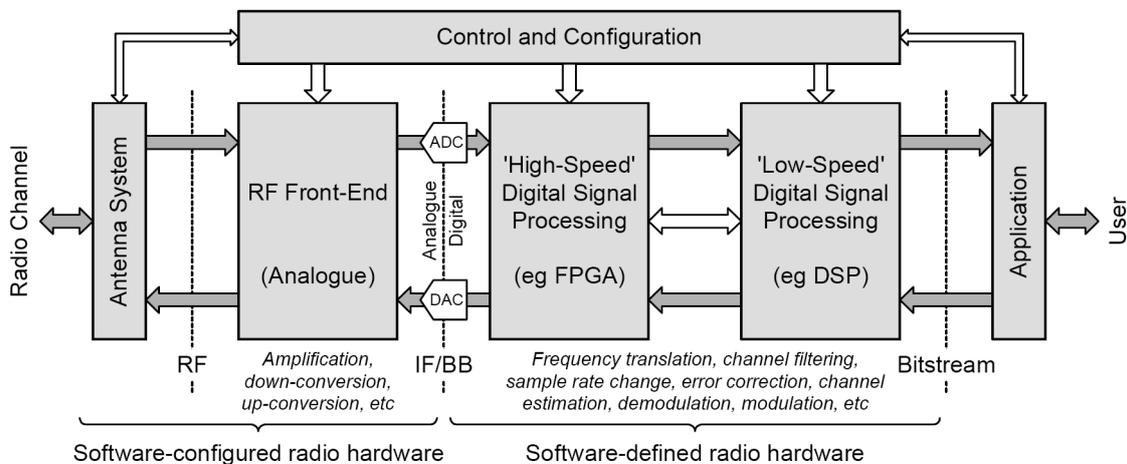


Abbildung 3.9.: Eine vereinfachte Darstellung einer SDR-Plattform, wie sie mit verfügbaren Technologien realisiert werden könnte [142].

Aufgrund der konzeptionellen Vorteile von SDR liegen die primären Anwendungsszenarien im militärischen Bereich und auf dem Gebiet des zivilen Mobilfunks. Hier spielt der einschränkende Faktor der hohen Leistungsaufnahme eine eher untergeordnete Rolle. Bspw. wäre der Einsatz von SDR in den Basisstationen zellulärer Mobilfunknetze in hohem Maße effizient und die Betriebskosten könnten deutlich reduziert werden, da die Betreiber auf technologische Änderungen innerhalb kürzester Zeit durch entsprechende Softwaremodule reagieren können. Auch die Verschmelzung bestehender Funknetze, welche auf völlig unterschiedlichen Protokollen

²⁸Integrierte Bandbreiteneffiziente Mobile Software-Radio System

3. Stand der Forschung

und Frequenzbändern betrieben werden, wäre durch eine solche Technologie mit wenig Aufwand realisierbar, da durch SDR die nötige Protokollkonvertierung in Echtzeit erfolgen würde.

Im militärischen Bereich wurden große Projekte finanziert, um die Kommunikationsnetze der verschiedenen Truppenteile zu vernetzen und gegnerische Funkkommunikation noch effizienter auszuwerten. Finnland entwickelte mit dem *FSRP*²⁹ [140, 184] einen wichtigen Vertreter. Mit dem *JTRS*³⁰ [54] plant die USA ein taktisches Kommunikationsnetz der nächsten Generation auf Basis von SDR, welches die Vielzahl aktuell eingesetzter Kommunikationstechnologien vereinen soll. Mit Hilfe eines *einheitlichen Entwicklungsframeworks (SCA)*³¹ sollen hier den Entwicklern Richtlinien zur Verfügung gestellt werden, wie die einzelnen SDR-Elemente von Hard- und Software konfliktfrei operieren können.

Durch SDR wird somit ein Höchstmaß an Flexibilität und Modularität erzielt. Aufgrund der modularen Erweiterbarkeit kann SDR die Entwicklungskosten für neue Anwendungsgebiete im Bereich mobiler Kommunikationslösungen signifikant reduzieren, da Kosten für Anpassungen lediglich im Softwarebereich anfallen und teure Hardwareentwicklungen weitestgehend vermieden werden können. Der wesentliche Nachteil dieser Technologie bleibt für absehbare Zeit die hohe Leistungsaufnahme. Für die Umsetzung von SDR-Systemen sind leistungsfähige Hardwarekomponenten nötig, die über die entsprechende Rechenleistung für die hochkomplexe Signalverarbeitung verfügen. Aus diesem Grund sind auch die anfallenden Kosten für die Anschaffung und den Betrieb relativ hoch. Bedingt durch den hohen Energiebedarf eines SDR-Systems ist der Einsatz in kompakten, mobilen Endgeräten in naher Zukunft nur schwer realisierbar, da heutige Akkumulatoren die benötigte Energiedichte nicht zur Verfügung stellen können. Der im Rahmen dieser Arbeit vorgestellte Ansatz für eine funkstandardübergreifende Kommunikation in MANET greift diese Problemstellung auf und bietet eine vielversprechende Alternative.

3.2.2. Cognitive Radio

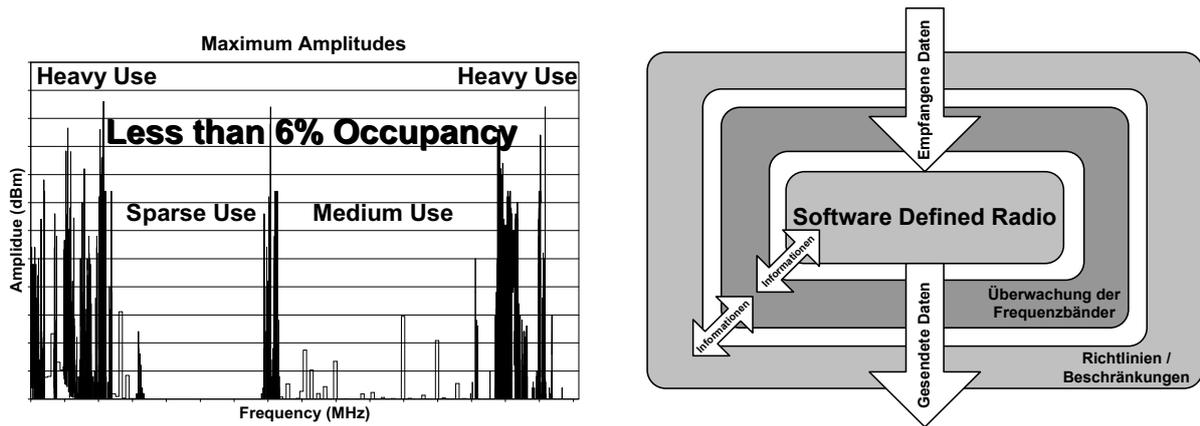
Cognitive Radio (CR) kann als die nächste Entwicklungsstufe im Bereich der *rekonfigurierbaren, adaptiven Funksysteme* betrachtet werden. Da in der Literatur eine Vielzahl unterschiedlicher Definition für *CR* existieren, kommt im Rahmen dieser Arbeit folgende Definition nach [143] zum Einsatz:

„Ein Cognitive Radio repräsentiert einen Funk-Transceiver, der durch sein Design eine Abtastung des verfügbaren Frequenzbandes ermöglicht. Freie bzw. wenig genutzte Frequenzbereiche werden automatisch erkannt und für die Datenübertragung genutzt. Nicht benötigte Ressourcen werden unmittelbar nach der Verwendung wieder freigegeben und stehen somit anderen Nutzern oder Diensten zur Verfügung, wodurch interferenzbedingte Störungen im Netzwerk minimiert werden.“

²⁹Finnish Software Radio Program

³⁰Joint Tactical Radio System

³¹Software Communications Architecture



(a) Derzeitige Nutzung verfügbaren Frequenzspektrums zur Datenübertragung. Nur sehr kleine Bereiche werden effizient ausgelastet [53].

(b) Schematische Darstellung eines "Cognitive Radio" Systems. Die Kernfunktionalität wird durch ein "Software Defined Radio" zur Verfügung gestellt. Mit Hilfe zusätzlicher Ebenen werden die verfügbaren Frequenzbänder überwacht und nach Bedarf genutzt.

Abbildung 3.10.: a) Ausnutzung der verfügbaren Frequenzspektrums,
 b) Cognitive Radio System als eine umfassende Evolutionsstufe von SDR.

CR stellt dabei ein erweitertes SDR dar (Abbildung 3.10(b), welches zusätzlich über die Fähigkeit verfügt, seine Umgebung zu analysieren, um somit selbstständig und dynamisch auf Änderungen in der Konfiguration seiner Umwelt zu reagieren [143]. Da Cognitive Radio konzeptionell bedingt das genutzte Frequenzspektrum nur als einen variablen Kommunikationsparameter definiert, erlaubt es den Endgeräten die dynamische Anpassung der verwendeten Übertragungsspektren, um somit jederzeit eine bestmögliche Kommunikation zu gewährleisten [21, 64].

„Ein CR nutzt eine Intelligente Signalverarbeitung³² auf dem Physical Layer eines drahtlosen Kommunikationssystems. Es kombiniert folglich die ISP mit dem SDR.“

Ein solches System ist in der Lage, völlig autonom Entscheidungen über die Wahl des zu nutzenden Frequenzspektrums und des Kommunikationsstandards zu treffen. Dabei spielen Faktoren wie funktionale Anforderungen an den Kommunikationskanal oder Aussagen über die Übertragungsqualität im jeweiligen Frequenzband eine wichtige Rolle. Einhergehend mit der begrenzten Anzahl frei nutzbarer Frequenzbänder und den regionalen Unterschieden in den Nutzungsrestriktionen wird ein Großteil der verfügbaren Frequenzen nur ineffizient genutzt [173]. Das Problem wird in Abbildung 3.10(a) dargestellt, welche die derzeitige Situation verdeutlicht. Weniger als 10% des nutzbaren Frequenzspektrums werden effizient ausgelastet.

Zu unterscheiden sind zwei Varianten des Cognitive Radio, welche im Umfang der Adaptionsmöglichkeiten variieren. *Full Cognitive Radio* wird in der Literatur auch unter dem Namen *Mitola Radio* [93] behandelt. Hier werden alle erdenklichen Parameter bei der Kommunikationsstrategie dynamisch angepasst. Somit wird hier der gesamte Protokollstack über alle Ebenen betrachtet. Im Gegensatz dazu bietet das *Spectrum Sensing Cognitive Radio* lediglich

³²Intelligent Signal Processing - ISP

3. Stand der Forschung

eine Teilfunktionalität, bei der nur das genutzte Frequenzspektrum adaptiert wird. Beide Forschungsansätze brechen mit den derzeitigen gesetzlichen Bestimmungen hinsichtlich der eingeschränkten Verwendung bestimmter Frequenzbereiche. CR setzt als Ressource einen breiten, frei verfügbaren Frequenzbereich voraus, welcher durch die einzelnen Systeme partiell allokiert und wieder freigegeben werden kann.

Den wichtigsten Aspekt im Konzept des Cognitive Radio repräsentiert also das *Spectrum Sensing*, mit dessen Hilfe ungenutzte Frequenzbereiche erkannt und Interferenzen minimiert werden können. Die Funktionalitäten des *Advanced Spectrum Managements* umfassen die Bereiche *Spectrum Analysis* und *Spectrum Decision*. Hier kann durch eine gezielte *Spectrum Reallocation* die Reservierung und Freigabe einzelner lizenzierter Frequenzbereiche für spezielle Dienste oder Nutzergruppen erfolgen. *Cognitive Radio* ist somit in der Lage, QoS^{33} Funktionalitäten über die Menge aller verfügbarer Frequenzbänder zur Verfügung zu stellen [21, 43, 107].

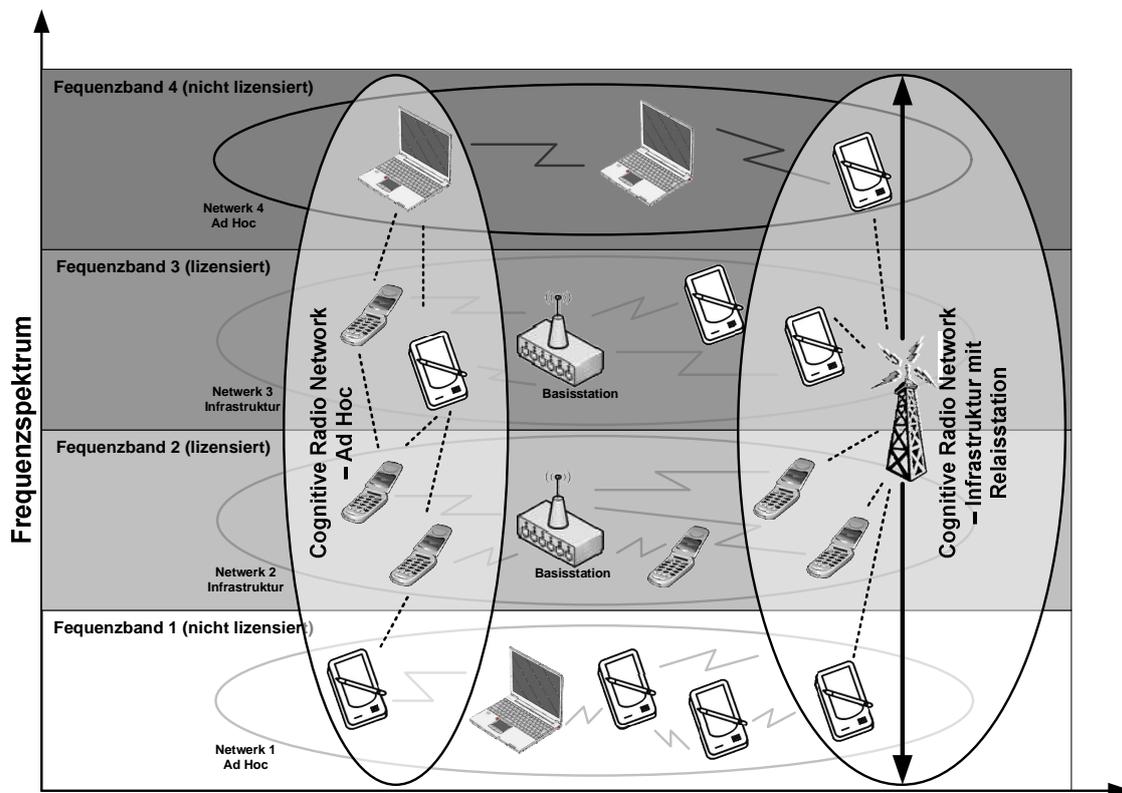


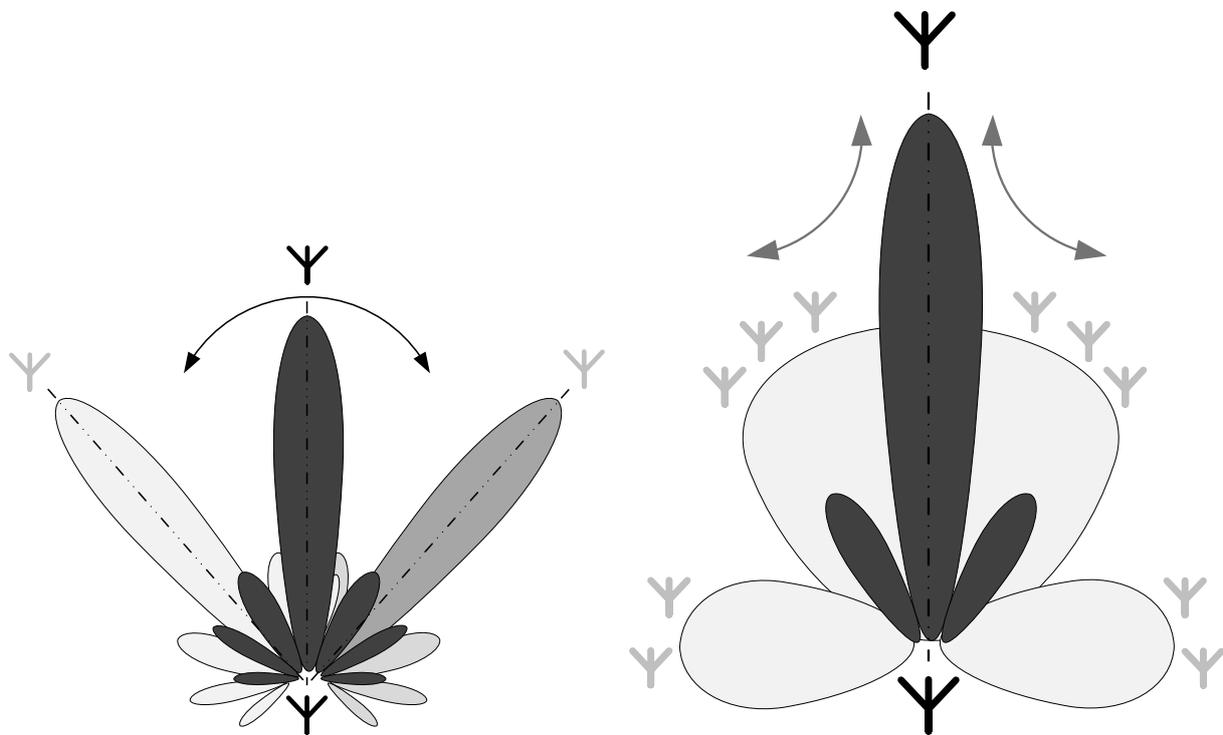
Abbildung 3.11.: Schematische Darstellung eines *Cognitive Radio Szenarios*. Die Endgeräte kommunizieren auf unterschiedlichen Frequenzbändern sowohl im Ad Hoc Modus als auch über eine hierarchische Infrastruktur.

Durch die Einführung des erweiterten *Frequenzspektrum Managements* soll Cognitive Radio eine langfristige Lösung im Bereich der effizienten Bandbreitennutzung zur Verfügung stellen. Das Konzept ermöglicht dem Nutzer eine zeitlich begrenzte Allokation bestimmter Frequenzbereiche und somit den Zugriff auf kurzfristig benötigte Ressourcen. Die Ressourcenverwaltung im Netzwerk wird hierbei dynamisch organisiert, wobei Methoden zur Synchronisation zwischen den jeweiligen Kommunikationspartnern bisher noch nicht definiert wurden.

³³Quality of Service

3.2.3. Smart Antennas / Intelligent Antennas

In den vorherigen Abschnitten wurden unterschiedliche Forschungsansätze für eine effiziente Datenübermittlung im Bereich mobiler Ad Hoc Netzwerke diskutiert. Die Schwerpunkte im Forschungsbereich *intelligenter Antennensysteme* korrelieren dabei teilweise mit dem Themenkomplex *Cognitive Radio*. Auch hier stehen die physikalischen Beschränkungen des Übertragungsmediums Luft im Fokus [62, 183]. Bedingt durch die wenigen frei nutzbaren Frequenzbänder kommt es bei einer Vielzahl simultan ablaufender Datenübertragungen verstärkt zu Beeinträchtigungen in Form von Interferenzen. Die damit einhergehenden Zugriffskonflikte bei der Allokation des Mediums führen zu einer starken Reduzierung der Übertragungsrate und in Extremfällen zu partiellen Ausfällen im Netzwerk. Hier bieten *Smart Antennas* einen interessanten Lösungsansatz auf physikalischer Ebene [25, 49].



(a) *Switched-Beam Antennensystem* - Durch den Einsatz von Motorik oder mehrerer Antennensektoren kann der Abstrahlwinkel des Systems gezielt auf einen Bereich gerichtet werden.

(b) *Adaptives Antennensystem* - Das System kann den Sendebereich zusätzlich in der Distanz anpassen, wodurch wahlweise ein großflächiger oder ein gerichteter Sendebereich erzeugt wird. Die Abstrahlleistung bleibt dabei konstant.

Abbildung 3.12.: Funktionsweise adaptiver Antennensysteme.

Anpassung der *Abstrahlrichtung* (a) und der *Abstrahldistanz/-fläche* (b).

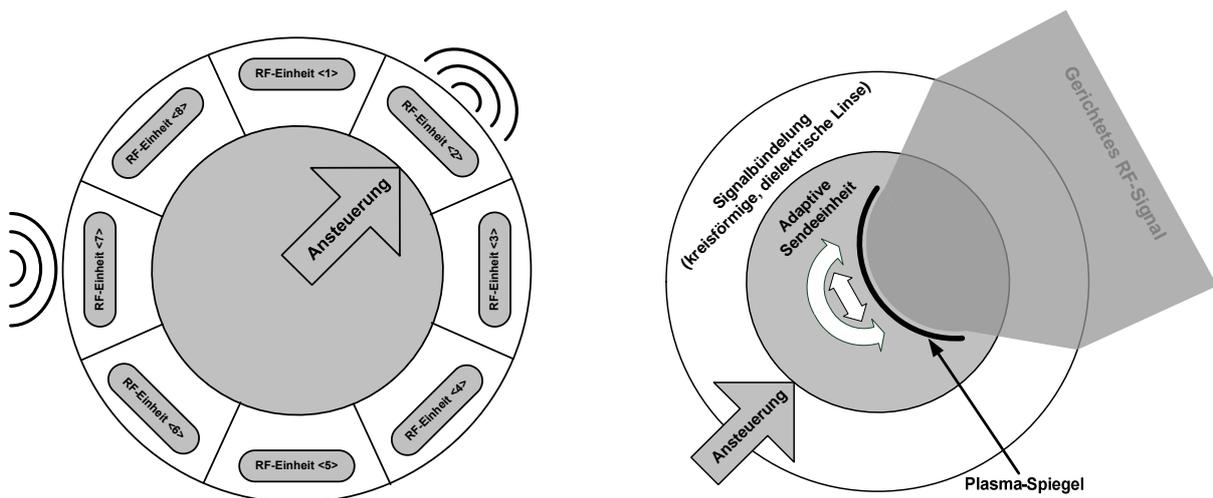
Die grundlegende Idee solcher Antennensysteme wird in Abbildung 3.12a) und 3.12b) schematisch dargestellt. Smart Antennas ermöglichen eine aktive Beeinflussung des Sendebereichs (*Beam-Forming*) durch den Nutzer bzw. durch die Algorithmen der Topologiekontrolle und des Routings [31, 63]. Somit ist es möglich, die verfügbare Netzwerktopologie durch eine Ausrichtung der einzelnen Sendeanlagen an spezifische Bedürfnisse, bspw. Redundanz oder Datenrate,

3. Stand der Forschung

anzupassen. Auf Routingebene können *Smart Antennas* eine deutliche Optimierung der Paketlaufzeit durch das Netzwerk ermöglichen, da durch eine Kopplung der aktiven Antennensysteme mit den Routingalgorithmen völlig neue Möglichkeiten der *Pfadfindung* ermöglicht werden.

Auf dieser Basis kann eine deutlich höhere Skalierung drahtloser Netzwerkstrukturen realisiert werden und es entsteht eine Vielzahl neuer Anwendungsgebiete in Bereichen, die bisher nur durch drahtgebundene Netzwerke realisierbar waren [100, 163].

Smart Antennas können in zwei wesentliche Kategorien unterteilt werden - *Switched-Beam* Systeme [87, 111, 125] und *Adaptive-Array* Systeme [44, 147, 155]. Systeme der *Switched-Beam* Klassifizierung (auch *Semi-Smart Antennas* genannt) bestehen aus einer Vielzahl von Einzelantennen, welche durch eine entsprechende Ansteuerung separat selektiert werden können. Ein solches System ist in Abbildung 3.13a) dargestellt und beschreibt eine ringförmige Anordnung der einzelnen Antennen mit definierten Abstrahlbereichen. Eine Steuerlogik analysiert eingehende Informationen und übernimmt die Zuordnung des Signals zur jeweiligen Antenne. Anhand dieser Informationen können nun zielgerichtet Entscheidungen über die Weiterleitung der Daten getroffen werden.



(a) *Adaptives Antennens Array* für den gerichteten Signaltransport zum Empfänger (*Switched-Beam* System [31]).

(b) *Adaptives Antennensystem* auf Basis von *frei formbaren Plasma Spiegeln*. Ein solches System benötigt für die räumliche Signalanpassung keinerlei mechanische Komponenten. Eine zusätzliche Vorrichtung für die Signalbündelung fokussiert das Funksignal (*Adaptive-Array* System [31]).

Abbildung 3.13.: a) *Adaptives Antennens Array*, b) *Adaptive Plasma Antenne*.

Adaptive-Array Antennen sind *voll rekonfigurierbare Antennensysteme*, bei denen nicht nur der Abstrahlwinkel verändert werden kann, sondern auch der Abstrahlbereich, wodurch bei gleicher Sendeleistung unterschiedliche Ausprägungen der Sendeabdeckung erreicht werden (Abbildung 3.12(b)). Aktuelle Forschungen auf dem Gebiet der *Plasma-Antennen* [142] (Abbildung 3.13b)) bieten durch den Einsatz von *Plasmaspiegeln* die Möglichkeit, flexible Antennensysteme zu schaffen. Die durch elektromagnetische Felder frei formbaren Spiegel benötigen keinerlei

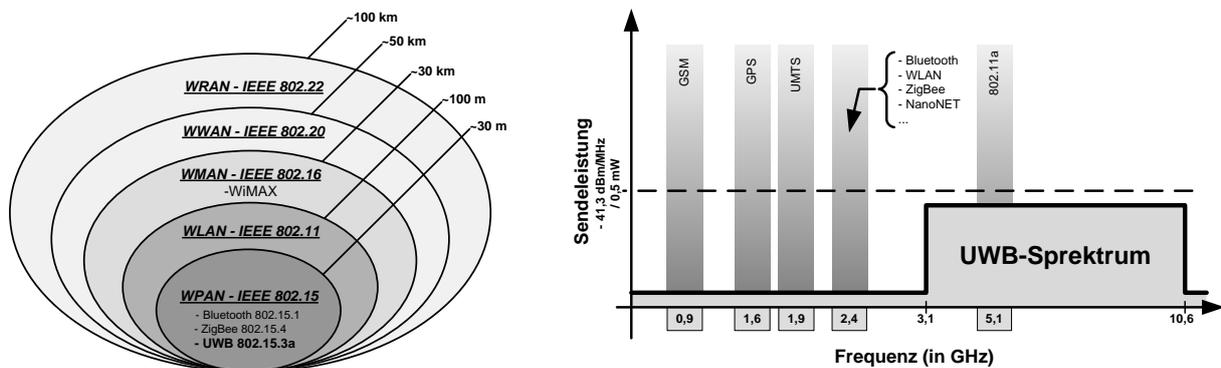
mechanische Komponenten und können somit innerhalb kürzester Zeit auf Änderungen in ihrer Umwelt reagieren.

Das im Rahmen dieser Arbeit entwickelte Konzept könnte auf Basis von Smart Antenna Systemen deutlich effizienter gestaltet werden. Die gegenseitigen Störeinflüsse unterschiedlicher Funkmodule reduzieren sich hierdurch signifikant. Da aktuell verfügbare Technologien jedoch noch keine adaptiven Antennensysteme vorsehen, ist der praktische Einsatz im Rahmen dieser Arbeit zum jetzigen Zeitpunkt nicht realisierbar.

3.2.4. Ultrabreitband

UWB³⁴ Technologien stellen ein separates Forschungsgebiet im Bereich moderner Nahbereichskommunikation dar. UWB bezeichnet dabei Signale, welche eine absolute Bandbreite von 500 MHz überschreiten [131]. In WPANs (*Wireless Personal Area Networks*) (Abbildung 3.14(a) bzw. Abschnitt 2.2.1) sollen durch UWB extrem hohe Datenraten von 480 - 1320 Mbit/s ermöglicht werden. Hierfür steht ein sehr breites Frequenzspektrum von 3,1 bis 10,6 GHz für die Datenübertragung zur Verfügung. Dieser Bereich ist in 500 MHz breite Bänder aufgeteilt, welche je nach Bedarf genutzt werden können.

Für die Allokation des Mediums kommen zwei mögliche Zugriffsverfahren zum Einsatz. *MB-OFDM*³⁵ - ein *Frequenz-Multiplexing* Verfahren, welches in ähnlicher Form bereits heute im Bereich *DAB*³⁶, WiMAX (IEEE 802.16) oder auch WLAN (IEEE 802.11) zum Einsatz kommt [131, 164]. Alternativ arbeitet die Firma Motorola mit dem *DS-CDMA*³⁷ Verfahren.



(a) Unterschiedliche drahtlose Kommunikationsstandards nach IEEE 802.x mit optimierten Eigenschaften für die jeweiligen Anwendungsbe-
reiche.

(b) Aktuell verwendete Frequenzbänder unterschiedlicher Funkstandards im Vergleich zum nutzbaren Frequenzspektrum vom UWB-Technologien mit deutlich verringerter Sendeleistung.

Abbildung 3.14.: a) IEEE spezifizierte Kommunikationsstandards
b) Übersicht Frequenzbänder und Sendeleistung.

³⁴Ultra Wide Band

³⁵Multiband Orthogonal Frequency Division Multiplexing

³⁶Digital Audio Broadcasting

³⁷Direct Sequence Code Division Multiple Access

3. Stand der Forschung

Ein wesentlicher Vorteil von UWB-Technologien ist die Reduzierung äußerer Störeinflüsse wie bspw. Wände oder andere Hindernisse. Durch das extrem breite Frequenzband beeinträchtigen diese die Kommunikation nur noch minimal. Eine weitere Problematik bei der drahtlosen Datenübertragung betrifft die physikalischen Ausbreitungsphänomene der Funksignale. Durch Beugung, Reflexion und Streuung entstehen sogenannte *Echosignale* des gesendeten Signals, welche auf der Gegenseite zeitverzögert und unter Umständen in vielfacher Ausführung empfangen werden (Abbildung 3.15). Auch hier weisen UWB-Systeme eine sehr hohe Resistenz gegen diese Art der *Multipath-Problematik* auf [71].

Um die Übertragungen anderer Funktechnologien nicht zu beeinträchtigen, wird die maximale Sendeleistung solcher Systeme auf nur 0,56mW begrenzt, was weniger als einem Tausendstel der Sendeleistung aktueller Mobiltelefone entspricht. Trotz dieser geringen Abstrahlleistung können extrem hohe Datenraten über eine effektive Sendereichweite von ca. 10 - 50 Metern erreicht werden [22, 131].

Auf Basis dieser Technologien sollen zukünftige Generation etablierter Funkstandards realisiert werden. Mit der Entwicklung und Einführung von *WUSB*³⁸ wird eine der weltweit am meisten genutzten Schnittstellen drahtlos realisiert. WUSB repräsentiert einen der ersten Vertreter von UWB-Technologien für den Massenmarkt. Nachfolgende Generationen von *Bluetooth* sollen ebenfalls UWB nutzen, um somit den Anforderungen aktueller multimedialer Anforderungen gerecht zu werden.

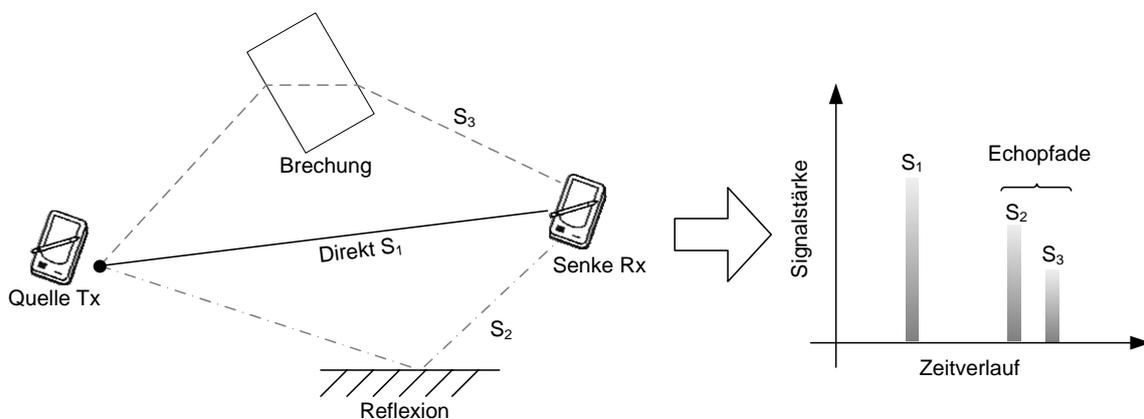


Abbildung 3.15.: *Multipath-Ausbreitungsszenario*: Der Sendeimpuls wird durch Reflexion und Brechung von der Senke in mehrfacher, zeitversetzter Ausführung empfangen, wodurch erhöhte Latenzzeiten entstehen. Einflüsse durch Interferenzen werden in diesem Zusammenhang zunächst vernachlässigt.

3.2.5. Ambient Networking

Der Forschungsbereich des *Ambient Networking* ist eng verknüpft mit den konzeptionellen Ansätzen der vorangegangenen Abschnitte des *Cognitive Radios* und *Software Defined Radio*. Ziel ist es hierbei, mit Hilfe einer *servicebasierten Abstraktionsschicht*, welche auf den

³⁸Wireless Universal Serial Bus

verfügbaren, physikalischen Netzstrukturen aufsetzt, eine *funkstandardübergreifende Interaktion* der Netzwerkknoten zu ermöglichen [20]. Dabei steht die Kopplung verfügbarer Netzwerke auf Basis unterschiedlicher Funktechnologien im Mittelpunkt der Bemühungen [128, 129]. Für den Nutzer wird somit versucht, eine *Always best connected* Situation zu schaffen [77], in der jedes zur Verfügung stehende Funknetzwerk für die Kommunikation genutzt wird (Abbildung 3.16).

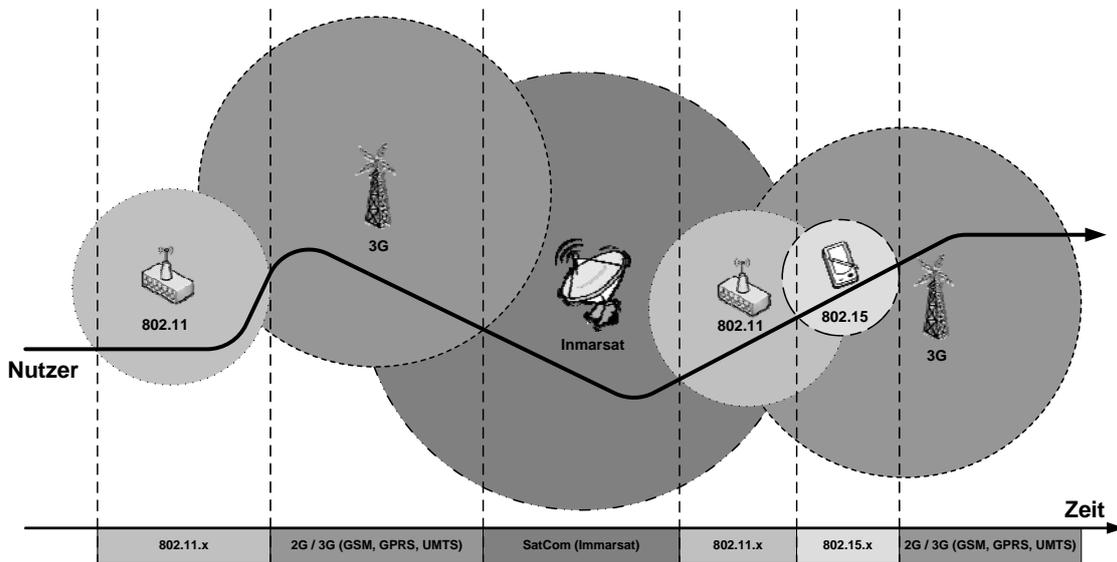


Abbildung 3.16.: Beispielszenario eines Ambient Network.

Je nach Anforderung an den Verbindungskanal und Verfügbarkeit entsprechender Kommunikationsstandards kann eine optimale Auswahl getroffen werden. Die sich daraus ergebenden Vorteile bei der Kommunikation, wie bspw. eine erhöhte Erreichbarkeit aller Knoten oder eine möglichst energieeffiziente Verbindungswahl, wurden in [98] sowie in [6] analysiert und in Teilen evaluiert.

Der schematische Aufbau eines solchen Systems wird in Abbildung 3.17 verdeutlicht. Es besteht im Wesentlichen aus drei Komponenten. Für Applikationen und den Nutzer steht eine definierte Schnittstelle zur Verfügung. Dieses *Ambient Service Interface* stellt eine Anzahl spezieller Dienste zur Verfügung, welche für die funkstandardübergreifende Kommunikation genutzt werden. Die Dienste sind Teil des *Ambient Control Space*, welcher die Kernkomponente repräsentiert. Hier sind weiterhin zusätzliche Komponenten für die Kommunikation in heterogenen Netzstrukturen verankert, wie bspw. ein erweitertes Ressourcenmanagement ([113]) oder eine Prioritätenverwaltung für spezielle Dienste.

Durch ein sogenanntes *Ambient Resource Interface* wird der Zugriff auf die verfügbare Hardware in Form unterschiedlicher Funkmodule realisiert. Diese sind in der *Ambient Connectivity* Komponente gekapselt und repräsentieren die unteren Kommunikationsschichten. Die Menge dieser Module stellt die äußere Schnittstelle zur Umwelt bereit. In Form eines abstrakten *Ambient Network Interface* können somit Kommunikationskanäle zu Netzwerken unterschiedlicher Funkstandards aufgebaut werden [104, 112].

3. Stand der Forschung

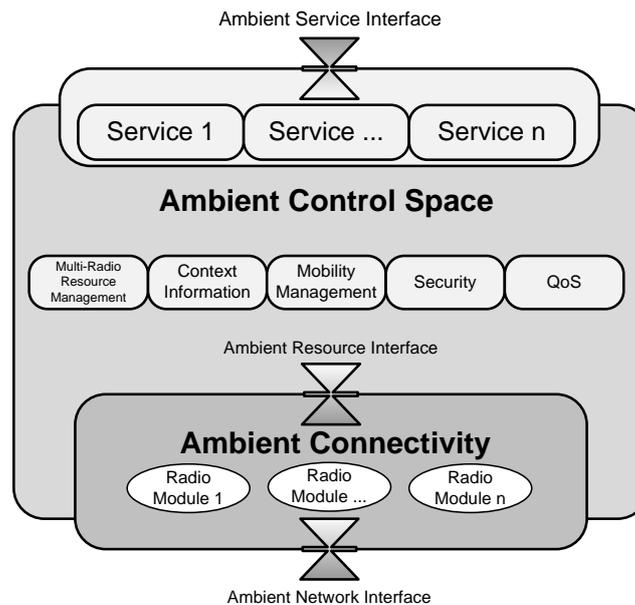


Abbildung 3.17.: Schematischer Aufbau eines *Ambient Network*.

[105] beschreibt eine mögliche Implementierung auf Medienzugriffsebene. Das darin vorgestellte System verfügt über eine dedizierte *Radio Access Selection* Einheit, welche die Nutzdaten auf die unterschiedlichen Funkmodule leitet. Im Gegensatz zu dem im Rahmen dieser Arbeit vorgestellten Konzept wird bei diesem Lösungsansatz der eingehende Datenverkehr an die darüberliegenden Schichten des Betriebssystems bzw. der Applikationen weitergereicht. Diese müssen dann Entscheidungen für die eventuelle Weiterleitung der Nutzdaten treffen. Das System sieht keine Möglichkeit für ein selbstständiges Routing unterhalb der Betriebssystemebene vor, obwohl alle dafür notwendigen Informationen in Form von IP und Hardwareadressdaten zur Verfügung stehen. Ein weiterer, wesentlicher Unterschied stellt die abstrakte Umsetzung der nutzbaren Schnittstellen auf Anwendungsebene dar. Applikationen können somit nur auf eine Menge vordefinierter Dienste zugreifen, welche durch den *Ambient Control Space* (Abbildung 3.17) bereitgestellt werden [128, 129]. In den jeweiligen Anwendungen sind spezielle Anpassungen nötig, um die entstandene, heterogene Netzstruktur nutzen zu können. Eine zentrale Applikation verwaltet und steuert den Zugriff auf die physische Netzwerkschnittstelle, wodurch die Belastung des Hosts steigt.

Das im folgenden Kapitel vorgestellte Konzept umgeht diese Nachteile und stellt einen neuen Lösungsansatz im Bereich *Ambient Networking* vor. Die gesamte Komplexität für die Verwaltung und die Steuerung der verfügbaren Funkmodule wird dabei auf eine hardwarenahe Ebene verlagert. Aus diesem Grund wird es als *Embedded Ambient Networking* bezeichnet und verbindet die konzeptionellen Ideen des *Ambient Networking* mit der *funktionalen Kapselung in einem eingebetteten System*. Neben den Vorteilen durch einen deutlich reduzierten Verwaltungsaufwand für die Funkmodule und die Weiterleitung der IP-Pakete auf Betriebssystemebene, können Anwendungen die heterogene Netzinfrastruktur ohne Modifikationen nutzen. Die folgenden Kapitel geben einen detaillierten Einblick in die Funktionsweise und den Aufbau eines solchen Systems.

3.3. Zusammenfassung

Das Kapitel *Stand der Forschung* vermittelte einen umfassenden Einblick in aktuelle Forschungsbereiche auf dem Gebiet mobiler Ad Hoc und Sensornetzwerke. Dabei wurden die wesentlichen Problemstellungen, primäre Anwendungsgebiete sowie mögliche Lösungsansätze aufgeführt. Der Fokus lag dabei auf der Umsetzbarkeit in Bezug auf die Problemstellungen dieser Arbeit.

Zentrale Themen befassen sich zunächst mit der effizienten Kommunikation in mobilen Ad Hoc Netzwerken durch *Topologieoptimierungsverfahren* sowie angepasste *Routingstrategien*.

Im zweiten Teilabschnitt wurde auf zukünftige Generationen drahtloser Kommunikationstechnologien eingegangen. Die betrachteten Forschungsthemen umfassen *Software Defined Radio*, *Cognitive Radio* sowie *Ambient Networking*. Thematische Zusammenhänge mit dem im Rahmen dieser Arbeit vorgestellten Konzept wurden verdeutlicht. Arbeiten auf dem Gebiet intelligenter Antennensysteme (*Smart Antennas*) oder *Ultrabreitband-Technologien* runden den Themenkomplex ab.

Nachdem ein breites Spektrum aktueller Forschungsbereiche sowie grundlegende Kenntnisse auf dem Gebiet der drahtlosen Kommunikation vermittelt wurden, folgt nun die Einführung eines neuen Konzeptes für die funkstandardübergreifende Kommunikation in mobilen Ad Hoc Netzwerken basierend auf der Integration verfügbarer, drahtloser Kommunikationstechnologien.

4. Embedded Ambient Networking

Der Kernpunkt dieser Arbeit präsentiert ein funkstandardübergreifendes Kommunikationskonzept im Bereich mobiler Ad Hoc Netzwerke. Dabei sollen die Vorteile unterschiedlicher Kommunikationstechnologien genutzt und gleichzeitig wesentliche Nachteile der vorgestellten Lösungsansätze umgangen werden. Dies betrifft vor allem den Verzicht auf leistungsfähige und teure Spezialhardware. Desweiteren soll die Nutzung der geschaffenen Funktionalität ohne aufwendige Anpassungen auf Softwareebene ermöglicht werden.

Unter dem Namen *Embedded Ambient Networking (EAN)* wird eine solche Möglichkeit geschaffen, unterschiedliche Funkstandards in eine heterogene Netzstruktur zu integrieren [9, 10]. Dadurch ergeben sich vielfältige Optimierungsmöglichkeiten in Hinblick auf den Energiebedarf, die Erreichbarkeit sowie die Verbindungsqualität. Zum Einsatz kommen hierbei standardisierte Funkmodule, welche getestet und in großer Stückzahl am Markt verfügbar sind. Diese werden mit Hilfe einer zentralen Konverterkomponente hardwarenah gekoppelt und koordiniert.

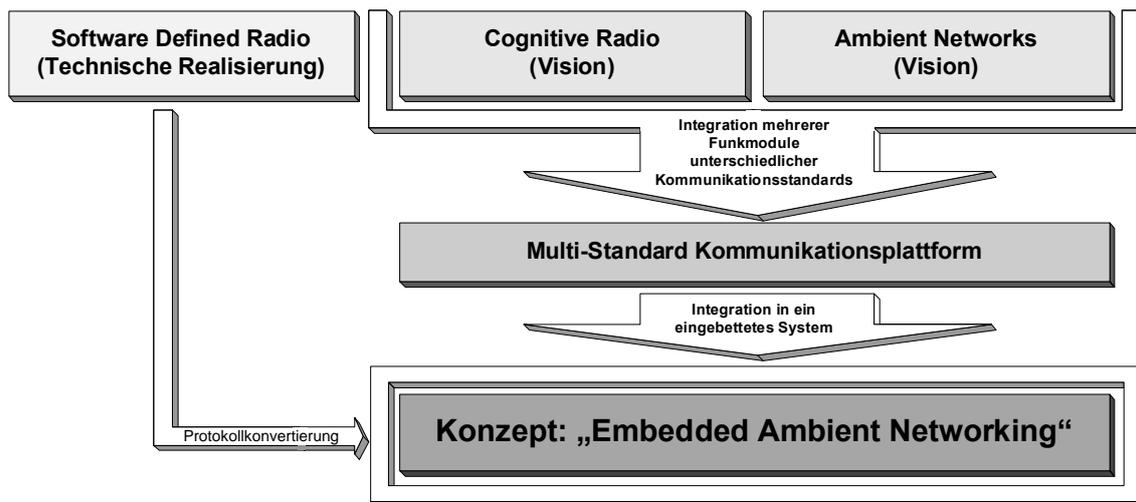


Abbildung 4.1.: Einordnung des Konzeptes in die aktuellen Forschungsthemen. Der vorgestellte Ansatz koppelt die funktionalen Eigenschaften von *Cognitive Radio* und *Ambient Networking* mit konzeptionellen Aspekten aus dem Bereich *Software Defined Radio* auf einer hardwarenahen Ebene.

Abbildung 4.1 verdeutlicht die Herleitung aus den zuvor diskutierten Lösungsansätzen. Wesentliche Vorteile können dabei wie folgt zusammengefasst werden [4, 6, 8]:

- Steigerung der Erreichbarkeit → Integration (teils inkompatibler) Knotenmengen in eine heterogene Netzwerkstruktur
- gleichzeitige Optimierung des Energiebedarfs über die gesamte Topologie
- Verwendung handelsüblicher, weitverbreiteter Hardwaremodule

4. Embedded Ambient Networking

- optimierte Datenübertragung durch Nutzung der Vorteile unterschiedlicher, standardisierter Funktechnologien
- Kapselung der Verwaltung in dedizierter Hardware → keine zusätzliche Belastung des Host-Systems
- transparente (nicht sichtbare) Auswahl des physisch genutzten Kommunikationsstandards für Applikationen
- modulare Erweiterbarkeit um neue Kommunikationsstandards

Bezugnehmend auf den aktuellen Stand der Technik und der fortschreitenden Entwicklung im Informations- und Kommunikationssektor wird der Mehrwert deutlich. Die Entwicklung immer neuer Funkstandards mit eng definierten, anwendungsspezifischen Übertragungseigenschaften verursacht enorme Kosten und bindet folglich viele Ressourcen. Die Leistungsparameter verfügbarer Technologien sind jedoch für nahezu jedes Anwendungsszenario ausreichend. Durch eine sinnvolle Kombination mehrerer Funkstandards in einem integrierten Kommunikationssystem eröffnen sich unzählige neue Einsatzbereiche. Somit kann der zeit- und kostenintensive Entwicklungsaufwand deutlich reduziert werden.

4.1. Konzept

Die Struktur des vorgestellten EAN Konzeptes wird in Abbildung 4.2 verdeutlicht. Das System realisiert dabei eine autarke, eingebettete Kommunikationsplattform.

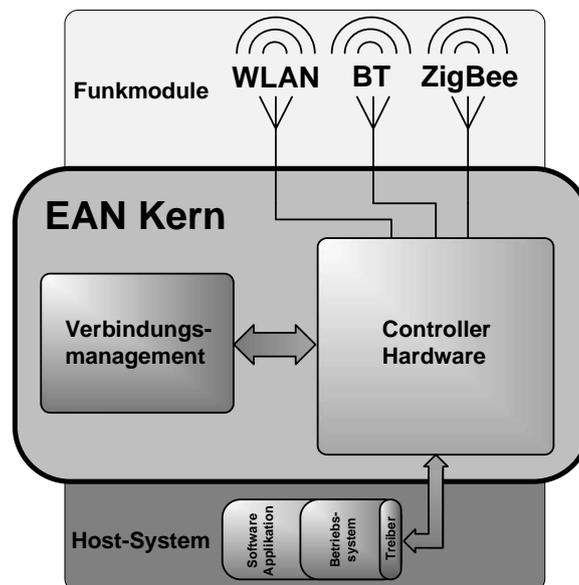


Abbildung 4.2.: Schematische Darstellung eines Netzwerkknoten nach dem hier vorgestellten Konzept.

Das vorgestellte Konzept unterteilt sich dabei in vier zentrale Komponenten. Zum einen umfasst es eine Menge *dedizierter Funkmodule* unterschiedlicher Kommunikationsstandards, wel-

che simultan betrieben werden und auf unterschiedlichen Protokollebenen an den EAN Kern gekoppelt sind.

Die zweite Komponente bildet die *Controller-Hardware*, welche die verschiedenen Funkmodule auf einer hardwarenahen Ebene integriert. Dieser Hardware-Block koordiniert alle prozeduralen Abläufe und realisiert somit eine Konvertierung zwischen den angeschlossenen Modulen.

Für das Host-System steht eine *dedizierte Schnittstelle* zur Verfügung, welche die dritte Komponente innerhalb des EAN Konzeptes bildet. Der Host erhält dabei empfängerspezifisch gefilterte Daten von der Controller-Einheit. Im Umkehrschluss nutzen Softwareapplikationen und das Betriebssystem die Anbindung zur Controller-Hardware für ausgehende Daten. Das für die Übertragung genutzte, physische Interface ist in Folge dessen für den Host nicht sichtbar.

Eine vierte, zentrale Komponente realisiert das gesamte *Verbindungsmanagement* für die integrierten Funkmodule und ist dabei für die Initialisierung und Pflege aller Kommunikationskanäle verantwortlich.

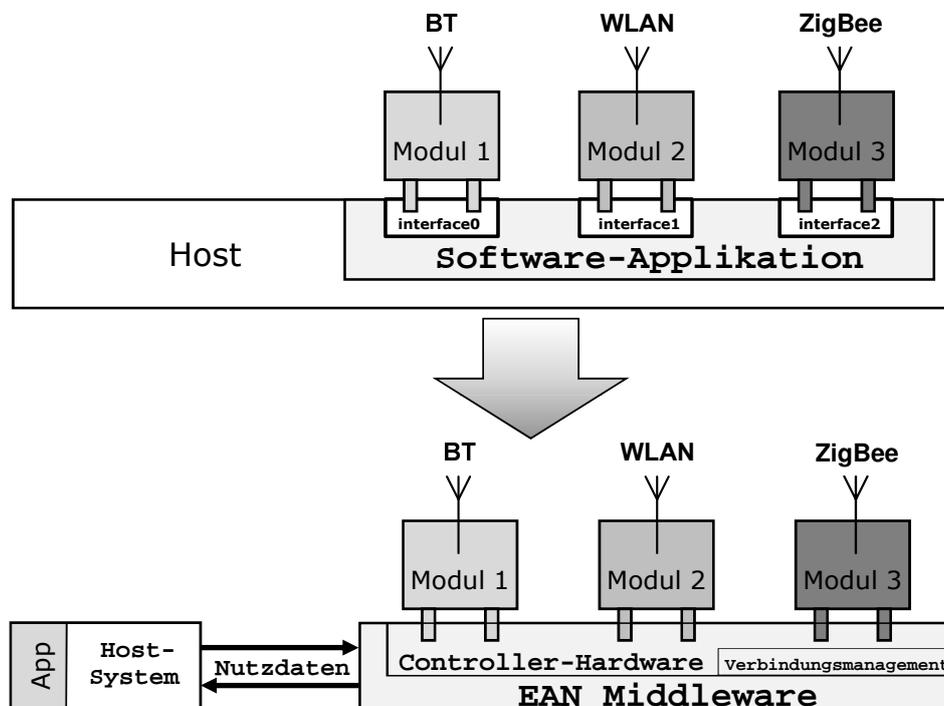


Abbildung 4.3.: Das vorgestellte Konzept *Embedded Ambient Networking* kapselt und koppelt die Funkmodule auf einer hardwarenahen Ebene (Bild unten). Eine entsprechende Steuereinheit leitet und kontrolliert den Datenfluss. Für das Host-System steht nur eine virtuelle Netzwerkschnittstelle zur Verfügung.

Der Unterschied zur klassischen Verwaltung mehrerer Netzwerkschnittstellen wird in Abbildung 4.3 verdeutlicht. Herkömmliche Systeme verwalten die Schnittstellen der jeweiligen Module durch Treiber auf Softwareebene, was speziell in ressourcenbeschränkten, eingebetteten Systemen eine Belastung des Betriebssystems zur Folge hat. Es steht folglich eine Menge einzelner, gekapselter Netzwerkschnittstellen zur Verfügung, welche durch die Softwareapplika-

4. Embedded Ambient Networking

tionen oder das Betriebssystem separat genutzt werden können. Ist das Zielsystem über diese Schnittstelle nicht erreichbar, bricht die Datenübertragung ab. Eine Möglichkeit, die Kommunikation mit Hilfe eines alternativen Netzwerkkinterfazes zu überbrücken, muss explizit implementiert werden.

Genau diese Problematik wird durch das *EAN* Konzept umgangen. Hier sucht die Controller-Hardware aus dem Pool verfügbarer Netzwerkschnittstellen geeignete Verbindungskanäle zum Zielhost. Bei einem Ausfall der Verbindung ist das System in der Lage, dynamisch und für die Softwareapplikation nicht sichtbar, einen alternativen Pfad zur Verfügung zu stellen. Eine Anpassung der Software ist dazu nicht notwendig.

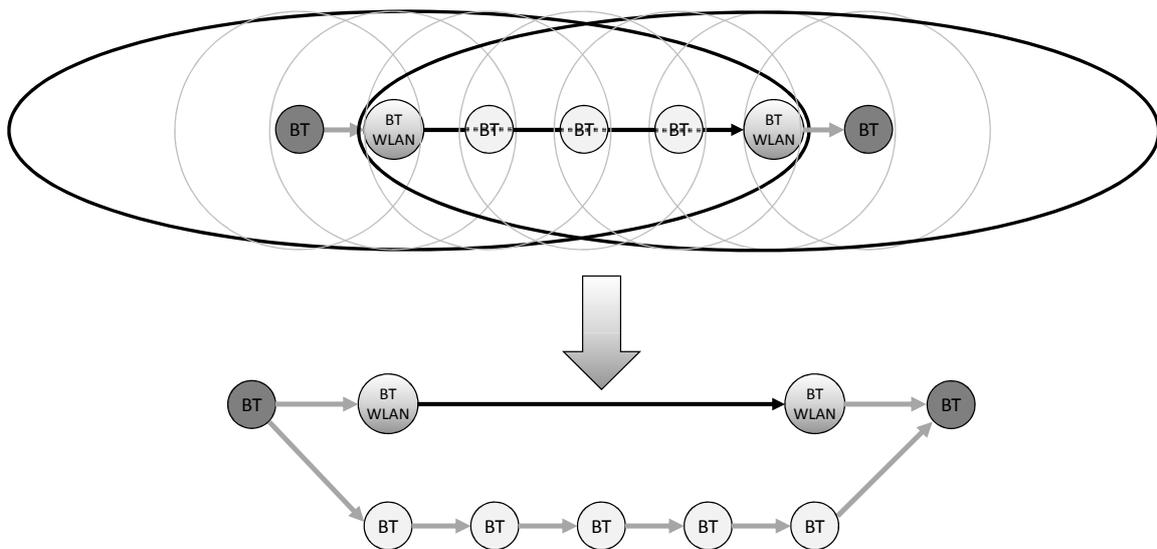


Abbildung 4.4.: Ein wesentlicher Vorteil des hier vorgestellten Konzeptes - Die Integration mehrerer Funkstandards und die damit verbundene Steigerung der Erreichbarkeit.

Ausgehend von dem in Abbildung 4.4 dargestellten Beispiel können die Vorteile auf einfache Weise erläutert werden. Im Zuge einer Multihop-Kommunikation der beiden Bluetooth-Knoten müssen nach dem klassischen Ansatz fünf Zwischenstationen in Anspruch genommen werden (Abbildung 4.4 - unterer Kommunikationspfad). Dies hat entsprechende Auswirkungen auf die Netzwerklatenzen. Da keine alternativen Pfade zur Verfügung stehen, ist diese Verbindung im aktuellen Zustand zwar als stabil zu bezeichnen, jedoch kann es aufgrund von äußeren Störeinflüssen relativ schnell zu Unterbrechungen kommen. Auf Basis des EAN Konzeptes ändert sich das Szenario deutlich. Quell- und Zielknoten verfügen weiterhin ausschließlich über eine Bluetooth-Schnittstelle. Gelangen die Daten allerdings zum ersten Zwischenknoten, stehen nun zwei physische Schnittstellen zur Verfügung. Mit Hilfe einer intelligenten Steuerung kann die Controller-Hardware somit eine alternative Schnittstelle auf Basis von WLAN nutzen, um so bspw. die Latenz des Verbindungskanals zu optimieren. Indem die WLAN-Brücke drei Bluetooth Zwischenknoten überspringt, reduziert sich folglich die Anzahl der Hops erheblich (Abbildung 4.4 - oberer Kommunikationspfad). Innerhalb der übersprungenen Zwischenknoten sinkt zusätzlich der Energiebedarf und somit kann die Betriebsbereitschaft länger aufrecht erhalten werden. Aber auch im Falle von Störungen kann dynamisch und adaptiv zwischen zwei

Verbindungskanälen gewählt werden. Diese zusätzliche Redundanz erhöht in direkter Konsequenz die Erreichbarkeit aller Knoten im Netzwerk. Desweiteren integriert ein solches System zwei homogene, inkompatible Knotenteilmengen in eine heterogene Netzwerkstruktur.

Im konkreten Beispielszenario aus Abbildung 4.4 entsteht durch das zusätzliche WLAN-Netzwerkinterface zunächst eine negative Energiebilanz für die beiden Multistandard-Knoten, da der Betrieb des zweiten Funkmoduls entsprechend mehr Energie benötigt. Betrachtet man jedoch die gesamte Knotentopologie, können während der Kommunikation durch die Nutzung des WLAN-Funkstandards drei Bluetooth Zwischenknoten übersprungen werden. Dementsprechend benötigen diese Knoten keine Energie für das Empfangen, Bearbeiten und Weiterleiten der Daten. Die durchschnittliche Energiebilanz über die gesamte Topologie fällt somit trotz des zusätzlichen Betriebs der WLAN-Module mit hoher Wahrscheinlichkeit positiv aus. Grundlage für diese Annahme ist jedoch ein durchschnittliches Datenaufkommen in der Topologie, da bei einem Szenario ohne Netzlast auch keine zusätzlich Energie durch die Übertragung benötigt wird.

Besonders deutlich wird das Energieeinsparungspotenzial in komplexeren Netzstrukturen mit einer höherer Knotenanzahl. Das Konzept setzt dabei nicht ausschließlich Multistandard-Knoten voraus. Existiert in der Topologie ein hinreichend hoher Prozentsatz solcher Knoten, reicht dies vollkommen aus, um die gesamte Kommunikation im Netzwerk deutlich zu optimieren. Angepasste Wegfindungsalgorithmen berechnen hier kostenoptimierte Pfade durch die Topologie. Je nach Anforderung an den Verbindungskanal erfolgt eine optimale Auswahl des genutzten Funkstandards. Optimierungskriterien sind dabei:

- Datenrate
- Energiebedarf
- Latenz (Verbindungsinitialisierung sowie die gesamte Signallaufzeit)
- Sendereichweite
- Frequenzband
- Interferenzbedingte Störanfälligkeit

Auf Basis dieser Metriken kann eine Priorisierung existierender Routenpfade erfolgen. Die Erreichbarkeit aller Knoten in der Topologie wird durch ein das EAN Konzept signifikant gesteigert und in Folge dessen sinkt Anfälligkeit gegen Störeinflüsse.

4.2. Einordnung in das ISO/OSI Modell

Das EAN Konzept koppelt mehrere Funktechnologien auf unterschiedlichen Ebenen. Der Verarbeitungsprozess auf den unteren Protokollstack-Ebenen wird dabei in die Funkmodule verlagert. Hier übernehmen die integrierten Controller alle prozeduralen Abläufe.

4. Embedded Ambient Networking

Als gemeinsame Basis für die Adressierung und Verarbeitung dient das IP Protokoll auf Netzwerkebene (siehe Abbildung 4.5). Innerhalb des ISO/OSI Modells entspricht dies der Vermittlungsschicht (Ebene 3). Innerhalb des EAN Kerns können die IP Headerinformationen problemlos analysiert und ggf. angepasst werden.

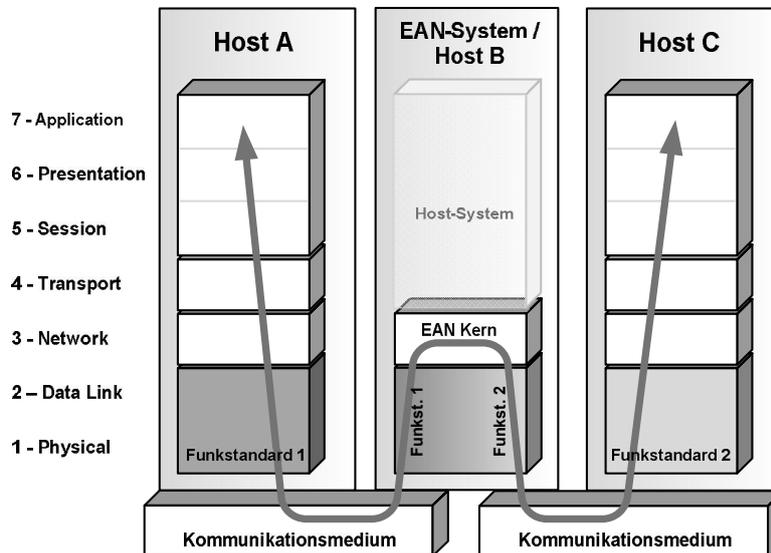


Abbildung 4.5.: EAN: Einordnung in das ISO/OSI Modell.

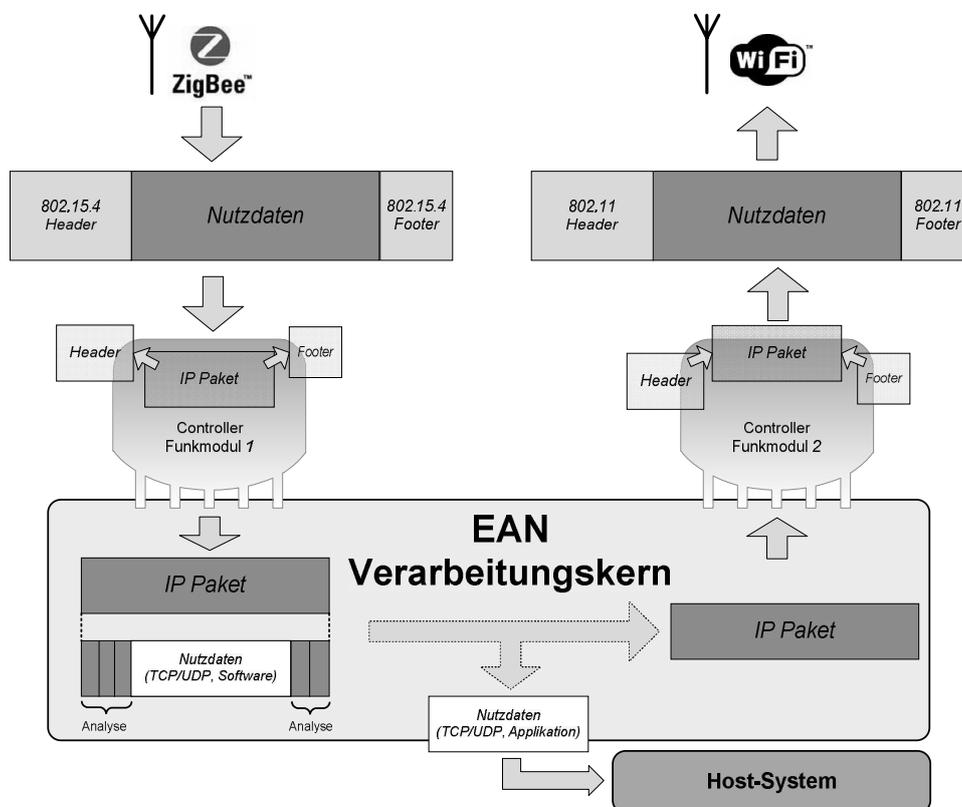


Abbildung 4.6.: EAN Verarbeitungsprozess auf Basis von IP als gemeinsames Adressierungsmodell.

Im Zuge der EAN Datenverarbeitungsprozesses entsprechen die Nutzdaten somit vollständigen IP-Paketen, welche zielgerichtet weitergeleitet werden. Im weiteren Verlauf dieser Arbeit definiert sich somit der Begriff Nutzdaten im Kontext des EAN Systems als eine Menge zu verarbeitender IP-Pakete. Abbildung 4.6 verdeutlicht diesen Prozess schematisch.

Eingehende Datenpakete werden durch die Controller des jeweiligen Funkmoduls von den funkstandardspezifischen Protokollrahmen befreit. Der EAN Kern nimmt die Nutzdaten in Form eines kompletten IP-Paketes entgegen und analysiert anschließend die Headerinformationen. Sind die Daten an das angeschlossene Host-System gerichtet, leitet das EAN-System den Inhalt entsprechend weiter. Anderenfalls wird durch das Verbindungsmanagements ein geeignetes Ziel für die Paketweiterleitung ausgehandelt und ein Verbindungskanal initialisiert. Der Verarbeitungsprozess endet dabei mit der Übergabe des IP-Paketes an den integrierten Controller des gewählten Funkmoduls, welcher das IP-Paket als Nutzdaten (*Payload*) in einem Protokollrahmen kapselt und anschließend überträgt.

4.3. Anforderungen

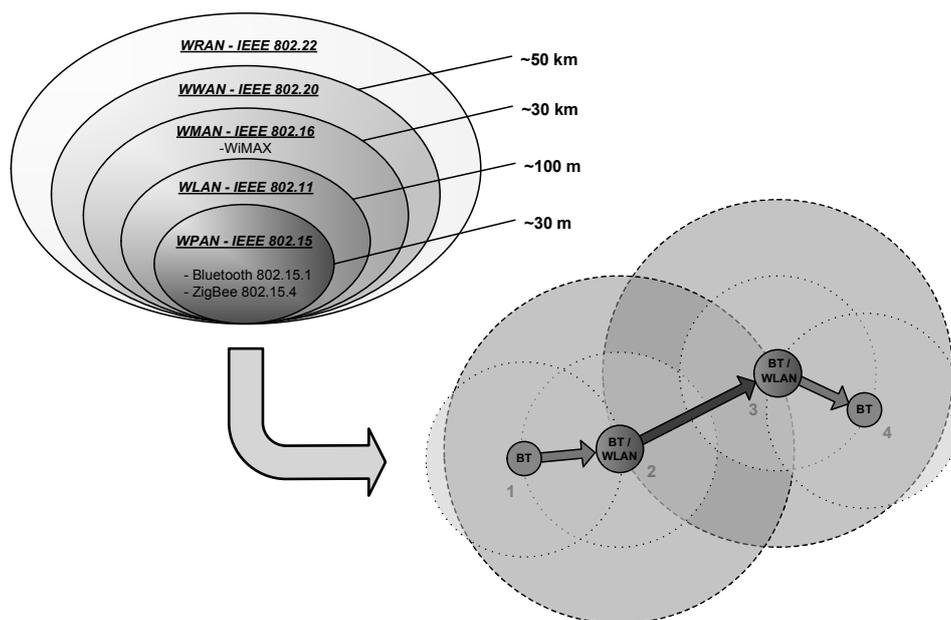


Abbildung 4.7.: Beispielszenario für eine funkstandardübergreifende Kommunikation in mobilen Ad Hoc Netzwerken. Knoten 1 und 4 können auf direktem Weg nicht kommunizieren. Verfügen dazwischenliegende Netzwerkknoten (2 und 3) über zusätzliche Kommunikationsstandards und entsprechende Konvertierungsmechanismen, kann die Erreichbarkeit in der Netzwerktopologie signifikant gesteigert werden. Die entstandene, heterogene Netzstruktur integriert dabei mehrere Kommunikationstechnologien, welche jeweils optimierte Eigenschaften für spezifische Anwendungsbereiche aufweisen.

An das vorgestellte EAN Konzept werden Anforderungen auf unterschiedlichen Ebenen gestellt. Auf der einen Seite betrifft dies nutzerseitige Anforderungen, welche für Nutzung der

4. *Embedded Ambient Networking*

geschaffenen Netzinfrastruktur essentiell sind. Auf der anderen Seite entstehen durch die Integration verschiedener Kommunikationstechnologien mehrere Problemstellungen, welche im auf konzeptioneller Ebene spezifiziert und im Rahmen einer praktischen Realisierung gelöst werden müssen. Das in Abbildung 4.7 dargestellte Beispielszenario zeigt den Umfang des EAN Konzeptes.

4.3.1. Nutzerseitige Anforderungen

Das EAN Konzept stellt die Rahmenbedingungen für eine Integration unterschiedlicher Kommunikationstechnologien zur Verfügung. Um auf dieser Basis eine effiziente Kommunikation sicherzustellen, existieren vielfältige, nutzerseitige Anforderungen.

4.3.1.1. Administrativer Aufwand innerhalb des Host-Systems

Klassische Architekturen verwalten mehrere Netzwerkschnittstellen auf Betriebssystemebene als gekapselte, autarke Kommunikationseinheiten. Im Gegensatz dazu koppelt ein EAN System die Funkmodule auf einer unteren, hardwarenahen Ebene mit Hilfe einer Middleware. Die entwickelte Controller Hardware realisiert dabei eine Konvertierung zwischen den unterschiedlichen Übertragungsprotokollen. Somit entfällt betriebssystemseitig die Verwaltung mehrerer separater Netzwerkadapter. Auf Applikationsebene sind somit keine expliziten Anpassungen notwendig. Zum EAN System steht ausschließlich eine dedizierte Netzwerkschnittstelle für die Kommunikation zur Verfügung. Somit werden auch Entscheidungen über die Wahl des physischen Interface auf EAN Ebene getroffen. Das Betriebssystem, speziell im Umfeld mobiler Kleincomputer, kann sowohl in administrativer Hinsicht als auch in Hinblick auf die verfügbaren Ressourcen deutlich entlastet werden. Die gesamte Verwaltung und Steuerung erfolgt autark auf den unteren Ebenen der EAN Plattform. Die Komplexität auf Betriebssystemebene bleibt auch bei Erweiterungen der Kommunikationsschnittstellen konstant. Dies stellt eine zentrale Anforderung an das EAN Konzept dar.

4.3.1.2. Transparenz der physischen Netzwerkschnittstellen

Die im EAN System integrierten, realen Schnittstellen spielen für Softwareapplikationen auf dem Host-System keine wesentliche Rolle. Die Menge der physischen Kommunikationsschnittstellen wird auf ein einzelnes virtuelles Interface abgebildet. Host-System-seitig wird über diese Netzwerkschnittstelle die gesamte Kommunikation über die EAN Middleware abgewickelt. Entscheidungen über die Auswahl des zu nutzenden Netzwerkadapters verlagern sich folglich in das EAN System. Die eingehenden Datenpakete werden hinsichtlich der enthaltenen Zielinformationen durch den EAN Kern analysiert. Anschließend sind auf dieser Wissensbasis Entscheidungen über die zu nutzende, physische Schnittstelle möglich. Für die Softwareapplikation auf dem Host-System sind diese Entscheidungen nicht sichtbar. Das EAN System stellt sicher, dass ein optimaler Verbindungskanal gewählt wird. Die dazu notwendige Intelligenz muss in

der Controller-Hardware zur Verfügung gestellt werden.

4.3.1.3. Metadaten - softwareseitige Anforderungen an den Verbindungskanal

Das Konzept sieht vor, dienstspezifische Anforderungen hinsichtlich des Verbindungskanals zu berücksichtigen und dementsprechend die Routenwahl anzupassen. Je nach Applikation sind dabei Faktoren wie bspw. Datenrate oder maximale Latenzzeiten von besonderem Interesse. Diese Beeinflussung kann durch das *TOS*¹ Feld innerhalb des des verwendeten IPv4 Headers umgesetzt werden (Abbildung 4.8). Die Spezifikation sieht hierbei eine Priorisierung nach Latenz, Datendurchsatz und Zuverlässigkeit vor [137]. Seit 2001 greift eine detailliertere und umfangreichere Unterscheidung nach [145] auf Basis von DSCP² und ECN³.

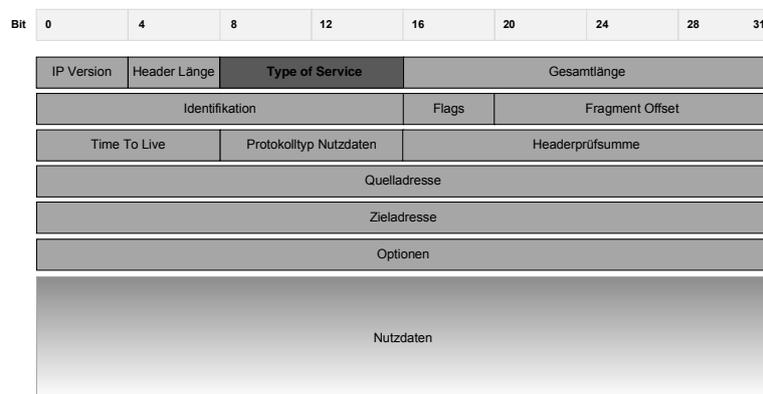


Abbildung 4.8.: Aufbau eines IP Version 4 (IPv4) Paketes.

Im Zuge einer späteren Weiterentwicklung integriert IPv6 in seiner Protokollspezifikation explizit QoS Funktionalitäten, welche ebenfalls genutzt werden können (*IPv6* - Abbildung 4.9).

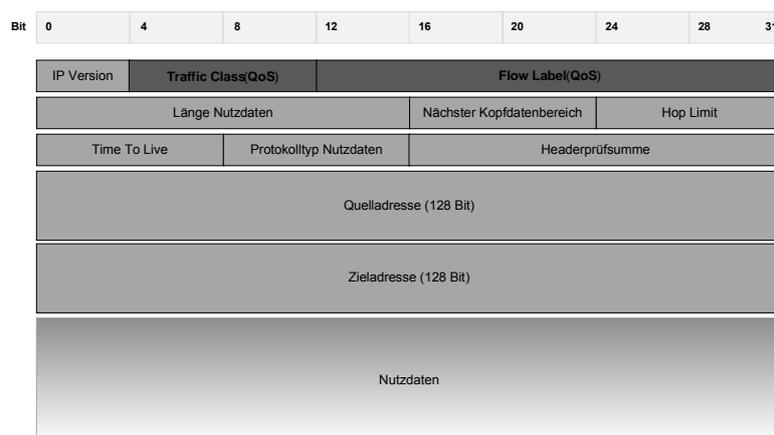


Abbildung 4.9.: Aufbau eines IP Version 6 (IPv6) Paketes. Die im Rahmen von IP Version 6 spezifizierten QoS Funktionalitäten sind dunkel hinterlegt.

¹Type of Service

²Differentiated Services Code Point

³Explicit Congestion Notification - IP-Flusskontrolle

4. Embedded Ambient Networking

Je nach dem, welche IP Version zur Anwendung kommt, werden die Pakete entsprechend codiert und durch die Controller-Hardware des EAN Systems anschließend interpretiert. So haben Nutzer eine Möglichkeit, mit verfügbaren Funktionen des Betriebssystems anwendungsspezifische Anforderungen auf ein definiertes Prioritätslevel abzubilden. Die Controller-Hardware erlaubt dann eine optimierte Pfadsuche. Die Bereitstellung solcher Metadaten könnte alternativ auch auf höherer Protokollebene erfolgen, was jedoch host- bzw. softwareseitig Anpassungen zur Folge hätte.

4.3.1.4. Kompatibilität

Das EAN Konzept soll neben den vorgestellten Multistandard-Systemen auch herkömmliche Endgeräte integrieren (Abbildung 4.10). Diese treten dabei ausschließlich als passive Endgeräte ohne Routingmechanismen auf, da diese Funktionalitäten von den Herstellern meist nicht vorgesehen bzw. nicht umgesetzt wurden. Primär betrifft diese Geräte wie Mobiltelefone oder PDAs mit ihren drahtlosen Schnittstellen. Mobile PC-Systeme und Laptops repräsentieren Grenzfälle, bei denen die Funktionalität softwareseitig durch eine Vielzahl angepasster Betriebssysteme und Softwareapplikationen erweitert werden kann. Alternativ ist daher die vollständige, softwareseitige Umsetzung eines EAN Systems auf Applikationsebene realisierbar. Erweiterte Routingaufgaben sind auf dieser Ebene ebenfalls möglich. Durch eine Softwarerelösung ist jedoch keine Schnittstellentransparenz mehr gegeben, was die damit verbundenen Nachteile mit sich bringt.

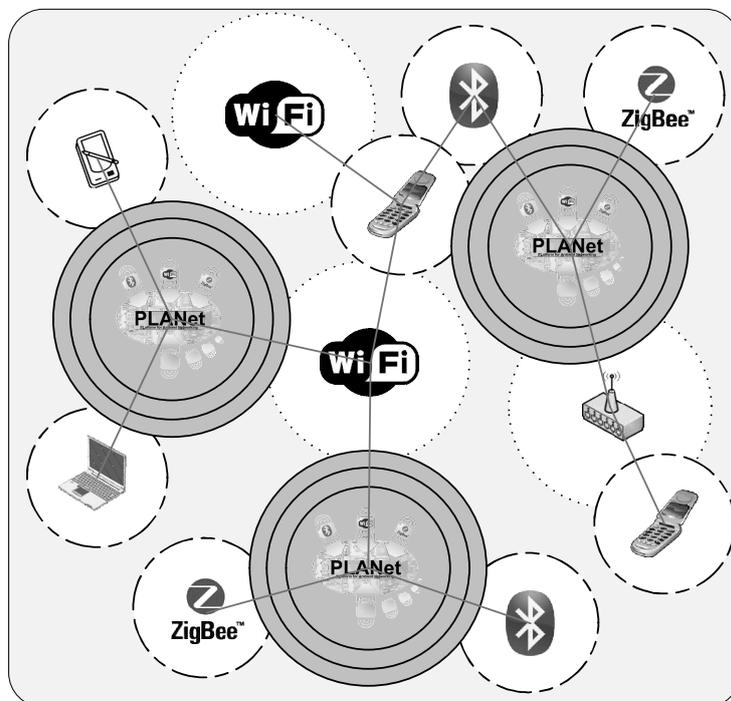


Abbildung 4.10.: Beispielszenario für die Kompatibilitätsanforderungen innerhalb des EAN Konzeptes. Die dargestellte, heterogene Netztopologie integriert sowohl EAN-konforme Multistandard-Knoten als auch herkömmliche, drahtlose Endgeräten.

Der EAN Ansatz verwendet IP als gemeinsames Adressierungsprotokoll. Entsprechende Implementierungen für TCP/IP Protokollstacks stehen für sehr viele Hardwareplattformen zur Verfügung. In Folge dessen kann auf den höheren Protokollebenen eine kompatible Kommunikation sichergestellt werden. Probleme treten ausschließlich auf den unteren Protokollebenen auf. Einige existierende Kommunikationsstandards, z.B. Bluetooth, durchlaufen komplexe Prozesse im Zuge des Verbindungsaufbaus. Die damit verbundenen Verzögerungen müssen beim Systementwurf entsprechend beachtet werden.

Weitere Probleme treten bspw. auf, wenn mobile Kleincomputer mit stark begrenzten Hardwareressourcen, wie bspw. im Bereich drahtloser Sensornetzwerke (Kapitel 3.1.3), integriert werden sollen. Hier kommt ein vollständiger TCP/IP Stack nur in seltenen Fällen zum Einsatz. Folglich muss eine Entscheidung getroffen werden, ob diese Szenarien grundlegend auszuschließen sind. Auf diese Problematik wird im Abschnitt *Konzeptionelle Grenzen* 4.6 näher eingegangen.

4.3.1.5. Modulare Erweiterbarkeit

Die modulare Erweiterbarkeit des vorgestellten Lösungsansatzes repräsentiert eine weitere, zentrale Anforderung und umfasst mehrere Bereiche. Innerhalb des EAN Systems muss der Einsatz neuer Funkmodule eines bereits integrierten Standards gewährleistet werden. Speziell im Zuge von Weiterentwicklungen und der Umsetzung neuer Projekte ist dies von besonderer Wichtigkeit, da die Verfügbarkeit der verbauten Komponenten teils stark schwankt. Sofern die alternativen Module schnittstellenkompatibel sind und über den nötigen Funktionsumfang verfügen, ist mit keinen Komplikationen zu rechnen.

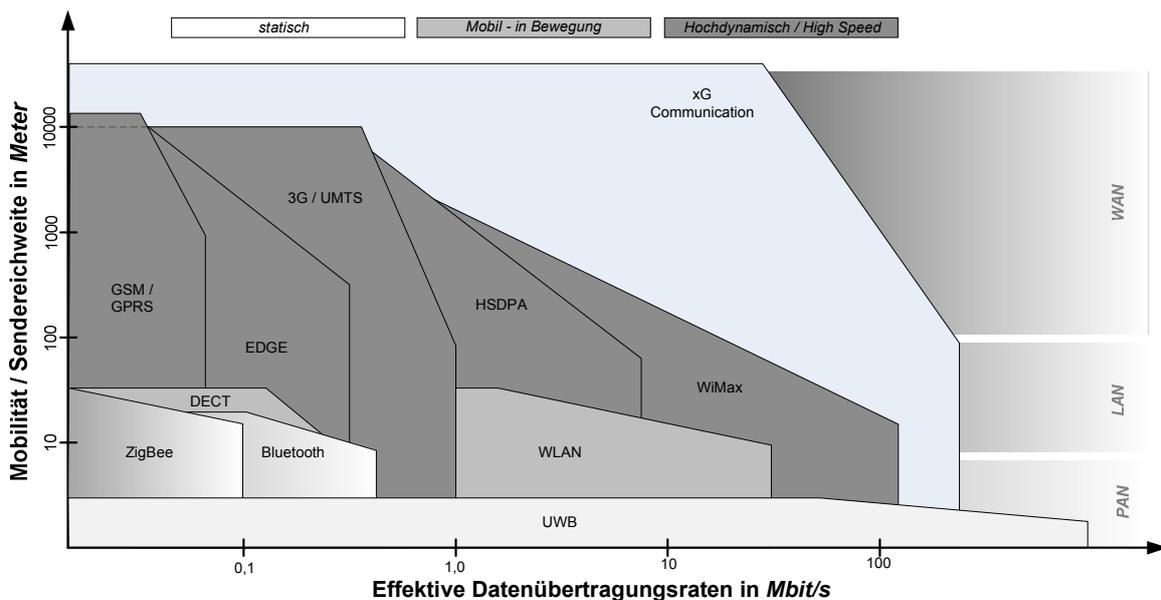


Abbildung 4.11.: Einordnung aktueller, drahtloser Kommunikationstechnologien hinsichtlich der nutzbaren Datenrate und der Einteilung in die Sendebereiche. Zusätzlich werden auf der Y-Achse auch Einschränkungen durch unterschiedliche Bewegungsmodelle integriert.

4. *Embedded Ambient Networking*

In Folge des technologischen Fortschritts kommen zusätzlich auch neue Kommunikationstechnologien zum Einsatz. Um das EAN Konzept langfristig effizient nutzen zu können, ist die modulare Erweiterung mit neuen Funkstandards essentiell. Betrachtet man die Fülle bereits verfügbarer Standards in Abbildung 4.11, wird diese Notwendigkeit deutlich.

Im Rahmen der Arbeit *Methoden zur modularen Erweiterung funkstandardübergreifender Hardware* [14] wurden aus diesem Grund mögliche Konzepte analysiert und bewertet. Primäres Ziel war dabei eine möglichst einfache Integration neuer Hardware. Durch ein geeignetes Konzept soll die Wissensbasis des EAN Systems für die prozedurale Verarbeitung neuer Funkmodule erweitert werden. Grundlegende Funktionalitäten umfassen dabei den Verbindungsauf- und -abbau sowie die Überwachung aktiver Kommunikationskanäle und die Suche nach Geräten in Funkreichweite. Zwei Lösungsansätze erscheinen an dieser Stelle hinreichend.

Formale Beschreibungstechniken

Protokolle können durch sogenannte *FDT*⁴ formal beschrieben und spezifiziert werden. Diese stellen eine eindeutige Interpretation des gesamten Kommunikationsprozesses hinsichtlich Syntax und Semantik sicher. Hier könnten etablierte Verfahren wie *Estelle*, *LOTOS* und *SDL*⁵ zum Einsatz kommen.

Die aktuell etablierten, kommerziellen Funkstandards wurden jedoch nicht formal spezifiziert. Somit ist eine nachträgliche, formale Beschreibung auf Basis von Zustandsdiagrammen/-tabellen und den zeitlichen Abläufen nötig. Durch die teils enorme Komplexität der zu integrierenden Funkstandards ist der Arbeitsaufwand hierfür kaum abzuschätzen. Desweiteren benötigt die Interpretation innerhalb des Gesamtsystems entsprechende Hardwareressourcen, welche die Leistungsfähigkeit herkömmlicher Mikrocontroller meist übersteigt.

Hardwarespezifische Ansteuerung

Da der Einsatz formaler Beschreibungen innerhalb eines EAN Systems nicht praktisch umsetzbar ist, wird ein Lösungsansatz auf Basis einer hardwarespezifischen Ansteuerung der Funkmodule favorisiert. Dabei werden die konkreten Spezifikationen der einzelnen Kommunikationsstandards weitestgehend vernachlässigt. Anstatt dessen kommen die von den Hardwareherstellern zur Verfügung gestellten APIs zum Einsatz. In den meisten Fällen ermöglicht dies eine unkomplizierte Nutzung der Funkhardware. Der Befehlssatz ist überschaubar und die Funktionalität bereits umfassend getestet. Ein standardisiertes Interface vereinfacht die Einbindung der Hardware in das EAN System.

⁴Formal Description Techniques

⁵Specification and Description Language

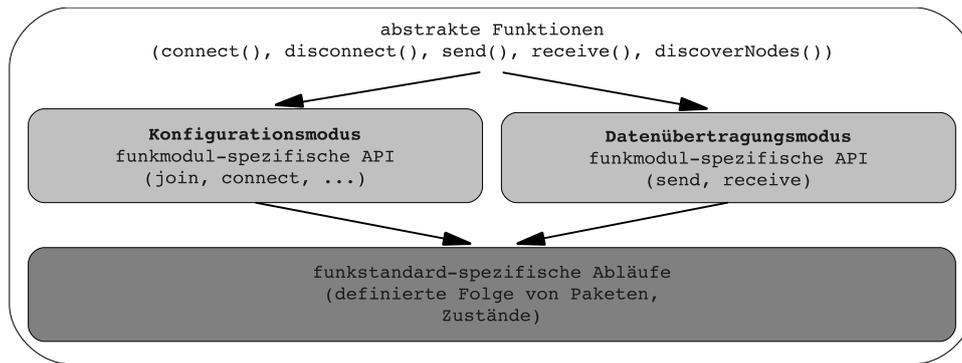


Abbildung 4.12.: Hardware-spezifische Ansteuerung der Funkmodule. Im Zuge einer Integration neuer Funkmodule werden die abstrakten Kommunikationsbefehle auf hardware-spezifische Befehle abgebildet.

Aufbauend auf den modulspezifischen Befehlssätzen folgt die Einführung abstrakter API-Funktionen für das Verbindungsmanagement, bspw. `connect()`, `disconnect()`, `send()`, `receive()` und `discoverNodes()`, welche einen typischen Kommunikationsablauf repräsentieren. Diese Funktionen werden einmalig abstrakt definiert und im Anschluss funkmodul- bzw. funkstandard-spezifisch implementiert. Im Zuge der Implementierung bilden sie auf eine definierte Abfolge von Systembefehlen ab, welche sich in Datenübertragungs- sowie Konfigurationsaufrufe unterteilen (Abbildung 4.12). Die in den Funkmodulen verbaute Controller-Hardware übernimmt alle prozeduralen Abläufe auf Hardwareebene und gewährleistet somit eine protokollkonforme Nutzung der drahtlosen Schnittstellen. Auf EAN Systemebene entfällt das aufwendige Testen komplexer Abläufe innerhalb der Funkhardware. Im Betrieb stehen dem EAN Kern Kontroll- und Kommunikationskanäle zu den verbauten Modulen zur Verfügung.

Im Zuge der praktischen Realisierung eines EAN-konformen Systems betrifft die Erweiterbarkeit auch die Anzahl verfügbarer Sockel-Schnittstellen zur Anbindung geeigneter Module. Die Struktur des EAN Systems ermöglicht hierbei, sofern sinnvoll, eine modulare Erweiterung der Schnittstellenmenge. Dabei ist die Implementierung neuer Interface Handler notwendig, welche die prozedurale Verarbeitung der zusätzlichen Module übernehmen und diese in die Gesamtstruktur einbinden.

Für spätere Aktualisierungen des EAN Kerns sind unterschiedliche Verfahren für die drahtlose Rekonfiguration umsetzbar. Ein vielversprechender Lösungsansatz wird in der Diplomarbeit *Konfiguration heterogener drahtloser Sensornetzwerke* [19] vorgestellt und umgesetzt. Im Rahmen dieser Arbeit soll auf eine dynamische Rekonfiguration jedoch nicht näher eingegangen werden.

4.3.2. Konzeptionelle Anforderungen

Im Zuge der Bereitstellung aller EAN Funktionalitäten treten drei grundlegende Problemstellungen auf. Für die Umsetzung eines EAN-konformen Kommunikationssystems müssen diese konzeptionellen Anforderungen durch entsprechende Lösungsansätze erfüllt werden.

4.3.2.1. **Verbindungsmanagement / Topologieaufbau**

Die erste von insgesamt drei wesentlichen Problemstellungen im Rahmen der EAN Konzeption bezieht sich auf den Aufbau einer geeigneten Topologie und das damit gekoppelte Verbindungsmanagement innerhalb der Knoten. Im Gegensatz zu den bisherigen Einsatzgebieten des IFB Konzeptes stehen in den betrachteten Szenarien keine statischen, drahtgebundenen Schnittstellen zur Verfügung. In einer sich dynamisch ändernden, drahtlosen Netzwerkstruktur existieren weitaus komplexere Schnittstellen und Übertragungsprotokolle.

Das Verbindungsmanagement muss die Interaktion über diese Schnittstellen ermöglichen. Der elementare Funktionsumfang kann wie folgt aufgeschlüsselt werden:

- Suche nach verfügbaren Knoten in Sendereichweite
- Abfrage des Verbindungsstatus
- Aufbau einer neuen Verbindung
- Konfiguration funkstandardspezifischer und funkmodulspezifischer Parameter
- Verarbeitung von ausgehandelten Verbindungsidentifikatoren *Connection Handles / Connection ID*
- Schließen eines aktiven Kommunikationskanals

Bei einer anstehenden Protokollkonvertierung muss zunächst ein gültiger Kommunikationskanal initialisiert und somit der Zugriff auf das Medium sichergestellt werden. Der Datenverarbeitungsprozess muss dazu angepasst und zeitlich koordiniert werden. Die auftretenden Verzögerungen repräsentieren dabei einen kritischen Faktor. Im Falle eines exklusiven Verarbeitungszyklus muss die Paketweiterleitung dazu unterbrochen werden. Alternativ könnte dieser Problematik durch mehrere parallel arbeitende Verarbeitungsinstanzen entgegengewirkt werden. An dieser Stelle sind jedoch höhere Hardwareressourcen erforderlich.

Topologieoptimierung / -minimierung

Das Verbindungsmanagement ist auch für Optimierungen der Topologie aus globaler Sicht verantwortlich. Eine Einschränkung der Anzahl aktiver Kommunikationskanäle verkürzt dabei die Latenz im Zuge der Routenfindung und vermindert das Interferenzniveau, wodurch die Verbindungsqualität gesteigert werden kann.

Diese Forderung wird besonders in hochskalierten Netzwerkstrukturen deutlich, bei denen der Vernetzungsgrad je nach Knotendichte schnell einen kritischen Wert erreicht. Dabei werden sowohl interferenzbedingte Störeinflüsse als auch die administrative Verwaltung aller aktiven Kommunikationspfade zu wesentlichen Problemstellungen. Gleichzeitig steigt in diesem Zusammenhang der Energieverbrauch erheblich an, was die Betriebsdauer unnötig reduziert.

Folglich sieht das EAN Konzept die Integration entsprechender Topologieoptimierungsalgorithmen vor, welche der Problematik entgegenwirken. Welches Verfahren gewählt wird, ist dabei

frei definierbar. In Kapitel 3.1.1 wurden mehrere Algorithmen vorgestellt. Die Verfahren arbeiten dabei auf Basis einer dynamischen Anpassung der Sendeleistung, auf Graphenalgorithmien zur Kantenminimierung oder mittels selektiver Vernetzung. Auch ein Beam-Forming mit Hilfe von Smart Antenna Systemen (siehe Kapitel 3.2.3) ist an dieser Stelle möglich, um die Topologie zu optimieren. Je nach Anwendungsszenario fällt die Wahl dabei auf unterschiedliche Ansätze. Ein wesentliches Entscheidungskriterium stellt der gewünschte Grad an Redundanz in der Topologie dar. Aber auch die verfügbaren Energie- und Hardwareressourcen sind zu beachten. Aus diesen Anforderungen ergeben sich Einschränkungen in Bezug auf die Menge möglicher Verfahren. Innerhalb eines EAN Systems sind jedoch leichtgewichtige und laufzeitoptimierte Algorithmen zu bevorzugen. Hier empfiehlt sich eine selektive Anpassung der Sendeleistung [146], sofern die Systemhardware dies ermöglicht, oder auch das LMST-Verfahren [108].

4.3.2.2. Routing / Forwarding

Die zweite Problemstellung innerhalb des EAN Konzeptes stellt das Routing in der geschaffenen, heterogenen Netzwerktopologie dar. Erst wenn eine hinreichend stabile Pfadsuche gewährleistet wird, ist eine Kommunikation durch die geschaffene Netzinfrastruktur möglich.

Die eingesetzten Verfahren müssen dabei mehrere Verbindungen zwischen zwei benachbarten Knoten auf Basis unterschiedlicher Funkstandards verwalten und gewichten. Neben dieser grundlegenden Integration mehrerer Funkstandards steht aber auch die Optimierung der Routenpfade je nach Anforderung an den Verbindungskanal im Fokus der Betrachtungen. Desweiteren sind speziell im Bereich mobiler Ad Hoc Netzwerke Störeinflüsse zu berücksichtigen und durch robuste Verfahren zu minimieren. Wesentliche Anforderungen werden im Folgenden kurz aufgeschlüsselt.

RIP Standardkonforme Paketstruktur

Routingfähigen Komponenten in der Topologie können in zwei Typen unterteilt werden. Einerseits existieren EAN-konforme Multistandard-Knoten mit entsprechender Gateway-Funktionalität, deren Aufbau im bisherigen Verlauf dieses Kapitels bereits beschrieben wurde. Auf der anderen Seite nutzen herkömmliche Routerkomponenten meist RIP [84] oder OSPF als standardisiertes, einfaches Routingverfahren mit hop-optimierter Metrik. Der Aufbau ist denkbar einfach und beinhaltet Adressfamilien mit den dazugehörigen Metrikwerten, welche die Anzahl benötigter Hops von der Quelle zur Datensinke repräsentieren. Abbildung 4.13 veranschaulicht den Aufbau von RIP (Version 1).

Ziel des im Rahmen dieser Arbeit entwickelten Routingverfahrens ist es, den dargestellten Paketaufbau von RIP unverändert zu übernehmen. Anpassungen und Optimierungen erfolgen ausschließlich über die Berechnungsfunktion und die gewählte Metrik. Wertebereich sowie Struktur der Routing-Pakete sind weiterhin protokollkonform. Somit kann in gemischten Topologien mit Multistandard-Routingknoten als auch herkömmlichen Routing-Komponenten eine Pfadsuche sichergestellt werden. Eine Optimierung der Routenpfade ist bei einem solchen Mischbetrieb jedoch nicht vollständig gewährleistet, da sich die semantische Interpretation der Metrik-

4. Embedded Ambient Networking

werte in den jeweiligen Netzwerkknoten unterscheidet.

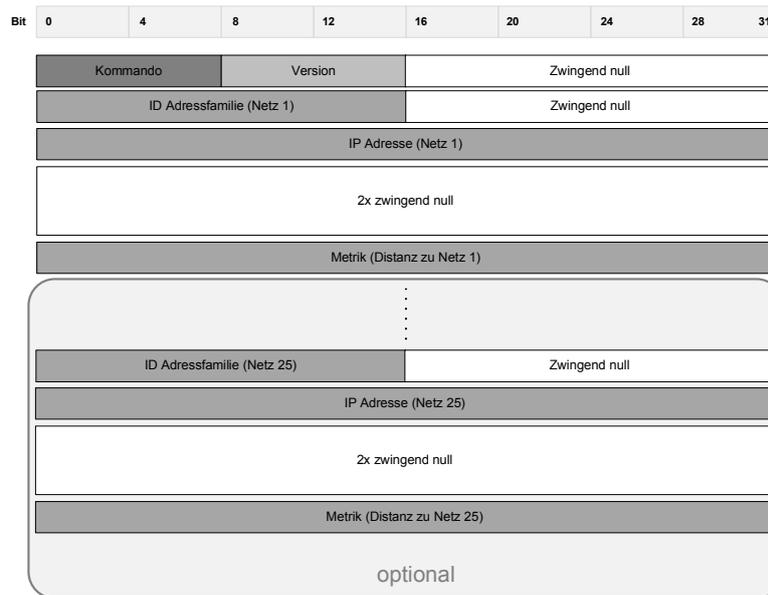


Abbildung 4.13.: Aufbau eines Paketes nach der Routing Information Protocol (RIP - Version 1) Definition.

Interferenz- und Lastoptimierung

Eine weitere Zielstellung für das eingesetzte Routingverfahren repräsentiert die Minimierung des Interferenzniveaus in der Topologie. Interferenzen und die damit verbundenen Störungen treten in nahezu allen drahtlosen Kommunikationsnetzen auf. Speziell bei zunehmender Knotendichte und in hochdynamischen Netzstrukturen stellen diese Störeinflüsse eine signifikante Messgröße dar.

Das vorgestellte EAN Konzept kann diese Interferenzen weder vollständig eliminieren noch aktiv beeinflussen. Wirksame Mittel, das Störpotential passiv zu beeinflussen, ist die bereits erläuterte Topologieoptimierung sowie die Senkung der Netzlast. In Folge dessen sinkt auch die Wahrscheinlichkeit für Kollisionen auf dem Medium. Da die zu übertragende Menge an Nutzdaten im Netzwerk als konstant vorausgesetzt wird, beschränken sich die Überlegungen auf leichtgewichtige Protokolle mit geringem administrativen Overhead. Die Menge benötigter Status- und Kontrollpakete soll dabei auf ein Minimum reduziert werden.

Routingprotokolle verursachen einen erheblichen Datenoverhead in MANETs mit stark variierenden Knotenstrukturen. Dies trifft primär auf proaktive Verfahren zu, welche kontinuierlich eine aktuelle Routentabelle pflegen. Aber auch reaktive Ansätze stoßen in hochdynamischen Topologien schnell an ihre Grenzen und können nicht mehr effizient eingesetzt werden.

Anders als bei herkömmlichen Routenprotokollen sollen im Rahmen des EAN Konzeptes neben den verfügbaren Netzwerkinformationen auch erweiterte Systeminformationen, wie Ladezustand, Signalstärke oder auch Signalqualität für die Routenoptimierung mit einbezogen werden. Die Routen-Metrik wird durch diese zusätzlichen Entscheidungskriterien deutlich verfeinert.

Durch den sinnvollen Einsatz von Caching-Mechanismen und der aktiven Vermeidung von

Broadcast-Storms (siehe 3.1.2 und [88, 127, 182]) kann das Interferenzlevel in der Topologie nochmals reduziert werden. Die messbaren Kenngrößen für Störungen durch Interferenzen sind dabei die Paketverlustrate, die Signalqualität und unter Umständen die Paketlaufzeit. Die Detektion solcher kritischer Parameter erlaubt einen Rückschluss auf das aktuelle, lokale Interferenzniveau. Angepasste Routingverfahren könnten hier einen bewussten Wechsel des genutzten Netzwerkinterfaces anstoßen und folglich aktiv zu Optimierungen in Hinblick auf interferenzbedingte Störungen beitragen. Integriert eine EAN Kommunikationsplattform mehrere Funkmodule auf Basis unterschiedlicher Frequenzbänder, können bei Störungen alternative Frequenzbereiche genutzt werden. Ein einfaches EAN System koppelt hierfür z.B. WLAN 802.11a Module im 5 GHz Bereich, Bluetooth 802.15.1 auf 2,4 GHz sowie Zigbee Module, die auf dem UHF Band bei 868-870 MHz arbeiten.

Pfadoptimierung

Die geschaffene Netzinfrastruktur enthält dabei unter Umständen mehrere Verbindungskanäle zwischen zwei Knoten auf Basis unterschiedlicher Funkstandards. Das EAN System und die gewählten Routing-Verfahren müssen in der Lage sein, diese Zuordnung einer Menge von parallelen Verbindungskanälen auf eine IP Adresse zu verwalten (Abbildung 4.14).

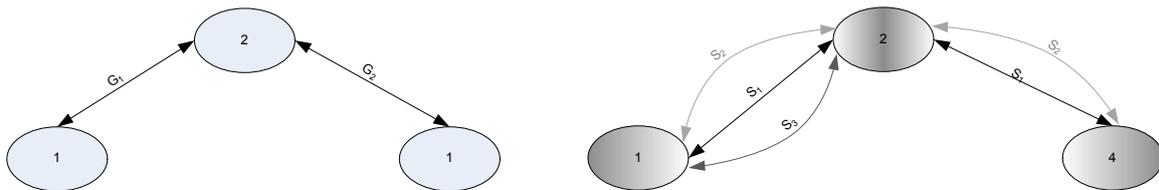


Abbildung 4.14.: Konzeptionelle Unterschiede zwischen herkömmlichen Routingverfahren in drahtgebundenen Netzen (bspw. RIP) mit jeweils einzelnen Kanten zwischen den Knoten (Abbildung links - Gewichtung RIP: $G_x = 1$) und erweiterten Routingkonzepten mit Unterstützung redundanter Verbindungskanäle sowie lastbasierten, mehrdimensionalen Metriken. Hierzu gehört auch das EAN Routingkonzept auf der Grundlage unterschiedlicher Funkstandards (Abbildung rechts; Funkstandards $S_1 \dots S_3$). Die Modellierung und die Entscheidungsfindung müssen dementsprechend angepasst werden.

Soll nun die Auswahl des Kommunikationskanals nutzer- oder anwendungsseitig beeinflusst werden, müssen Methoden gefunden werden, diese Metainformationen von der Applikationsebene in die Controller Hardware des EAN Systems zu übertragen. Mögliche Lösungen, die auf TOS Informationen des IPv4 Protokolls oder auf QoS Funktionalitäten von IPv6 aufsetzen, wurden bereits in Abschnitt 4.3.1.3 erläutert. Der konzipierte Routingalgorithmus kann diese Metadaten verarbeiten und die Routenberechnung dynamisch zur Laufzeit anpassen.

Kooperative Verfahren zur Optimierung der Energieressourcen

Das vorgestellte EAN Konzept soll neben der Erreichbarkeit primär den Energiebedarf optimieren. Dabei ist zwischen lokalen Optimierungen im jeweiligen Knoten und globalen Optimierungen in der gesamten Topologie zu unterscheiden. Lokale Optimierungen sind vor allem durch eine gezielte, selektive Nutzung der verfügbaren Kommunikationsschnittstellen möglich. Bei der Übertragung geringer Datenmengen auf kurzen Sendedistanzen kann hier bspw. ein

4. *Embedded Ambient Networking*

niederenergetischer Nahbereichs-Funkstandard bevorzugt werden. Dem Optimierungspotenzial steht ein erhöhter Grundenergiebedarf mehrerer simultan betriebener Funkmodule entgegen. Folglich muss geprüft werden, ob die Energiebilanz positiv ausfällt und der Einsatz sinnvoll erscheint.

Die Situation ist aber auch aus globaler Sicht zu beurteilen. Stehen mehrere redundante Multihop-Kommunikationspfade von der Quelle zur Senke zur Verfügung, kann in Abhängigkeit von den Verbindungsanforderungen und dem aktuellen Verbindungsstatus ein optimierter Pfad gewählt werden. Hierbei können die Energieniveaus der Netzwerkknoten mit in die Berechnungen einfließen.

Es ist nun einerseits möglich, Knoten mit kritischem Ladezustand zu schonen, indem Routenpfade über Netzwerkknoten mit hinreichenden Energiere Ressourcen bevorzugt werden. Durch diesen kooperativen Ansatz steigt die durchschnittliche Laufzeit der Netzwerkknoten und folglich stehen diese für die Datenweiterleitung länger zur Verfügung.

Desweiteren können durch die Verwendung eines alternativen Kommunikationsstandards je nach Situation mehrere Hops übersprungen werden. Die Last und der Energiebedarf in diesen Knoten sinkt. Aus globaler Sicht steht also ein erhöhter Grundenergiebedarf mehrerer Funkmodule dem Optimierungspotential durch verkürzte Routenpfade gegenüber. Qualitative Aussagen hinsichtlich des Optimierungspotentials sollen mit Hilfe einer geeigneten Simulationsumgebung in Kapitel 5 evaluiert werden.

4.3.2.3. Protokollkonvertierung

Ein dritter zentraler Punkt im Rahmen der Umsetzung des EAN Konzeptes stellt die Protokollkonvertierung dar. Im Gegensatz zu ursprünglichen IFB Konvertierungsszenarien weisen die hier integrierten Netzwerkprotokolle eine wesentlich höhere Komplexität auf. Speziell im Bereich der drahtlosen Kommunikationstechnologien verfügen die jeweiligen Funkmodule über umfangreiche Schnittstellen. Dementsprechend muss auch der Konvertierungsprozess innerhalb des EAN Konzeptes an diese komplexen Abläufe angepasst werden.

Protokollkonforme Daten

Die Konvertierung muss eine vollständige Weiterleitung aller eingehenden Daten auf Basis eines gewählten Kommunikationsstandards gewährleisten. Alle für die Konvertierung nötigen Informationen müssen dabei entweder durch Instanzen der Controller-Hardware oder durch das Verbindungsmanagements zur Verfügung stehen. Alternativ müssen anzupassende Protokollelemente dynamisch berechenbar sein.

Für die reibungslose Kommunikation reicht die einfache Kapselung der Nutzdaten in einen protokollkonformen Rahmen unter Umständen nicht aus. Je nach Protokollspezifikation muss hierbei die maximale Paketgröße berücksichtigt werden. Übersteigt die Größe den Maximalwert, muss das Paket im Zuge der Konvertierung in entsprechend kleine Segmente unterteilt und sukzessiv übertragen werden. Für den Datenverarbeitungsprozess stellt diese Segmentie-

rung in Abhängigkeit vom gewählten Zielprotokoll somit eine wichtige Aufgabe dar.

Minimierung des Protokoll-Overheads

Sollen die Latenzen im Zuge einer Paketkonvertierung optimiert werden, repräsentiert der anfallende Protokoll-Overhead einen wichtigen Faktor. Dieser entsteht hauptsächlich durch die schlechte Segmentierung der Datenpakete während der Verarbeitung. Dies führt in einigen Fällen zu einer suboptimalen Aufteilung der Daten (Abbildung 4.15). Daraus resultiert eine erhöhte Netzlast, was die Verarbeitungseinheiten zusätzlich belastet. Im Umkehrschluss könnten mit entsprechendem Aufwand auch mehrere kleine Pakete eines Funkstandards *A* in einem größeren Datenpaket des Standards *B* zusammengeführt werden.

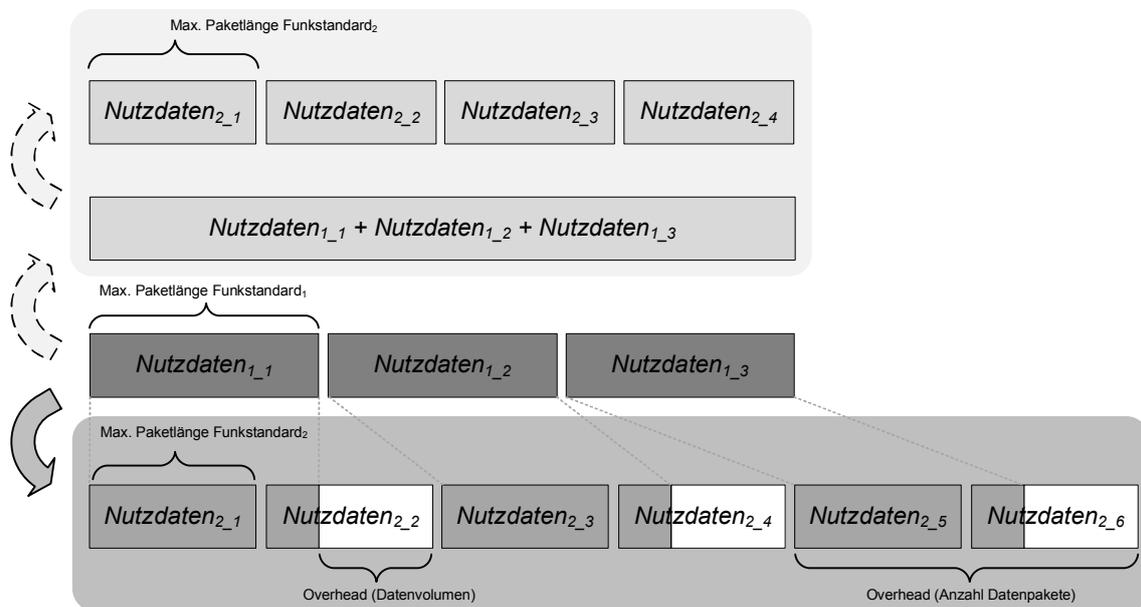


Abbildung 4.15.: Erzeugter Protokoll-Overhead durch eine nötige Segmentierung innerhalb des Konvertierungsprozesses. Die Controller-Hardware verarbeitet sequenziell einzelne Pakete als gekapselte Einheiten (Bild unten), wodurch sich die Anzahl zu übertragender Datenpakete zusätzlich erhöht. Wünschenswert wäre hier eine Nutzdaten-Pufferung mehrerer, aufeinanderfolgender Datenpakete und eine anschließende, effizientere Konvertierung in das Zielprotokoll (Bild oben), was jedoch erhöhte Latenzen zur Folge hätte. Bei Protokollen mit fester Paketlänge erzeugt die Konvertierung unter Umständen erheblichen Overhead innerhalb der einzelnen Datenpakete.

Dies erfordert jedoch die Verarbeitung mehrerer, zusammenhängender Pakete. Der Inhalt wird dabei zunächst zusammengefasst und anschließend effizient neu aufgeteilt. Neben dem erhöhten Speicherbedarf und steigenden Übertragungslatenzen resultiert aus diesen komplexen Abläufen ein erheblicher Mehraufwand.

Aufgrund der begrenzten Hardwareressourcen in einem EAN System sollen die Abläufe während der Paketverarbeitung jedoch möglichst kompakt gestaltet werden. In Folge dessen ist diese Problematik nur schwer handhabbar.

4. Embedded Ambient Networking

Latenzoptimierte Konvertierung

Als kritisch sind auch die auftretenden Verzögerungen im Zuge der Protokollkonvertierung einzuordnen. Da die Datenpakete ausschließlich sequenziell verarbeitet und analysiert werden, führt die Abarbeitung durch die jeweiligen Zustandsautomaten zu Latenzen. Bei mehreren, simultan agierenden Funkmodulen unter Last ist der notwendige Verarbeitungsaufwand nicht zu unterschätzen.

Um dieser Problematik entgegenzuwirken, ist die verwendete Controller-Hardware entsprechend zu dimensionieren.

4.4. Konzeptumsetzung

Für die Umsetzung der EAN Funktionalität müssen die konzeptionellen Anforderungen zwingend durch entsprechende Lösungsansätze erfüllt werden. Die damit verbundenen Problemstellungen umfassen die eigentliche Protokollvertierung sowie das Verbindungsmanagement und das Routing. Das geschaffene Gesamtsystem realisiert dabei die vorgestellten, nutzerseitigen Anforderungen.

4.4.1. IFB Protokollkonvertierung

Der Kern des hier vorgestellten Konzeptes basiert auf den wissenschaftlichen Arbeiten der Dissertation *Modeling and Automated Synthesis of Reconfigurable Interfaces* [90]. Wie in Kapitel 2.4 bereits beschrieben, setzt die Schnittstellensynthese dabei auf einem zentralen Hardwareblock auf, welcher verschiedene, inkompatible Schnittstellen koppelt und so die Kommunikation der angeschlossenen Tasks ermöglicht. Der ursprüngliche Ansatz auf Basis eines solchen *IFBs*⁶ betrachtet stets eine konsequente Trennung zwischen prozeduralen und syntaktischen Aspekten im Zuge der Protokollverarbeitung.

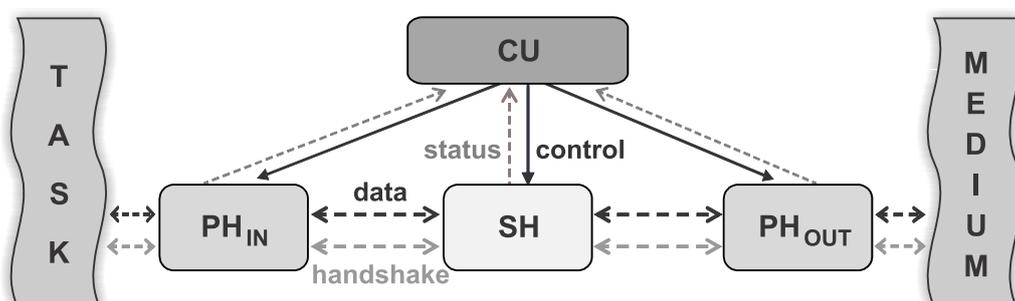


Abbildung 4.16.: Funktionaler Ablauf innerhalb eines IFB mit eingehender Paketverarbeitung des anliegenden Tasks über den *Protocol Handler PH_{IN}*. Anschließend erfolgt die Nutzdatenkonvertierung mit Hilfe des *Sequence Handler SH* und das protokollkonforme Versenden der Daten auf das Zielmedium. Die *Control Unit (CU)* steuert dabei den gesamten Ablauf.

⁶Interface Block

Es erfolgt eine Unterteilung in Protocol Handler (PH) und Sequence Handler (SH), welche über eine Koordinierungsinstanz, der Control Unit (CU), miteinander verbunden sind (Abbildung 4.16). Dabei sind die jeweiligen Protocol Handler für die prozedurale Abwicklung der angeschlossenen Schnittstellen verantwortlich. Die protokollspezifischen Datenpakete werden durch diese korrekt empfangen und versendet. Die Sequence Handler bereiten im Gegensatz dazu die Nutzdaten für das Zielprotokoll auf.

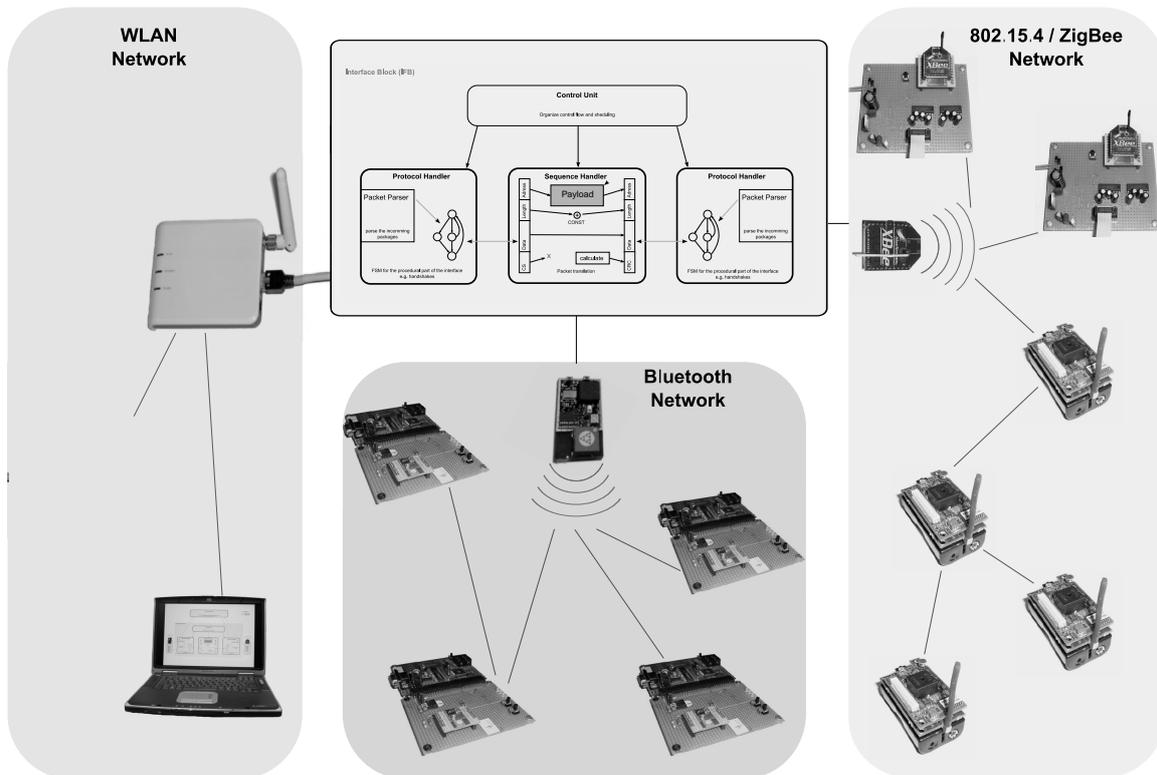


Abbildung 4.17.: Funkstandardintegration auf Basis des IFB Konzeptes. Ziel ist eine hardwarenahe Kopplung unterschiedlicher Funkmodule in Verbindung mit einer direkten Konvertierung zwischen den jeweiligen Protokollen. Hier exemplarisch dargestellt die Hardwareplattform.

Das hier vorgestellte EAN Konzept fokussiert jedoch Protokolle für die drahtlose Netzwerkkommunikation (Beispielszenario Abbildung 4.17), deren Komplexität sich deutlich von den bisher adaptierten, statischen Protokollen unterscheidet. Dabei bestehen vielfach Abhängigkeiten zwischen Paketaufbau und Kommunikationsablauf. Betrachtet man Übertragungsprotokolle mit variabler Paketlänge, wird die Problematik schnell deutlich. Hier müssen Parameter in den Headerdaten der Pakete, wie z.B. die Nutzdatenlänge oder entsprechende Prüfsummen, dynamisch während der Protokollkonvertierung erzeugt bzw. adaptiert werden. Beim Einlesen eines entgegengenommenen Datenpaketes wird dazu zunächst das Längenfeld im Header interpretiert, bevor das Paketende korrekt erkannt werden kann. Gleiches gilt für Generierung passender Bestätigungspakete, sofern dies erforderlich ist. In den Protocol Handlern ergibt sich folglich eine deutlich höhere Verarbeitungscomplexität.

Der PH hat neben der standardkonformen Abwicklung aller Protokollabläufe mehrere Aufgaben im Rahmen des Konvertierungsprozesses. Die wesentlichen Elemente einer syntaktischen

4. Embedded Ambient Networking

Paketspezifikation sollen daher analysiert werden:

Delimiter sind Sonderzeichen oder auch eine spezielle Abfolge von Zeichen, durch welche Paketgrenzen erkannt werden. Für die Protocol Handler ist die korrekte Auswertung essentiell, um aufeinanderfolgende Pakete eindeutig abgrenzen zu können.

Escape Sequenzen und Paketlänge ermöglichen es, die für Delimiter reservierten Sonderzeichen trotzdem als Teil des Zeichenalphabetes für die Übertragung zu verwenden. Sie werden dabei durch den Protokollautomaten automatisiert auf spezielle, vordefinierte Zeichenfolgen abgebildet. Da diese Repräsentationen das Datenpaket verlängern und dies Auswirkungen auf die Länge der zu übertragenden Nutzdaten hat, müssen sie ebenfalls vom PH bearbeitet werden. Daraus ergibt sich folglich auch die Notwendigkeit, *Längenangaben* korrekt zu interpretieren. Das Setzen dieser Längenangaben kann jedoch auch im Sequence Handler vorgenommen werden, da hier bereits alle nötigen Informationen, d.h. Gesamtlänge der Nutzdaten exklusive Escapesequenzen, vorliegen.

Auch die *Sequenznummern*, welche häufig für die Bestätigung empfangener Datenpakete zum Einsatz kommen, sollten effizient durch die Protocol Handler verarbeitet werden, um den Verbindungsaufbau oder die Routensuche nicht unnötig zu verzögern. Die Generierung der Acknowledgement-Pakete erfolgt dabei im Protocol Handler des Verbindungsmanagements.

Prüfsummen dienen der Sicherung der Datenübertragung und werden meist über Header- und Nutzdaten gebildet. Da einige Headerinformationen jedoch erst im Protocol Handler zur Verfügung stehen, kann auch die Prüfsumme erst hier berechnet und gesetzt werden.

Identifikatoren zur Erkennung des Pakettyps sind für den Protocol Handler ebenfalls von Relevanz, da spezielle Kontrollpakete ggf. direkt an das Verbindungsmanagement weitergeleitet werden müssen. Im Falle einer Konvertierung durchlaufen diese Pakete jedoch nicht den herkömmlichen Verarbeitungsprozess und müssen separat betrachtet werden.

Ausschließlich vom Sequence Handler verarbeitete Datenfelder sind bspw. *Adressen*, *Nutzdaten* sowie *Konstanten* (z.B. Flags oder Versionsnummern) und ggf. *Konfigurationen*. Hierbei ist ein minimaler Aufwand zu veranschlagen.

Die vorangegangenen Erkenntnisse zeigen, dass viele Paketelemente exklusiv durch die Protocol Handler analysiert und verarbeitet werden müssen. Der Funktionsumfang und die Komplexität des Sequence Handlers beschränkt sich auf ein Minimum.

Im ursprünglichen IFB Ansatz war ein wesentlicher Grund für diese Funktionstrennung der parallele Betrieb aller Funktionseinheiten im Zuge der Hardwareumsetzung auf einem FPGA. Auch die Realisierung des hier vorgestellten EAN Konzeptes ist mit einer strikten Trennung von SH und PH konzeptionell möglich, sofern wie in Abbildung 4.18 dargestellt, eine zusätzliche Komponente das Verbindungsmanagement kapselt und den Zugriff auf das Medium sichert.

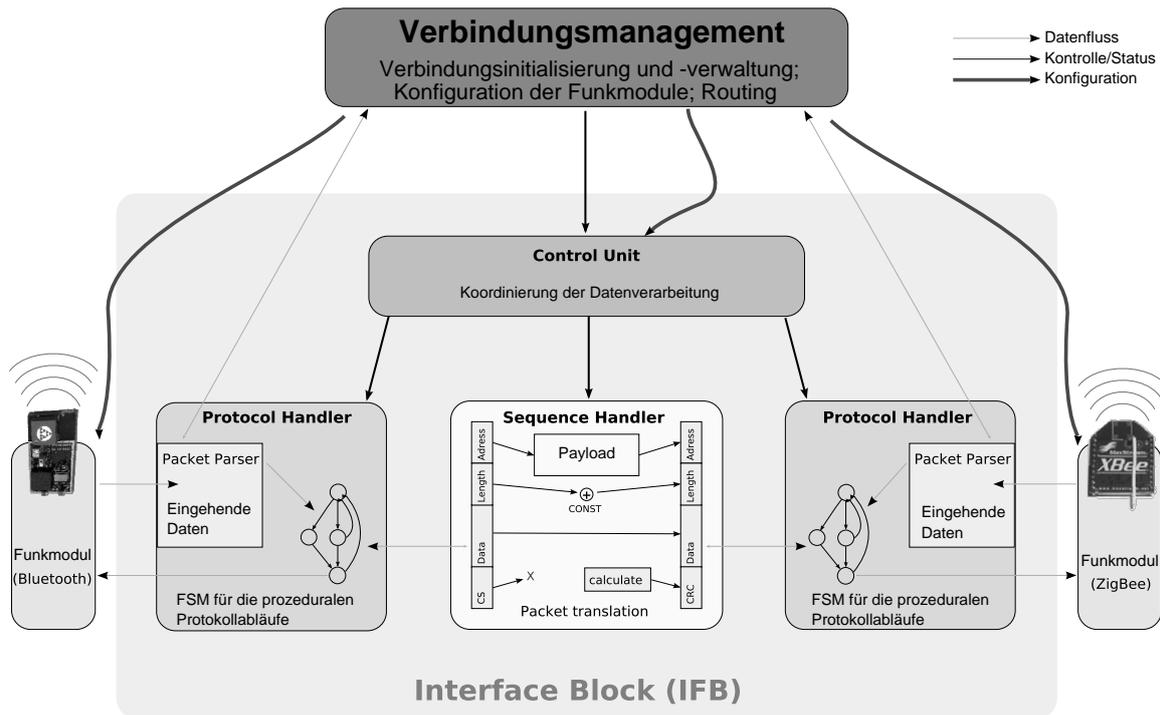


Abbildung 4.18.: Anwendung des IFB Konzeptes auf drahtlose Netzwerkschnittstellen. Ein Verbindungsmanagement muss dazu alle relevanten Informationen bzgl. Quell- und Zieladressen sowie die Stati aktiver Kommunikationskanäle verwalten. Bei Bedarf können hardwarespezifische Steuerkommandos an die einzelnen Funkmodule gesendet werden.

4.4.2. Extended Interface Block (xIFB)

Der im Rahmen dieser Arbeit vorgestellte Ansatz soll jedoch in Software umgesetzt werden. Eine Mikrocontroller-Implementierung mit μ -Betriebssystem und entsprechender Software-Verarbeitungskomponenten ist dabei deutlich einfacher realisierbar und bietet eine flexiblere Ausgangsbasis für modulare Erweiterungen. An dieser Stelle ist eine Integration von PH und SH in eine umfassende Verarbeitungseinheit erheblich besser beherrschbar. Desweiteren ist eine FPGA Realisierung unverhältnismäßig teurer. Dieser wirtschaftliche Aspekt ist besonders bei einem Einsatz in großen Stückzahlen und in mobilen Endgeräten ein wichtiges Entscheidungskriterium.

Zentraler Punkt der IFB Anpassungen ist daher die Auflösung der bisherigen Aufgabenverteilung von Sequence Handler und Protocol Handler. Im Zuge der *Extended Interface Block (xIFB)* Erweiterung erfolgt hier eine komplette Integration sowohl von prozeduralen als auch von syntaktischen Protokolleigenschaften in ein überspannendes Element, welches als *Interface Handler (IH)* definiert wird (siehe Abbildung 4.19). In Hinblick auf das Softwaredesign ist dieser Schritt sinnvoll und hinreichend. Die einzelnen IH werden weiterhin durch eine zentrale Steuerinstanz koordiniert. Eine zusätzliche Komponente kapselt das Verbindungsmanagement für alle drahtlosen Kommunikationskanäle.

4. Embedded Ambient Networking

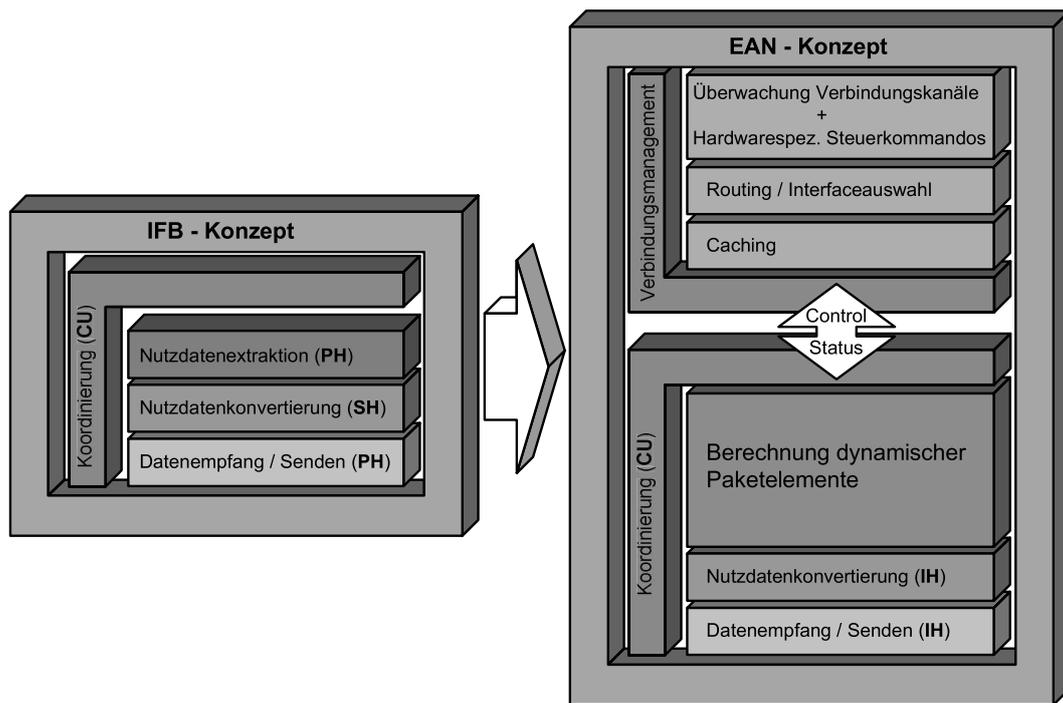


Abbildung 4.19.: Ursprüngliche IFB Funktionalität (links) und die Erweiterungen / Modifikationen im Rahmen dieser Arbeit (rechts). Drei wesentliche Punkte werden deutlich: 1) die Integration von Protocol Handler und Sequence Handler in einen umfassenden *Interface Handler (IH)*. 2) Die erweiterten Funktionalitäten zur korrekten Bearbeitung variabler Paketlängen sowie zur Generierung dynamischer Headerelemente. 3) Die gekapselte Funktionalität für das Verbindungsmanagement.

Das entstandene Design weist dabei Ähnlichkeiten zu herkömmlichen Routern auf, welche ebenfalls Daten zwischen verschiedenen Netzwerkschnittstellen und Teilnetzen auf Basis einer definierten Metrik weiterleiten. Bei genauer Betrachtung bestehen jedoch deutliche Unterschiede. Zum einen sind die IH generisch an das jeweilige Ursprungs- und Zielprotokoll adaptierbar. Diese Flexibilität ermöglicht schließlich die Konvertierung innerhalb verschiedener Protokollstacks und auf verschiedenen Protokollebenen. Die im Rahmen dieser Arbeit betrachteten, drahtlosen Kommunikationsschnittstellen unterscheiden sich dabei in den gesamten Protokollstacks. Herkömmliche Routerhardware könnte diese Kopplung nicht übernehmen.

Desweiteren kapselt das EAN Konzept das komplettes Verbindungsmanagement sowohl auf dem MAC Layer als auch auf Netzwerkebene in eine separate Komponente (siehe Abschnitt 4.4.3). Auf dem MAC Layer beinhaltet dies Prozesse für den funkstandardkonformen Verbindungsauf- und -abbau sowie eine automatisierte Überwachung aller aktiven Verbindungskanäle und ggf. auch Methoden zur Topologieoptimierung. Auf Netzwerkebene kommen angepasste Cross-Layer Routing-Funktionalitäten hinzu (siehe Abschnitt 4.4.4). Durch die Kapselung des Verbindungsmanagements bleibt die Komplexität der Interface Handler beherrschbar.

Die einzelnen Funkmodule verfügen jeweils über dedizierte Schnittstellen zur Ansteuerung auf Basis einer eigenständigen API. Für die Kopplung mehrerer solcher Funkmodule ist folglich kein Zugriff auf alle funkstandardspezifischen Protokolldetails notwendig. Der xIFB setzt direkt auf den Schnittstellen der Module auf und erlaubt eine hardwarenahe Integration.

Abbildung 4.20 verdeutlicht die integrierten Funktionalitäten des EAN Konzeptes in einem Gesamtschema. Die Segmentierung der jeweiligen Funktionsblöcke ist deutlich erkennbar. Sowohl das Host-System als auch die LCU werden über einen dedizierten IH angebunden. Gleiches gilt für die einzelnen Funkmodule. Jedes IH Modul kann voneinander unabhängig implementiert und bei Bedarf ausgetauscht werden. Die CU koordiniert den Verarbeitungsprozess.

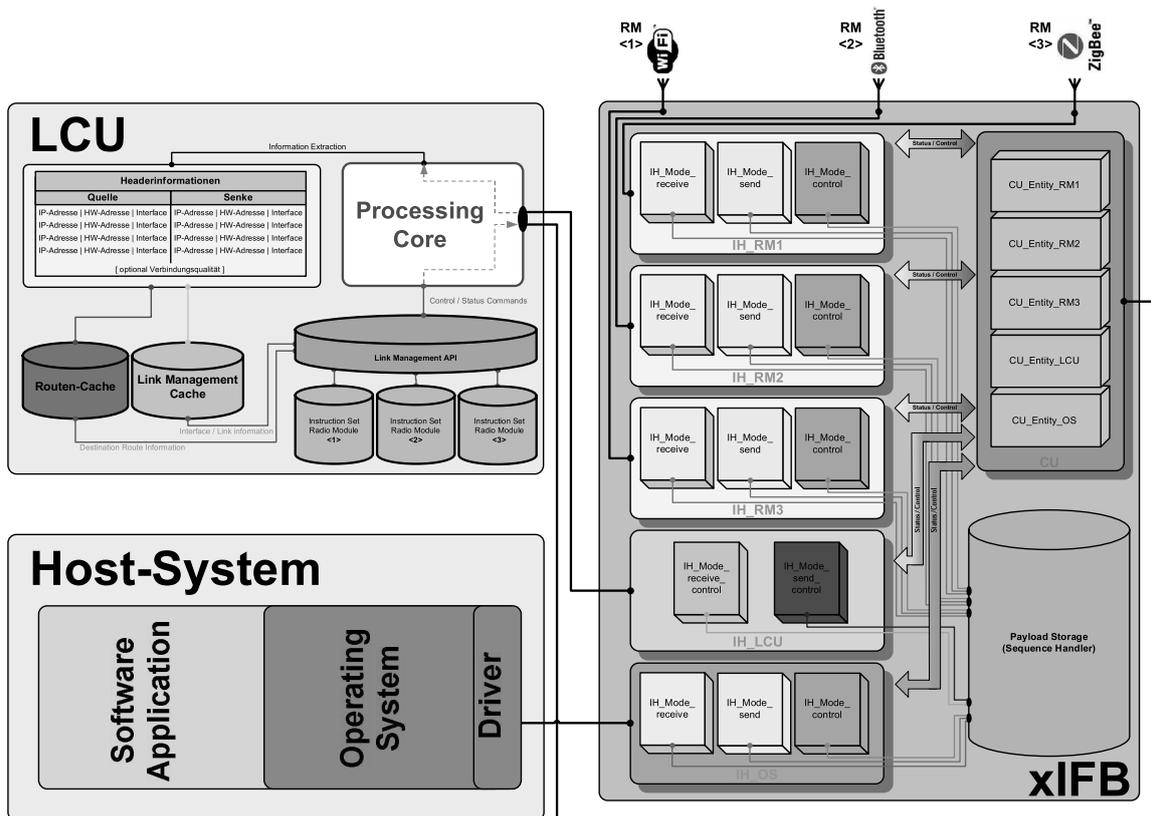


Abbildung 4.20.: xIFB Anpassungen mit der Integration der Protocol Handler und Sequence Handler in die *Interface Handler (IH)*. Die *Link Controller Unit (LCU)* als weitere zentrale Komponente wird über einen separaten IH eingebunden.

4.4.3. Link Controller Unit (LCU)

Im Rahmen des EAN Konzeptes kommt für das Verbindungsmanagement eine gekapselte Komponente zum Einsatz - die *Link Controller Unit (LCU)*. Diese verwaltet auf der unteren Protokollebene Punkt-zu-Punkt Datenkanäle in der direkten Nachbarschaft (*Single-Hop*). Gleichzeitig prüft die LCU auf Netzwerkebene (*Multi-Hop*) Routenpfade zur den jeweiligen Zielknoten hinsichtlich ihrer Erreichbarkeit.

Abbildung 4.21 zeigt die Struktur des Moduls. Auf die einzelnen Komponenten wird nun näher eingegangen.

4. Embedded Ambient Networking

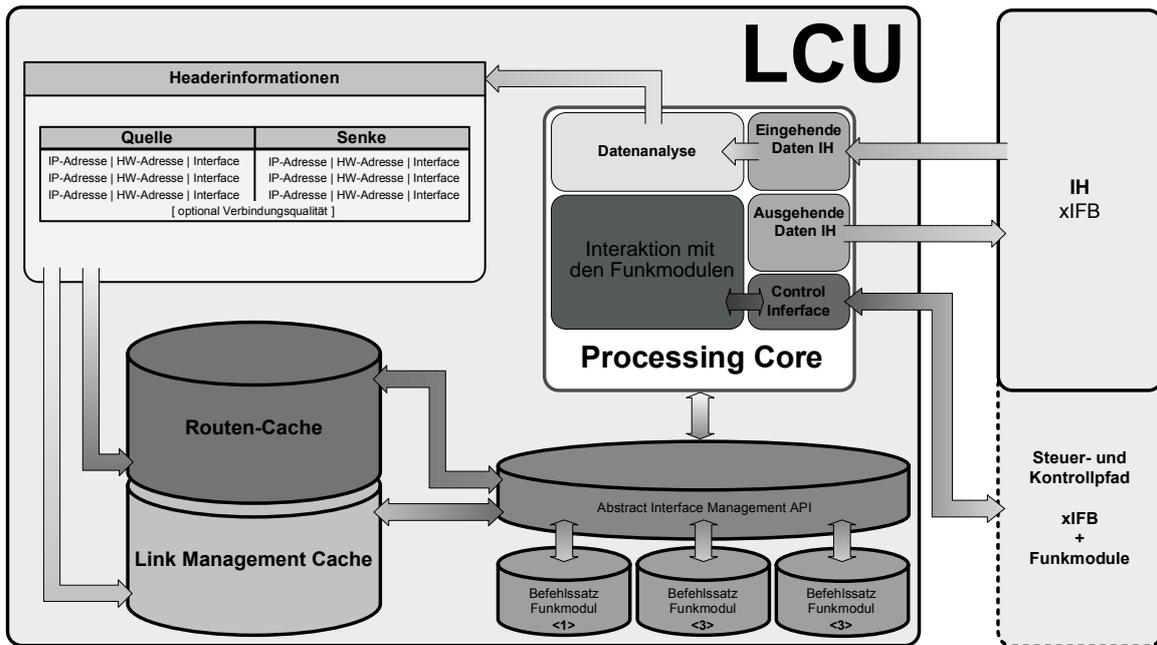


Abbildung 4.21.: Schematische Darstellung der LCU Einheit und deren Anbindung an den EAN Kern. Die einzelnen Komponenten und deren Interaktion sind deutlich erkennbar.

4.4.3.1. Processing Core

Der Processing Core repräsentiert die zentrale Kontrollinstanz innerhalb der LCU. Er koordiniert die Analyse und Verarbeitung aller Adressinformationen. Er stellt gleichzeitig die Schnittstellen zum EAN Gesamtsystem zur Verfügung. Weiterhin greift der Processing Core auf hinterlegte, funktionspezifische Befehlssätze zu und generiert bei Bedarf hardware-spezifische Steuerkommandos für die Funkmodule.

4.4.3.2. Control Interface und Interface Handler

Für die Anbindung an den xIFB existieren zwei dedizierte Kommunikationspfade. Über einen separaten IH werden Headerinformationen aus den Datenpaketen vom xIFB empfangen und anschließend analysiert. Diese umfassen primär Routen- und Adressinformationen sowie dazugehörige Netzwerkinterfaces. Optionale Parameter, welche Rückschlüsse auf die Qualität des Verbindungskanals ermöglichen, werden ebenfalls verarbeitet.

Ein separater Steuer- und Kontrollpfad (*Control Interface*) bietet eine direkte Anbindung an die CU sowie an die einzelnen Funkmodule. Die Anbindung ermöglicht dabei eine zeitliche Koordinierung des Paketverarbeitungsprozesses zwischen xIFB und LCU.

4.4.3.3. Routen-Cache

Um die Netzlast in der Topologie während des Betriebes zu reduzieren, nutzt die LCU Caching-Mechanismen, welche es erlauben, aktuelle Routeninformationen in einem definierten zeitlichen Rahmen und in begrenztem Umfang zwischenspeichern. Diese stehen somit ohne Verzögerung und ohne ein erneutes Route-Request zur Verfügung.

Route Cache Container					
Netzadresse	Subnetz	Gateway	Metric	Interface	Age
134.109.1.1	255.255.254.0	134.109.2.254	1	0	4
134.109.18.22	255.255.192.0	134.109.63.254	2	2	6
1.2.3.4	255.240.0.0	134.109.184.254	0	1	1
...

Abbildung 4.22.: Eine mögliche Struktur innerhalb des Routen-Cache Containers.

Der Routen-Cache-Container enthält dabei die extrahierten Routeninformationen für gültige Pfade in der n -Hop-Nachbarschaft ($n > 1$) (Abbildung 4.22). Er ist nach dem *LRU*⁷ Prinzip organisiert. Bei Erreichen des Kapazitätslimits entfernt dieser selbstständig die am längsten nicht mehr genutzten Einträge. Diese verwendete Caching-Strategie ist leicht umzusetzen und erfordert nur minimale Hardwareressourcen. Alternativ ist auch eine *FIFO*⁸-basierte Cacheorganisation denkbar, bei der jeweils strikt die ältesten Einträge verdrängt werden.

4.4.3.4. Link Management Cache (LMC)

Dieser Cache-Container verwaltet Low-Level Verbindungsinformationen für Kommunikationskanäle in der direkten Nachbarschaft. Er beinhaltet Hardware-Adressen sowie notwendige Verbindungsidentifikatoren *Connection Handles*, welche im Zuge einer Verbindungsinitialisierung generiert werden (Abbildung 4.23). Die interne Organisation gleicht dabei der des Routen-Caches.

LMC Container				
Hardwareadresse	Verbindungsidentifikator	Qualität	Interface	Age
00:21:85:AD:2A:6C	<connectionHandle_1>	1	0	3
12:53:AF:B3:26:32	<connectionHandle_2>	1	2	1
00:11:22:33:44:55	<nicht notwendig>	4	1	4
...

Abbildung 4.23.: Eine mögliche Struktur innerhalb des Link Management Cache Containers.

⁷Least Recently Used

⁸First In First Out

4.4.3.5. Abstract Interface Management API (AIM)

Diese Komponente stellt eine API mit abstrakten Funktionen für das Verbindungsmanagement bereit. Die einzelnen Funktionen sind dabei auf eine Menge funktionsmodulspezifischer Low-Level Befehlsätze (*Instruction Sets*) abgebildet und repräsentieren gekapselte Zustandsautomaten. Der Funktionsumfang beinhaltet grundlegende Prozesse für die Netzwerkkommunikation, wie z.B. *discoverNodes()*, *connect()*, *disconnect()* sowie *send()* und *receive()*. Diese Parametrisierung dieser Funktionen hängt von der Hardware als auch vom Funkstandard ab und muss an die Gegebenheiten angepasst werden. Lösungsansätze für die modulare Hardware-Integration und Erweiterung wurden in Abschnitt 4.3.1.5 bereits vorgestellt. Während des Betriebes greift die AIM auf die Wissensbasis des Link Management Caches und des Routen-Caches zu.

4.4.4. Energy Balanced Cooperative Routing (EBCR)

Resultierend aus den konzeptionellen Routing-Anforderungen an das EAN Konzept wurde ein geeignetes Verfahren konzipiert. Dabei entstand ein hybrides Routingprotokoll - das *Energy Balancing Cooperative Routing (EBCR)* Verfahren [5, 7]. Es kombiniert Routingstrategien aus dem Bereich mobiler Ad Hoc Netzwerke mit zusätzlichen, kooperativen Aspekten drahtloser Sensornetzwerke.

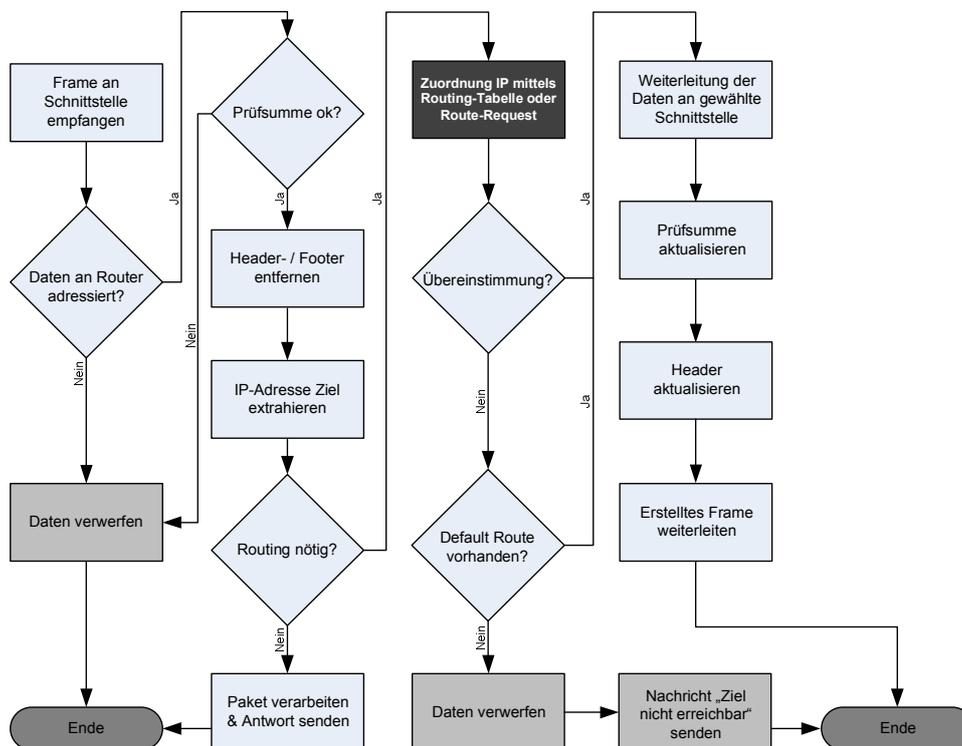


Abbildung 4.24.: Der allgemeine Ablauf eines IP-basierten Routingprozesses, dargestellt als UML Zustandsdiagramm.

Zum besseren Verständnis verdeutlicht Abbildung 4.24 schematisch einen allgemeinen, proze-

duralen Routingablauf. Der Zeitpunkt für eine evtl. notwendige Pfadsuche ist dunkel markiert. Dieser Ablauf unterteilt sich in den Empfang der Daten, die anschließende Prüfung und Verarbeitung der darin enthaltenen Informationen sowie die Bestimmung der zu nutzenden Netzwerkinterfaces und das Versenden der angepassten Datenpakete.

EBCR repräsentiert einen Cross-Layer-Routing Ansatz, welcher zusätzlich lokale Systeminformationen der Netzwerkknoten in seine Wissensbasis integriert [89, 193, 195]. In Folge dessen stützen sich die Routenberechnungen auf Informationen unterschiedlicher Protokollebenen, wodurch vielfältige Möglichkeiten für Optimierungen entstehen.

Innerhalb des EBCR Verfahrens werden diese zusätzlichen Informationen der Systemhardware genutzt, um differierende Energieniveaus zwischen einzelnen Netzwerkknoten auszugleichen. Das Routingverfahren prüft dazu den Ladezustand der jeweiligen Knoten. Diese Informationen fließen in die Gewichtung und Priorisierung der gefundenen Routenpfade ein. Durch die daraus resultierende Lastverteilung kann eine Optimierung der Lebensdauer aller Knoten erfolgen, was die Erreichbarkeit der gesamten Topologie steigert. Knoten mit kritischem Ladezustand werden geschont, Pfade mit hinreichend Energiereserven entsprechend favorisiert.

EBCR arbeitet auf Basis einer reaktiven Pfadsuche. Im Gegensatz zu proaktiven Verfahren existiert keine permanent gepflegte Routingtabelle, welche zu jedem Zeitpunkt einen gültigen Pfad für einen Zielknoten bereitstellt. Im MANET Umfeld würde dies in Folge der hohen Knotendynamik zu einem erheblichen Protokolloverhead führen. Folglich stoßen proaktive Verfahren an dieser Stelle schnell an ihre Leistungsgrenzen und sind nicht mehr effizient einsetzbar.

Reaktiven Routingverfahren starten im Zuge der Verbindungsinitialisierung zunächst ein *Route-Request*. Somit erfolgt erst bei Bedarf eine Pfadsuche und -selektion. Diese Suche verursacht Verzögerungen zu Beginn einer jeden Kommunikation.

Im Umfeld von MANETs erscheint aufgrund dessen der Einsatz eines hybriden Routingverfahrens sinnvoll. EBCR arbeitet reaktiv und integriert zusätzliche Caching-Mechanismen um die Latenzen zu verringern. Dabei speichert das Verfahren gefundene, gültige Routen für ein vordefiniertes Zeitfenster und kann bei Bedarf auf diese zurückgreifen ohne erneut eine aufwendige Pfadsuche zu starten. Parameter, wie bspw. die maximale Anzahl oder die Gültigkeitsdauer der Einträge im Routingcaches, müssen dabei an die Dimensionen sowie die Dynamik der Topologie angepasst werden.

4.4.4.1. Definition Kostenvektor

Zentraler Bestandteil des EBCR-Verfahrens ist ein dynamischer Kostenvektor zur Berechnung der Route, welcher funkstandard- und hardwarespezifische Parameter aus den lokal verfügbaren Netzwerkinformationen sowie aus dem Systemstatus integriert. Die abstrakten Kostenwerte beinhalten dabei neben Grundkosten für den Betrieb der Funkmodule und mehreren funkstandardspezifischen Parametern auch Kosten für die Konvertierung zwischen unterschiedlichen Kommunikationsstandards.

4. Embedded Ambient Networking

- Anzahl Hops in Form von Grundkosten für die Behandlung von Datenpaketen (*Basic Costs per Hop* C_{base})
- Eigenschaften des genutzten Funkstandards
 - Komplexität der Verbindungsinitialisierung (*Channel Initialisation* C_{init})
 - Energiebedarf in Form der benötigten Feldstärke (Sendeleistung für das Erreichen des Kommunikationspartners *Transmitting Power Consumption* C_{tpow})
 - Datenrate / Übertragungszeit (*Overall Data Transmission Time* C_{time})
 - Latenzzeiten (*Data Packet Latency* C_{lat})
- Kosten für den Wechsel des Funkstandards (*Swap radio standard* C_{swap})
- verfügbare Energieressourcen (*gerätespezifisch - Current Energy Level* $C_{e.level}$)

Die verfügbaren Energieressourcen repräsentieren einen systemhardware-bezogenen Parameter, welcher für das bereits erwähnte Ausbalancieren der topologieweiten Energieniveaus durch das EBCR Verfahren eine zentrale Rolle einnimmt und Routenentscheidungen wesentlich beeinflusst. Die jeweiligen Parameter fließen in eine Gesamtfunktion ein, welche die Kosten für einen gefundenen und gültigen Routenpfad abstrahieren. Einzelnen Faktoren müssen dabei genauer definiert werden:

$$C_{e.level} = \frac{1}{P_{current}} \text{ mit } 0.1 \leq P_{current} \leq 1.0$$

$$C_{tpow} = 1 + (C_{tpow_{max_{RS_x}}} \cdot P_{transmit}^2) \\ \text{mit } 0.1 \leq P_{transmit} \leq 1.0$$

$$C_{time} = C_{time_{max_{RS_x}}} \cdot P_{transmit} \\ \text{mit } 0.1 \leq P_{transmit} \leq 1.0$$

$P_{current}$... aktueller Ladezustand

$P_{transmit}$... zur Übertragung nötige Sendeleistung

Basierend auf den Parametern folgt die Herleitung der Gesamtkosten. Dabei wird jeder Kostenparameter mit dem jeweils definierten Gewichtungsszenario multipliziert. Die Summe aller Faktoren ergibt den abstrakten Kostenwert für die Datenübertragung in einem Netzwerkknoten. Zur Bestimmung der Gesamtkosten eines Routenpfades werden diese Kosten über alle Hops

der Route aufsummiert:

$$\begin{aligned}
 C_{overall} &= \sum_{i=1}^{\#Hops} \left(\sum_{j=1}^n (w_j \cdot C_j) \right) \\
 &= \sum_{i=1}^{\#Hops} (w_{base} \cdot C_{base} + w_{e_level} \cdot C_{e_level} \\
 &\quad + w_{init} \cdot C_{init} + w_{tpow} \cdot C_{tpow} + w_{time} \cdot C_{time} \\
 &\quad + w_{lat} \cdot C_{lat} + w_{swap} \cdot C_{swap})
 \end{aligned}$$

n ... Anzahl Parameter

w_x ... Gewichtung des Parameters C_x

Je nach Anwendungsszenario ermöglicht ein zusätzlicher Gewichtungsfaktor w_x eine verfeinerte Priorisierung der einzelnen Parameter C_x , bspw. Netzwerklatenz oder Datendurchsatz, was die Selektion der optimalen Route entsprechend beeinflusst. Die Kostenfunktion integriert weiterhin Parameter für den nötigen Aufwand eines initialen, funkstandardspezifischen Kanalaufbaus, der sich von Technologie zu Technologie teilweise stark unterscheidet. Ein typisches Beispiel hierbei ist der in Kapitel 2.2.1 vorgestellte, komplexe Kommunikationsprozess zweier Bluetooth-Endgeräte mit der dazugehörigen Aushandlung eines Dienstprofils, welcher erhebliche zeitliche Ressourcen bindet. Im Gegensatz dazu ist die Kommunikation via ZigBee deutlich einfacher. Diese Unterschiede müssen in der Modellierung entsprechend beachtet werden.

4.4.4.2. Prozeduraler Ablauf des EBCR Verfahrens

Grundlage für das EBCR Verfahren ist eine rekursiv ablaufende Tiefensuche, welche vom routensuchenden Startknoten in Form eines *Route-Requests* initiiert wird. Im Zuge der Routenberechnung der Quelle *src* zur Senke *dst* sind die Abbruchbedingungen der Rekursion in einem Knoten n , welcher eine Routenanfrage empfängt, wie folgt definiert:

- $n = dst$, d.h. der Zielknoten ist erreicht.
- der Knoten n wurde im bereits zurückgelegten Suchpfad schon einmal gefunden (Kreisfreiheit sicherstellen).
- auf Basis vorangegangener Suchanfragen existiert im Routencache des Knoten n eine gültige Route zum Zielknoten *dst*.
- auf Basis vorangegangener Suchanfragen existiert im Routencache bereits einen gleich langer oder kürzerer Routenpfad von der Quelle *src* zum aktuellen Knoten n .

Im Falle des Eintreffens einer oder mehrerer dieser Bedingungen wird die Routenanfrage nicht weiter bearbeitet und verworfen. Anderenfalls startet ein neuer Rekursionsschritt durch das Weiterleiten der Routenanfrage von Knoten n an seine unmittelbare Nachbarschaft, wobei die

4. Embedded Ambient Networking

Kostenwerte des mitgeführten Kostenvektors entsprechend aktualisiert werden. Aufgrund der genannten Abbruchbedingungen erreicht das EBCR Verfahren eine lineare Laufzeit, was besonders in Hinblick auf große Ad Hoc Topologien von entscheidender Relevanz ist. Je nach Metrik wird aus den gültigen Antworten der Routinganfragen die jeweils beste Route (sofern vorhanden) selektiert. Abbildung 4.25 verdeutlicht den Ablauf des EBCR Verfahrens in einer Pseudocode-Darstellung.

Im Falle gültiger Einträge in den Cache-Containern existiert bereits ein geprüfter Pfad zum Zielknoten und der gesamte Wegfindungsprozess kann übersprungen werden. Alternativ können auch mehrere Alternativrouten gespeichert und genutzt werden, wodurch prinzipiell auch *Multipath*-Routingfunktionalitäten realisierbar sind (siehe Kapitel 3.1.2).

```
n : FIND-ROUTE(src, dst, isCostOptimal, costs, path)

    ▷ Rekursionsabbruch
1  if n = dst or n ∈ path then
2      return path
3  if bessere Route im Routencache then
4      return FALSE

    ▷ Eintrag im Routencache hinzufügen
5  routeCache.add(src, costs)

    ▷ Rekursion
6  list routes ← new list
7  for each neighbour m do
8      if isCostOptimal then
        ▷ kostenoptimal
9          newCosts ← Grundkosten pro Hop
10         newCosts ← newCosts + Kosten bei Funkstandardwechsel
11         routes.add(m.Find-Route(src, dst, TRUE, costs + newCosts, path ∪ {n})))
        else
        ▷ pfadoptimal
12         routes.add(m.Find-Route(src, dst, FALSE, costs + 1, path ∪ {n})))

    ▷ Liste auswerten
13 if routes ≠ ∅ then
14     return routes.getShortestRoute()
        else
15     return FALSE
```

Abbildung 4.25.: Das *Energy-Balanced Cooperative Routing (EBCR)* Konzept als Pseudocode dargestellt.

In Hinblick auf eine effiziente Pfadsuche stellen die Rahmenbedingungen mobiler Ad Hoc Netzwerke zentrale Problemstellungen für die Routingverfahren dar. Die Positionen der einzelnen

Knoten sind nur in seltenen Fällen bekannt oder deterministisch ermittelbar. Daher können bekannte und leistungsfähige Wegfindungsalgorithmen, wie bspw. *Dijkstra* oder *A** als heuristisches Verfahren, nicht zum Einsatz kommen. Die Knoten sind, bedingt durch die Dynamik, gezwungen, alle in der direkten Nachbarschaft verfügbaren Kommunikationspartner entweder periodisch auf Ihre Erreichbarkeit abzufragen (proaktiv) oder die Verbindungen jeweils bei Bedarf neu zu prüfen (reaktiv).

4.4.4.3. Beeinflussung der Gewichtungsfaktoren durch Anforderungen an den Kommunikationskanal

Die Gewichtungsfaktoren der einzelnen Kostenparameter können zur Laufzeit durch anwendungsspezifische Anforderungen an den Kommunikationskanal beeinflusst werden. Möglichkeiten um diese Metadaten von der Softwareebene in den EAN Kern zu übermitteln wurden bereits in Abschnitt 4.3.1.3 diskutiert. EBCR verarbeitet die in den IP Headern enthaltenen Metadaten um so die Pfadsuche entsprechend zu optimieren. Dies erfolgt durch eine dynamische Anpassung der einzelnen Gewichtungsfaktoren nach einem definierbaren Regelsatz.

Konzeptbedingt können an dieser Stelle auch Probleme auftreten. EBCR basiert auf einem reaktiven Routingansatz, welcher bei Bedarf eine optimale Route berechnet. Erfolgt während der Kommunikation eine Änderung der nutzerseitigen Anforderungen, treten unter Umständen Verzögerungen in Folge einer erneuten Pfadsuche auf. Diese Verzögerungen sollten eliminiert oder zumindest minimiert werden. Einerseits besteht eine einfache Möglichkeit in der Pflege mehrerer Routenpfade, welche jeweils unterschiedlichen Anforderungen genügen. Bei einem initialen Route-Request müssen dabei aber mehr Ressourcen zur Verfügung gestellt werden. Ein weiterer Lösungsansatz nutzt je nach Priorität der Daten unterschiedliche Routingverfahren. Zeitkritische Nutzdaten werden dabei nicht auf Basis des EBCR Verfahren übertragen. Anstatt dessen könnte ein Flooding-Mechanismus genutzt werden, um die Daten schnellstmöglich an den Zielknoten zu übermitteln. Ein erhöhter Energiebedarf wird hierbei bewusst in Kauf genommen. Derartige Erweiterungen stehen jedoch nicht im Fokus des vorgestellten Verfahrens und repräsentieren lediglich Ansatzpunkte für weiterführende, hybride Lösungsansätze.

4.5. Ablauf des Verarbeitungsprozesses

Nachdem der Aufbau des Systems verdeutlicht und die einzelnen Komponenten vorgestellt wurden, steht nun der prozedurale Ablauf für die Paketverarbeitung im Vordergrund.

Der in Abbildung 4.26 dargestellte Multistandard-Knoten zeigt dabei nochmals die Beziehung der bereits erläuterten Komponenten. Zusätzlich wird die Aufteilung in standardisierte Hardware, EAN Middleware sowie Host-System Anbindung verdeutlicht.

4. Embedded Ambient Networking

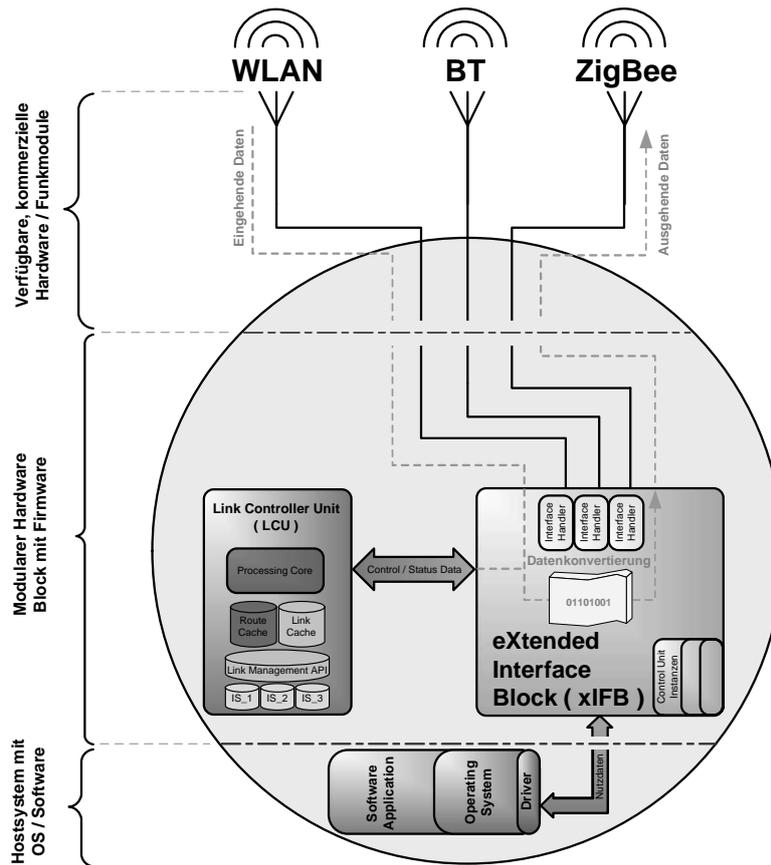


Abbildung 4.26.: Schematische Darstellung eines EAN Netzwerkknotens. Dabei ist die Umsetzung auf 3 Ebenen unterteilt. Gekapselte Funkmodule, welche über eine programmierbare Middleware koordiniert werden. Das Hostsystem mit Betriebssystem und Software ist über eine dedizierte Schnittstelle angebunden.

4.5.1. Protokollkonvertierungsprozess

Die Verarbeitung eines Datenpaketes im xIFB erfolgt in mehreren Stufen unterschiedlicher Komplexität. Abbildung 4.27 verdeutlicht den Gesamtprozess. Die statische Kernfunktionalität des ursprünglichen IFB Ansatz bleibt erhalten. Im Rahmen des vorgestellten EAN Konzeptes kommen Funktionsblöcke für die Bearbeitung dynamischer Paketelemente sowie für das Verbindungsmanagement hinzu.

Die Verarbeitungsprozess beginnt mit dem *protokollkonformen Empfang eines Datenpaketes*. Für die korrekte Behandlung von Datenpaketen mit variabler Länge wird dazu zunächst der Protokoll-Header analysiert. Ist dann die Paketlänge bekannt, kann das Paketende korrekt ermittelt werden. Im Anschluss werden die Adressinformation von Quell- und Zielhost aufbereitet und die Nutzdaten extrahiert.

Sind die Daten nicht für das eigene Host-System bestimmt, kann das Verbindungsmanagement nun aus der Menge möglicher Funkmodule je nach Metrik ein geeignetes Ausgabeinterface definieren. Bevor die Daten nun über das gewählte Interface übertragen werden können, muss das System sicherstellen, dass ein gültiger Verbindungskanal zur Gegenstation verfügbar ist. Falls

dies nicht der Fall ist, prüft das Verbindungsmanagement mittels hardware-spezifischer Steuerkommandos den Verbindungsstatus des Funkmoduls und baut gegebenenfalls einen neuen Kommunikationskanal auf. Schlägt dieser Vorgang fehl, muss der Prozessablauf unterbrochen und nach alternativen Verbindungen gesucht werden.

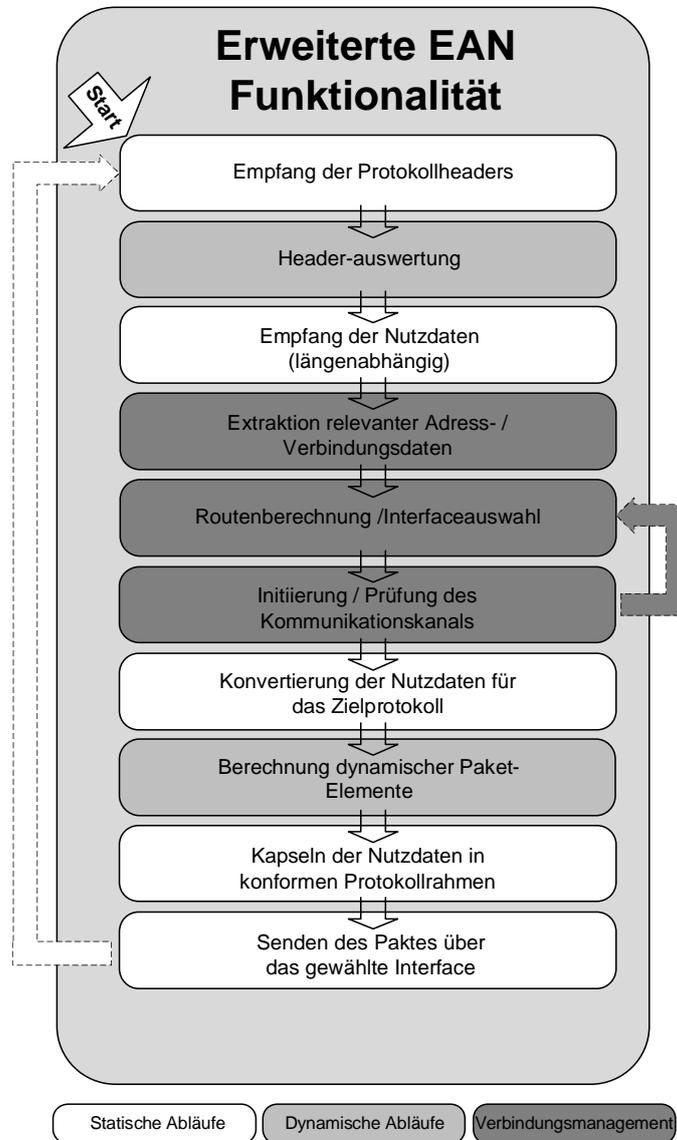


Abbildung 4.27.: Erweitertes IFB Konzept (xIFB). Neben der Integration von Sequence- und Protocol Handler in das übergeordnete *Interface Handler (IH) Konstrukt* umfasst der Ansatz im Wesentlichen die Funktionalitäten der Link Controller Unit, auf die im weiteren Verlauf dieses Kapitels noch detailliert eingegangen wird.

Verläuft die Verbindungsprüfung erfolgreich, beginnt die protokollkonforme Übertragung der Daten. Dazu werden die Nutzdaten für das Zielprotokoll angepasst. Es folgt die Berechnung aller dynamischer Header- und Footerelemente und die Kapselung der Nutzdaten in den Protokollrahmen.

Abbildung 4.28 verdeutlicht die einzelnen Anpassungsprozesse im Zuge der Protokollkonvertierung. Diese werden modular für die jeweils integrierten Funkstandards umgesetzt und auf

4. Embedded Ambient Networking

endliche Zustandsautomaten abgebildet.

Im Zuge der Header- und Footer-Konvertierung (Übergang von Funkstandard 1 zu Funkstandard 2) erfolgen je nach Kommunikationsprotokoll Änderungen in der Reihenfolge einzelner Elemente, inhaltliche Anpassungen sowie die Generierung neuer Elemente bzw. das Entfernen nicht benötigter Informationen. Eventuelle Anpassungen der Nutzdaten beschränken sich entweder auf Änderungen der Byte-Reihenfolge oder auf das Teilen zu langer Datenpakete in protokollkonforme Segmente.

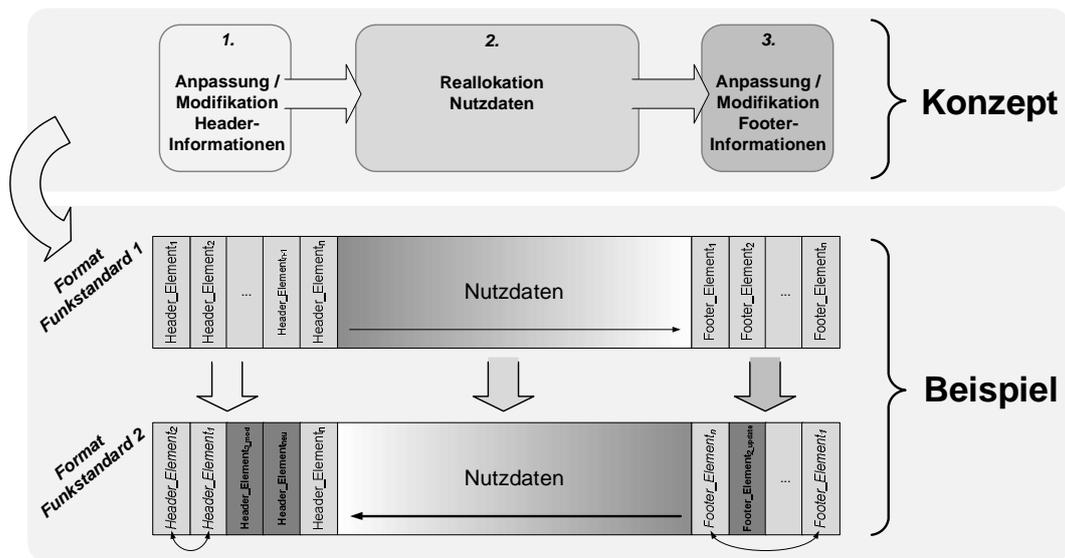


Abbildung 4.28.: Anpassungsprozesse während der Protokollkonvertierung geteilt in drei Abschnitte für Paket-Header, Nutzdaten (Payload) und Footer.

Die Änderungen erfolgen dabei sowohl statisch nach einem vorgegebenen Konvertierungsschema oder auch dynamisch. In diesem Fall müssen einzelne oder mehrere Elemente der Datenpakete in Abhängigkeit aller Rahmenbedingungen berechnet und angepasst werden. Die Erweiterung wurde bereits in Abschnitt 4.4.1 erläutert und führte im Rahmen der softwareseitigen Umsetzung auf einem Mikrocontroller zur Einführung der *Interface Handler* als umfassende Verarbeitungseinheiten.

Nachdem die Nutzdaten sowie die Paketrassen entsprechend dem ausgewählten Zielprotokoll adaptiert wurden, stehen diese im Anschluss für den Versand bereit.

4.5.2. Verbindungsmanagement innerhalb der LCU

Im Folgenden sollen nun die prozeduralen Abläufe innerhalb der LCU näher betrachtet werden. Abbildung 4.29 stellt diesen schematisch dar und zeigt die einzelnen Verarbeitungsschritte. Die Koordinierung des gesamten Bearbeitungsprozesses erfolgt durch den Processing Core der LCU, der zu Gunsten der Übersichtlichkeit in der Abbildung nicht mit dargestellt wurde.

Ausgangspunkt eines jeden Bearbeitungsprozesses ist ein eingehendes Datenpaket über die am

EAN System gekoppelten Funkmodule. Es erfolgt die Trennung von Nutzdaten sowie Adressinformationen und anderen Elementen des Protokollheaders. Anschließend wird der Interface Handler der LCU durch die CU angestoßen und die extrahierten Adress- und Verbindungsinformationen übergeben.

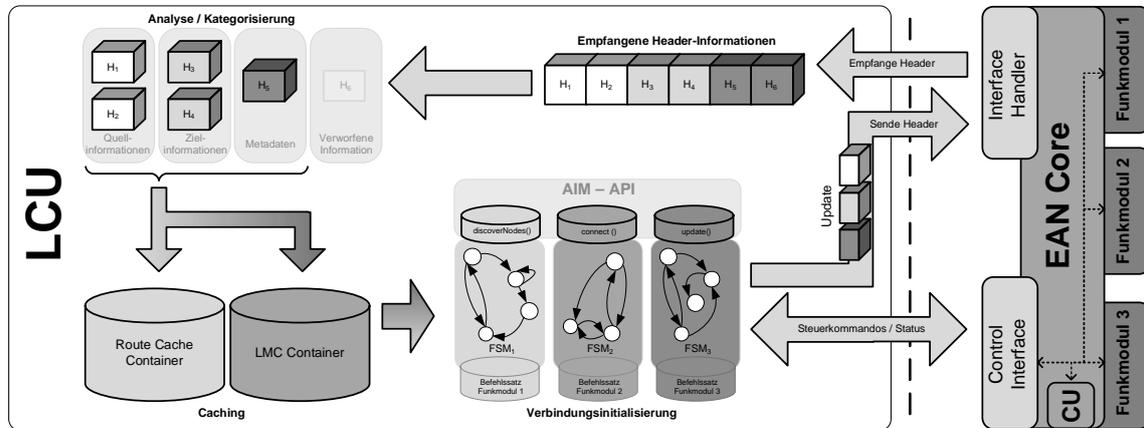


Abbildung 4.29.: Interner Verarbeitungsprozess der LCU und die Anbindung an den EAN Kern über zwei dedizierte Schnittstellen. Über ein *Control Interface* kann direkt mit den einzelnen Funkmodulen kommuniziert werden. Die bidirektionale Anbindung erlaubt die Übermittlung von Steuerkommandos und die Auswertung von Statusinformationen. Gleichzeitig wird über diesen Pfad der Paketverarbeitungsprozess mit der Control Unit koordiniert. Über den *Interface Handler* erfolgt die Kommunikation von Adressinformationen sowie Metadaten.

Daraufhin übernimmt der Processing Core der LCU die weitere Verarbeitung und separiert die übergebenen Elemente nach Quell-, Ziel- und Zusatzinformationen der jeweiligen Verbindung. Nach der Analyse werden diese Informationen in die Datenbasis von Routen-Cache und LMC eingepflegt. Dabei erfolgt im LMC eine Zuordnung von physischen Hardwareadressen auf funktionspezifische, eindeutige Identifikatoren, welche unter Umständen für eine erneute Nutzung der Verbindung benötigt werden. Der Routencache ordnet Quell- und Ziel-Informationen den jeweiligen physischen Funkmodulen zu. Die maximale Anzahl gepflegter Einträge ist limitiert und durch die Cachegröße vordefiniert.

Auf Basis der Zielinformationen und den Cache-Containern kann der Processing Core nun Entscheidungen hinsichtlich der Datenweiterleitung treffen. Liegen gültige Einträge für den Zielhost vor, erhält die CU des xIFBs ein Signal über das Control Interface. Die CU wieder die Koordinierung des weiteren Protokollkonvertierungsprozess. Im Anschluss werden alle für den Paketaufbau benötigten Verbindungsdaten über den Interface Handler an den xIFB übergeben und die protokollkonforme Übertragung des Paketes eingeleitet.

Steht in den Cache-Containern der LCU kein gültiger Übertragungskanal bereit, muss zunächst eine neue Verbindung initialisiert werden. Dazu nutzt der Processing Core die Funktionen der AIM-API. Die Abarbeitung der Verbindungsinitalisierung erfolgt direkt mit den einzelnen Funkmodulen über das Control Interface. Ausgehandelte Connection Handles sind im Anschluss den Cache-Einträgen zuzuordnen. Im Zuge der Verbindungsinitalisierung kommt auch das gewählte Routingverfahren zum Einsatz.

4.5.3. EBCR-basierter Routingprozess

Der Ablauf des im Rahmen dieser Arbeit vorgestellten EBCR Verfahrens soll anhand des nachfolgenden Beispielszenarios 4.30 veranschaulicht werden.

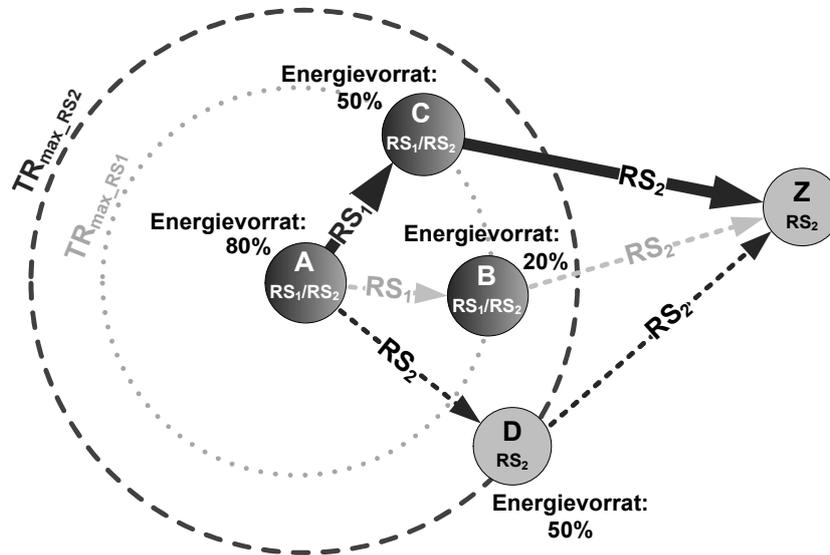


Abbildung 4.30.: Ein einfaches Beispielszenario mit drei möglichen Pfaden von der Quelle A zur Datensinke Z.

Es soll dabei ein optimaler Kommunikationspfad zwischen Knoten A und Z berechnet werden. Die initiale Pfadsuche liefert 3 mögliche Routen über Knoten B, C und D.

Die Knoten A, B und C verfügen dabei über zwei integrierte Funkmodule unterschiedlicher Standards (RS_1 und RS_2) und Sendereichweiten ($TR_{max_RS_1}$ und $TR_{max_RS_2}$). D und Z verfügen ausschließlich über ein Kommunikationsinterface (RS_2). EBCR berechnet nun auf Basis der eingeführten Kostenfunktion eine optimale Route. Dabei wird der aktuelle Ladezustand aller besuchter Knoten berücksichtigt. Die beiden Funkstandards RS_1 und RS_2 sowie eine stark energieoptimierte Metrik werden für das Bsp. wie folgt definiert:

PARAMETER	RS_1	RS_2	w_x	W_{Bsp}
Basiskosten pro Hop C_{base}	5	5	w_{base}	1.0
aktueller Ladezustand C_{e_level}	10-100	10-100	w_{e_level}	1.0
Verbindungsinitialisierung C_{init}	1	1	w_{init}	1.0
Energie für die Datenübertragung C_{tpow}	1-10	3-30	w_{tpow}	0.8
Übertragungsdauer C_{ttime}	2-20	0.5-10	w_{ttime}	0.3
Übertragungslatenzen C_{lat}	20	10	w_{lat}	0.3
Funkstandardwechsel C_{swap}	2	2	w_{swap}	1.0

Tabelle 4.1.: Definition der beiden Funkstandards RS_1 und RS_2 mit der Aufschlüsselung aller abstrakter Kostenparameter. Rechts das gewählte Gewichtungsszenario für die Kostenfunktion C .

Die Routenkosten berechnen sich dabei auf Basis der Kostenfunktion:

$$\begin{aligned}
C_{overall} &= \sum_{i=1}^{\#Hops} \left(\sum_{j=1}^n (w_j \cdot C_j) \right) \\
&= \sum_{i=1}^2 (w_{base} \cdot C_{base} + w_{e.level} \cdot C_{e.level} \\
&\quad + w_{init} \cdot C_{init} + w_{tpow} \cdot C_{tpow} + w_{ttime} \cdot C_{ttime} \\
&\quad + w_{lat} \cdot C_{lat} + w_{swap} \cdot C_{swap})
\end{aligned}$$

Daraus ergeben sich die folgenden abstrakten Routenkosten:

$$\begin{aligned}
C_{overall_A-B-Z} &= \left(1.0 \cdot (2 \cdot 5) + 1.0 \cdot \frac{10}{0.8} + 1.0 \cdot 1 + 0.8 \cdot 10 + 0.3 \cdot 20 + 0.3 \cdot 20 + 1.0 \cdot 0 \right) + \\
&\quad \left(1.0 \cdot (2 \cdot 5) + 1.0 \cdot \frac{10}{0.2} + 1.0 \cdot 1 + 0.8 \cdot 30 + 0.3 \cdot 10 + 0.3 \cdot 10 + 1.0 \cdot 2 \right) \\
&= \underline{136.5}
\end{aligned}$$

$$\begin{aligned}
C_{overall_A-C-Z} &= \left(1.0 \cdot (2 \cdot 5) + 1.0 \cdot \frac{10}{0.8} + 1.0 \cdot 1 + 0.8 \cdot 10 + 0.3 \cdot 20 + 0.3 \cdot 20 + 1.0 \cdot 0 \right) + \\
&\quad \left(1.0 \cdot (2 \cdot 5) + 1.0 \cdot \frac{10}{0.5} + 1.0 \cdot 1 + 0.8 \cdot 30 + 0.3 \cdot 10 + 0.3 \cdot 10 + 1.0 \cdot 2 \right) \\
&= \underline{106.5}
\end{aligned}$$

$$\begin{aligned}
C_{overall_A-D-Z} &= \left(1.0 \cdot (2 \cdot 5) + 1.0 \cdot \frac{10}{0.8} + 1.0 \cdot 1 + 0.8 \cdot 30 + 0.3 \cdot 10 + 0.3 \cdot 10 + 1.0 \cdot 0 \right) + \\
&\quad \left(1.0 \cdot (1 \cdot 5) + 1.0 \cdot \frac{10}{0.5} + 1.0 \cdot 1 + 0.8 \cdot 30 + 0.3 \cdot 10 + 0.3 \cdot 10 + 1.0 \cdot 0 \right) \\
&= \underline{109.5}
\end{aligned}$$

In Folge dessen wird die Route über Knoten B sehr niedrig priorisiert, da hier der Ladestand nur 25% beträgt. Die Route über D erfordert aufgrund der größeren Distanz die Nutzung des Funkstandards RS_2 mit einer deutlich höheren Sendeleistung. EBCR wählt in diesem Beispiel den Routenpfad über C, da in dem gewählten Gewichtungsszenario die Kosten für die Protokollkonvertierung geringer ins Gewicht fallen als die Datenübertragung über den hochenergetischen Funkstandard RS_2 . Je nach Metrik und Gewichtung der einzelnen Parameter kann diese Entscheidung daher aber auch zu Gunsten der Route A-D-Z getroffen werden.

Die EBCR Kostenfunktion berechnet die Güte eines gefundenen Routenpfades durch das Aufsummieren der Kostenwerte in jedem Hop. Diese Verfahrensweise bietet dabei auch eine Schwachstelle. In einem konstruierten Worst Case Szenario beinhaltet ein Routenpfad viele Knoten mit hervorragenden Eigenschaften, welche minimale Kosten für das Routing verursachen. Ein einzelner Knoten auf diesem Pfad mit schlechten Leistungsparametern oder geringen Energiereserven würde die Routenauswahl unter Umständen nicht beeinflussen. Die Gesamtkosten des Pfades bleiben trotz vereinzelter Schwachstellen weiterhin besser als die von ge-

4. Embedded Ambient Networking

gefundenen Alternativrouten. Die Folge wäre eine Verkürzung der Betriebsbereitschaft einzelner Knoten. Abbildung 4.31 verdeutlicht dies anhand eines Beispiels.

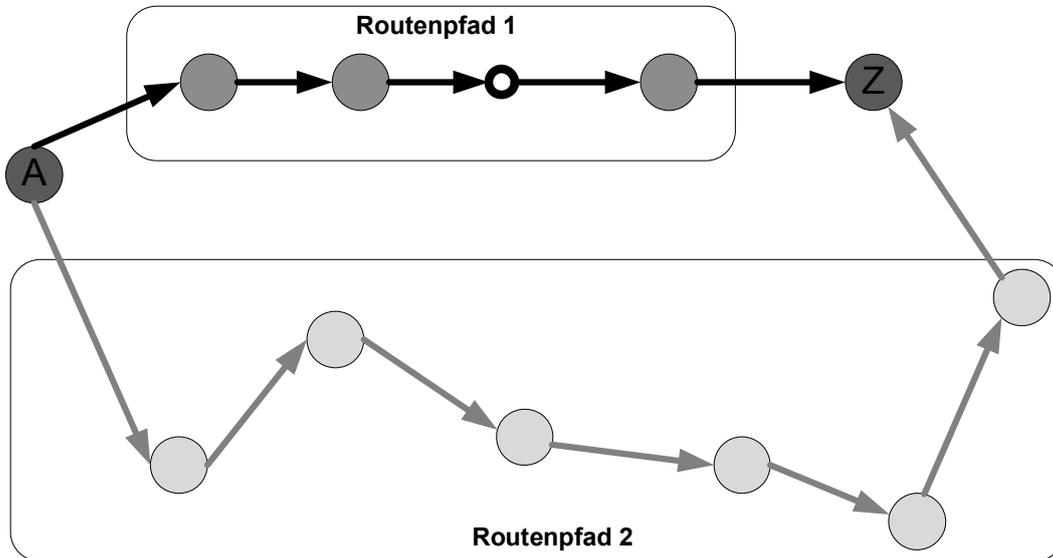


Abbildung 4.31.: Konstruiertes Worst Case Szenario für das EBCR Verfahren. Route 1 verursacht aufgrund der Rahmenbedingungen deutlich weniger Kosten als Route 2. Dabei fällt ein einzelner Knoten mit sehr schlechten Kommunikationsparametern oder auch verbrauchten Energieressourcen kaum ins Gewicht (Knoten im oberen Pfad hervorgehoben). Auch wenn diese Routenwahl aus Sicht der Kommunikationspartner sinnvoll erscheint, kommt hier der kooperative EBCR Ansatz hinsichtlich einer globalen Topologieoptimierung ungenügend zum tragen.

Die Problematik kann auf Basis unterschiedlicher Ansätze vermieden oder ausgeschlossen werden. Der Einfluss einzelner Kostenparameter wird wesentlich durch die Gewichtungsfaktoren definiert. Eine geeignete Parametrisierung ist somit essentiell. Je nach Anwendungsszenario können bestimmte Kostenparameter, wie bspw. das aktuelle Energieniveau eines Knotens, durch einen hohen Gewichtungsfaktor priorisiert werden. Somit wirken sich Schwächen in diesen Leistungsparametern stärker auf die Gesamtkosten einer gefundenen Route aus. Einzelne, benachteiligte Knoten beeinflussen so die Routenwahl effizienter.

Um der Problematik noch direkter entgegenzuwirken, kann die Auswertung der Kosten pro Hop nicht in Summe sondern selektiv erfolgen. Dabei werden Engpässe auf den Routenpfaden einzeln betrachtet und fließen nicht in einen Durchschnittsrechnung ein. Dies bedingt jedoch einen höheren Ressourcenbedarf, da detailliertere Informationen für jede Route zusammengeführt und ausgewertet werden müssen.

Durch eine sinnvolle Definition der Kostenwerte ist das Auftreten solcher Worst Case Szenarien unwahrscheinlich. Im Rahmen dieser Arbeit wurde daher die Kostenfunktion nicht weiter modifiziert.

4.6. Konzeptionelle Grenzen

Das EAN Ansatz ist weder uneingeschränkt anwendbar noch beliebig erweiterbar. Daher sollen an dieser Stelle die konzeptionellen Grenzen aufgeführt werden.

4.6.1. Leistungsfähigkeit EAN-Kern

Die zentralen Komponenten xIFB und LCU stellen in Hinblick auf die Leistungsfähigkeit des Gesamtkonzeptes begrenzende Faktoren dar, da der gesamte Datenfluss über diese Instanzen abgewickelt wird. Folglich entspricht der maximale Datendurchsatz des Gesamtsystems annähernd der Durchsatzrate innerhalb des xIFBs. Übersteigt das Datenaufkommen den Grenzwert, werden eingehende Pakete aufgrund überlaufender Puffer nicht mehr ordnungsgemäß bearbeitet. Folglich kommt es zu Störungen während der Netzwerkkommunikation, die sich in Form einer erhöhten Paketverlustrate [6] manifestieren.

Zusätzlich muss die LCU systemintern die angeschlossenen Funkmodule sowie alle Verbindungskanäle verwalten. Die maximale Anzahl integrierbarer Funkmodule in ein EAN System (RS_{max}) hängt dabei von drei Faktoren ab - funkstandardspezifischen Kommunikationsparametern, funkmodulspezifischen Parametern hinsichtlich der Anbindung an das EAN System sowie die zur Verfügung stehende Datendurchsatzrate R_{xIFB_intern} innerhalb der gegebenen xIFB Komponente:

$$RS_{max} = f(R_{xIFB_intern}, Prop_{RS_x}, Prop_{Modul_{RS_x}})$$

RS_x ... ein gewählter Funkstandard
 $Prop_{RS_x}$... Menge aller funkstandardspezifischen Eigenschaften (Datenrate, Latenz, ...)
 $Prop_{Modul_{RS_x}}$... Menge aller funkmodulspezifischen Eigenschaften (Schnittstelle, API, ...)
 R_{xIFB_intern} ... maximaler Datendurchsatz innerhalb der xIFB Komponente

Die Herleitung der maximalen xIFB Datendurchsatzrate R_{xIFB_intern} soll anhand eines Beispiels verdeutlicht werden: Sei R_{total} die Gesamtübertragungsrate dreier Funkmodule der Standards Bluetooth (R_{BT}), WLAN (R_{WLAN}) und ZigBee (R_{ZiBee}) entsprechend der Knotenrepräsentation aus Abbildung 4.26. Unter Einbeziehung eines zusätzlichen Korrekturparameters α , welcher einen Sicherheitsfaktor für die xIFB Leistungskapazität repräsentiert, errechnet sich ein Mindestwert für R_{xIFB_intern} wie folgt:

4. Embedded Ambient Networking

$$\begin{aligned}R_{BT} &= R_1 = 2.1\text{Mbit/s (Bluetooth 2.0+EDR)} \\R_{WLAN} &= R_2 = 54\text{Mbit/s (IEEE 802.11g)} \\R_{ZigBee} &= R_3 = 0.25\text{Mbit/s (IEEE 802.15.4)} \\R_{total} &= \left(\sum_{i=1}^3 R_i\right) + \alpha \\R_{total} &= 56.35\text{Mbit/s} + \alpha \\R_{xIFB.intern} &\geq 56.35\text{Mbit/s} + 5.635\text{Mbit/s} \\R_{xIFB.intern} &\geq 61.985\text{Mbit/s} \\ \alpha &\dots (10\% \text{ der Gesamtdatentransferrate}) \\ &\quad (\text{Datendurchsatz jeweils brutto und bidirektional})\end{aligned}$$

4.6.2. Komplexität und indirekte Störeinflüsse

In Hinblick auf die Integration unterschiedlicher Kommunikationstechnologien in ein EAN-konformes System steigt die hardwareseitige Komplexität erheblich an. Dies impliziert einen erhöhten Energiebedarf der einzelnen Knoten durch den parallelen Betrieb mehrerer Funkmodule, was sich auf abstrakter Ebene in Form von erhöhten Basiskosten für den Betrieb auswirkt. Um nun trotz der gesteigerten Betriebskosten eine Energieoptimierung zu ermöglichen, muss der Energiebedarf während der Kommunikation gesenkt werden. Dies kann einerseits durch eine intelligente Wahl des zu nutzenden Funkstandards erfolgen, sofern die Rahmenbedingungen dies ermöglichen. Auf der anderen Seite kann Energie durch die Verkürzung von Routenpfaden gespart werden, da in Folge dessen weniger Knoten für die Weiterleitung der Datenpakete einbezogen werden. Zusätzlich ist mit Hilfe geeigneter Cross-Layer Routing-Metriken ein Ausbalancieren unterschiedlicher Energieniveaus einzelner Netzwerkknoten realisierbar, wodurch die Erreichbarkeit der gesamten Topologie verlängert wird.

Um dieses Optimierungspotential zu nutzen, ist jedoch ein entsprechendes Datenvolumen vorauszusetzen. Je höher die topologieweite Netzlast, desto größer ist das Potential für Energieeinsparung durch das EAN Konzept. Im Gegensatz dazu ist bei Kommunikationsszenarien mit geringer Auslastung eine negative Energiebilanz wahrscheinlich.

Neben der funktionalen Paketverarbeitung im Zuge der Protokollkonvertierung ist auch die zeitliche Koordinierung der Kommunikation von Interesse. Das vorgestellte EAN Konzept leitet eingehende Daten nach einer unter Umständen notwendigen Konvertierung schnellstmöglich auf einem gewählten Interface weiter. Beim simultanen Betrieb mehrerer Funkmodule unterschiedlicher Standards beeinflusst das nicht synchronisierte Zeitverhalten das Interferenzniveau in der Topologie. Unabhängig vom genutzten Frequenzband wirkt sich dies auf die Verbindungsqualität aus. Derartige Störeinflüsse sind nur schwer zu modellieren und die evtl. negativen Auswirkungen auf die Kommunikation in der Topologie kaum berechenbar.

Weiterhin hat ein EAN System nur begrenzte Einflussmöglichkeiten auf die gekoppelten Funkmodule. Diese repräsentieren gekapselte Komponenten, deren Ansteuerung über einen begrenzten Befehlssatz auf höherer Protokollebene realisiert wird. Um eine zeitliche Koordinierung aller Module zu gewährleisten, ist ein Low-Level Zugriff auf dem MAC-Layer erforderlich. Dieser ist jedoch nur in Einzelfällen realisierbar. Daher wurden diese indirekten Einflussgrößen nicht mit in die Bewertung der Güte einer funkstandardübergreifenden Kommunikation einbezogen. Weiterführende Untersuchungen sowie Simulationen auf Basis korrekter physikalischer Modelle könnten hier interessante Informationen liefern.

4.6.3. IP Adressierungsmodell und Routing

Weitere konzeptionelle Einschränkungen ergeben sich durch das verwendete Adressierungsmodell auf Basis von IP. Mobile Endgeräte wie Mobiltelefone oder PDAs besitzen im normalen Betriebsmodus keine Routing-Funktionalitäten und werfen jegliche Datenpakete, die nicht dem eigenen Host zuzuordnen sind. Modifikation an den Systemen sind nur schwer realisierbar und wirken außerdem den bereits aufgeführten Anforderungen entgegen. Ohne zusätzlichen Aufwand ist es somit nicht möglich, diese Knoten für die Weiterleitung von Daten zu nutzen und sie agieren daher ausschließlich als passive Endgeräte. An dieser Stelle würde nur eine softwareseitige Erweiterung diese erweiterte Routingfunktionalität bereitstellen. Eine solche Umsetzung auf höheren Protokollebenen benötigt dabei ohne jeden Zweifel erhebliche Ressourcen für die Datenverarbeitung, was das System zusätzlich belastet.

In Hinblick auf die Kompatibilität stellen nicht IP-fähige Kleinstsysteme wie z.B. im Bereich drahtloser Sensornetzwerke einen Grenzfall dar. Diese verfügen über keinen TCP/IP Stack und nutzen meist proprietäre Übertragungsprotokolle, die auf den weniger komplexen, unteren Protokollebenen aufsetzen. Sofern Routing-Funktionalitäten umgesetzt wurden, geschieht dies auf Basis von gerätespezifischen Hardwareadressen bzw. anderen einfachen Identifikatoren. Die Routenfindung erfolgt vorwiegend über primitive Flooding-Mechanismen. Eine hierarchische Adressierungsbasis zur Kapselung in Teilmengen ist nicht vorhanden. Ohne ein solches Adressierungsmodell sind weiterführende Routingansätze nur stark eingeschränkt möglich. Die einzige mögliche Alternative für die Integration solcher Systeme ist die Implementierung spezieller *Low Level Kommunikationsprofile*, welche speziell an die jeweiligen Netzwerkknoten angepasst werden. Die Knoten-Hardware kann somit bei Bedarf direkt angesprochen werden. Ohne das IP-Adressierungsmodell ist jedoch eine direkte Verbindung zu mindestens einem EAN-konformen Netzwerkknoten zwingend notwendig, da ein Multihop-Mischbetrieb von IP und proprietären Verfahren nicht realisierbar ist. Daher wird Einbindung solcher Hardware im Rahmen des hier vorgestellten EAN Konzeptes zunächst vernachlässigt.

Handelsübliche Routing-Komponenten und deren TCP/IP Stacks verfügen über standardisierte Routingverfahren auf Basis einer trivialen Hop-Count Metrik. Der klassische, homogene Systemaufbau realisiert dabei meist nur die Auswahl zwischen verschiedenen drahtlosen oder drahtgebundenen Netzwerkkomponenten eines einheitlichen Übertragungsstandards. Ein dynamischer Wechsel des Kommunikationsstandards bei mehreren parallelen Übertragungskanälen zum gleichen Host, ist nur sehr stark eingeschränkt oder überhaupt nicht möglich. Routingver-

fahren wie RIP sind mit den erweiterten Routingprotokollen des EAN Konzeptes ausschließlich kompatibel. Effiziente Optimierungen des Routenpfades sind auf solchen Hardwarekomponenten nicht realisierbar und beschränken sich auf EAN-konforme Systeme. Durch die Kompatibilität zu standardisierten Routingprotokollen ist jedoch zumindest eine störungsfreie Kommunikation mit handelsüblichen Endgeräten auf Basis von IP sichergestellt. In der heterogenen Hardwarelandschaft des EAN Konzeptes genügt es dabei, eine ordnungsgemäße Verarbeitung der Route-Requests zu gewährleisten und RIP-protokollkonforme Routen-Pakete zu erstellen.

4.6.4. Protokollkonvertierung und Paketlaufzeit

Eine Konvertierung zwischen zwei Kommunikationsstandards, insofern diese nötig ist, wird in Form endlicher Automaten durchgeführt und verursacht Verzögerungen innerhalb des verarbeitenden Knoten. Je nach Komplexität der Protokolle schwanken diese Verarbeitungszeiten. Hinzu kommen Latenzen durch das Verbindungsmanagement, da die LCU sicherstellen muss, dass ein gültiger und freier Kanal für die Datenübertragung zur Verfügung steht. Der nötige Zeitaufwand ist nur schwer zu ermitteln und richtet sich primär nach den initialen *Handshake-Mechanismen* innerhalb der Kommunikationsstandards.

Diese Problematik ist jedoch nicht EAN-spezifisch, da die Überwachung und Koordinierung der Verbindungskanäle auch herkömmlicher, routing-fähiger Hardware von Relevanz ist. Eine Analyse dieser Verbindungsabläufe erfolgte in der Arbeit *Methoden zur modularen Erweiterung funkstandardübergreifender Hardware* [14], deren Ergebnisse in Abschnitt 4.3.1.5 sowie in Kapitel 6.3 einfließen konnten.

Summieren sich mehrere Funkstandardwechsel innerhalb eines Routenpfades, erreichen die Verzögerungszeiten während der Kommunikation unter Umständen einen kritischen Wert, bei dem Pakete verworfen und folglich Verbindungen gestört werden. Die konzeptionellen Vorteile des vorgestellten EAN Ansatzes treten dann in den Hintergrund, da der bereitgestellte Verbindungskanal nicht effektiv genutzt werden kann. Je nach Anforderungen an die Kommunikation wird dieser Grenzbereich unterschiedlich schnell erreicht.

Um der Problematik entgegenzuwirken, ist es von besonderer Wichtigkeit, innerhalb der Routingalgorithmen die Kostenparameter für einen Wechsel des Kommunikationsstandards sinnvoll und realitätsnah zu gewichten. Wie die Simulationen in Kapitel 5.6 verdeutlichen, ergeben sich bei einer hinreichend guten Parametrisierung deutliche Vorteile während der Übertragung. Die abstrakten Kostenwerte für einen Funkstandardwechsel wurden dabei so definiert, dass sie sich deutlich auf die Gesamtkosten einer Multihop-Verbindung auswirken. Ziel ist es, die maximale Anzahl nötiger Konvertierungen je nach Pfadlänge auf drei zu begrenzen. Konzeptionell ist im Rahmen einer praktischen Umsetzung des EAN Konzeptes in jedem Fall mit erhöhten Latenzen zu rechnen. In wie weit diese die Kommunikation negativ beeinflussen, wird in Kapitel 6 untersucht.

4.7. Zusammenfassung

Das in diesem Kapitel vorgestellte Konzept bietet einen effizienten Lösungsansatz für die funktionsstandardübergreifende Kommunikation in mobilen Ad Hoc Netzwerken. Der EAN Ansatz bezieht sich dabei auf die konzeptionelle Grundlagen der Forschungsgebiete *Cognitive Radio* und *Ambient Networking*. Das Ziel ist hierbei die Kopplung unterschiedlicher, standardisierter Funkmodule auf einer hardwarenahen Ebene, wodurch ein dynamischer Wechsel zwischen den verfügbaren Funkstandards realisiert werden kann. Eine Kontrollinstanz steuert die Verarbeitung und Weiterleitung eingehender Datenströme auf Basis funktionaler Anforderungen an den Kommunikationskanal.

Das Kapitel erläutert dabei zunächst die Idee des EAN Ansatzes und zeigt die Einordnung in das ISO/OSI Modell. Im Anschluss wurden die Anforderungen an das Konzept detailliert beschrieben und analysiert.

Es folgten die wesentlichen Kernpunkte des Kapitels, welche sowohl den Aufbau als auch den Ablauf innerhalb eines EAN-konformen Kommunikationssystems betrachteten. Dazu wurden im weiteren Verlauf die zentralen Problemstellungen ausführlich diskutiert und entsprechende Lösungsansätze präsentiert. Hinsichtlich des Protokollkonvertierungsprozesses wurde die xIFB Erweiterung vorgestellt. Die Konzeption der Link Controller Unit repräsentiert eine neue Komponente der EAN Systemstruktur, welche das komplette Verbindungsmanagement kapselt. In Bezug auf die Wegefindung in der neu geschaffenen, heterogenen Netztopologie wurde ein hybrides Cross-Layer Routingverfahren entwickelt, welches kooperative Aspekte aus dem Bereich drahtloser Sensornetze integriert und sich funktionalen Anforderungen an den Kommunikationskanal dynamisch anpassen kann.

Das Gesamtkonzept ermöglicht eine kostengünstige Integration mehrerer Kommunikationstechnologien in eine heterogene Knotenlandschaft. Somit kann die Erreichbarkeit in der Topologie signifikant gesteigert werden. Gleichzeitig wird der topologieweite Energiebedarf durch optimierte Routenpfade reduziert und die Verbindungsqualität verbessert.

Im folgenden Kapitel wird nun die entwickelte Simulationsumgebung vorgestellt. Auf Basis der durchgeführten Testszenarien sollen dabei die konzeptionellen Vorteile eines EAN Systems evaluiert werden.

5. Simulationsumgebung SimANet

Die Vorteile des EAN Konzeptes mit den präsentierten Lösungsansätzen sollen zunächst anhand verschiedener Simulationsszenarien evaluiert werden. Aufgrund von Einschränkungen verfügbarer Simulationswerkzeuge in Hinsicht auf Funktionalität und Leistungsfähigkeit wurde sich für die Neuimplementierung einer geeigneten Simulationsumgebung entschieden. Die entwickelte Software *SimANet* (*Simulation environment for Ambient Networking*) [11–13] bietet dabei eine modular erweiterbare Basis für eine Vielzahl möglicher Anwendungsbereiche.

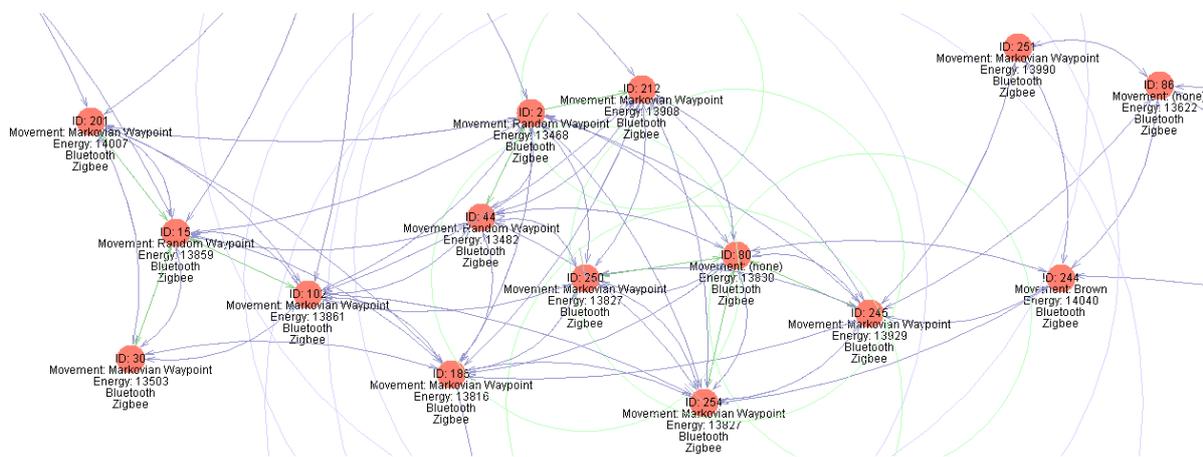


Abbildung 5.1.: SimANet - Simulation environment for Ambient Networking.

SimANet unterstützt den simultanen Betrieb mehrerer Funkmodule innerhalb eines Netzwerkknotens und erlaubt auch die Analyse hoch skalierten Topologien. Auf abstrakter Ebene wird die Definition unterschiedlicher Funkstandards ermöglicht, auf deren Basis neue Kommunikationskonzepte, Routingverfahren oder Topologieoptimierungsansätze getestet werden können.

Im Rahmen der durchgeführten Testszenarien stehen drei zentrale Fragestellungen im Fokus der Betrachtungen. Einerseits sollen die zur Verfügung stehenden Energieressourcen optimal genutzt werden. Auf der anderen Seite wird die mögliche Erhöhung der Erreichbarkeit aller Knoten in der Topologie analysiert. Zusätzlich sollen in der geschaffenen Topologie optimierte Kommunikationspfade genutzt werden. Die Güte des vorgestellten EBCR Verfahrens ist dabei von besonderem Interesse.

Im weiteren Verlauf dieses Kapitels wird SimANet detailliert beschrieben und die gewonnenen Erkenntnisse präsentiert.

5.1. Simulationsumgebungen im Bereich mobiler Ad Hoc Netzwerke

Um die Komplexität und den Entwurf moderner Kommunikationstechnologien in einem beherrschbaren Rahmen zu halten, ist die Simulation auf unterschiedlichen Entwicklungsebenen zwingend erforderlich. Neue Konzepte müssen dabei zunächst auf einer abstrakten Ebene spezifiziert und analysiert werden. Detailwissen und funkstandardspezifische Kommunikationsparameter sind in dieser Phase nur stark eingeschränkt von Interesse. Wurden die funktionalen Aspekte erfolgreich evaluiert, folgen im weiteren Verlauf der Entwicklung zusätzliche Simulationen unter Berücksichtigung konkreter technologischer Rahmenbedingungen. Erst im Anschluss erfolgen prototypische Umsetzungen.

Aussagekräftige Simulationsergebnisse haben somit direkten Einfluss auf die Entwicklungszeiten (*Time-To-Market*) und die Entwicklungskosten neuer Kommunikationssysteme. Leistungsfähige Werkzeuge zur Simulation komplexer Szenarien spielen daher aus wissenschaftlicher und wirtschaftlicher Sicht eine wesentliche Rolle.

In Folge dessen entstand innerhalb der letzten Dekade eine Vielzahl anwendungsspezifischer Simulationswerkzeuge im Bereich drahtgebundener und drahtloser Netzwerkkommunikation. Im Kontext mobiler Endgeräte wie Mobiltelefone, Laptops oder PDAs rücken Simulationsszenarien in heterogenen, dynamischen Netzwerktopologien in den Fokus wissenschaftlicher Betrachtungen.

Klassische Simulationstools wie bspw. *GloMoSim*¹ [27] oder *SSFNet*² [50] arbeiten ausschließlich kommandozeilenbasiert und folglich gestaltet sich die Realisierung anwendungsspezifischer Simulationsszenarien als schwierig und die anschließende Analyse zeitaufwendig. Eine Interaktion mit der Simulationsumgebung während der Laufzeit ist nur stark eingeschränkt möglich. Um diese Nachteile zu umgehen, wurde *GloMoSim* durch mehrere, plattformabhängige Module zur Visualisierung der Simulationsergebnisse erweitert. Auf Basis des Simulationskerns von *GloMoSim* entstand im weiteren Verlauf des Entwicklungsprozesses der *QualNET*-Framework [158], welcher ein großes Spektrum möglicher Anwendungsfelder abdeckt, jedoch rein kommerziell vertrieben wird. Im Gegensatz zu *GloMoSim* fokussiert *SSFNet* statische Kommunikationsszenarios in klassischen, drahtgebundenen Topologien. Einen großen Vorteil der Software stellt die Möglichkeit der Parallelisierung dar. Durch die Verteilung der Rechenlast wurde die Analyse komplexer Simulationen und sehr großer Knotentopologien ermöglicht [152]. Beide Simulationswerkzeuge beschränken sich aber auf die Modellierung einer einzigen Netzwerkschnittstelle pro Netzwerkknoten, wodurch eine Anwendung auf die hier vorgestellten Konzepte nicht realisierbar ist.

Spezielle Simulationswerkzeuge im Bereich der Sensornetze wie bspw. *Shawn* [65] oder *SENS*³ [177] skalieren hingegen auch ohne Parallelisierung gut und simulieren problemlos Topologien

¹Global Mobile Information Systems Simulation Library

²Scalable Simulation Framework Network Models

³A Sensor, Environment and Network Simulator

mit weit über 10.000 Knoten. Die Leistungsfähigkeit resultiert auch hier vorrangig aus der konsolenbasierten Nutzung und damit dem Verzicht auf eine Interaktion während der Laufzeit mit Hilfe einer grafischen Oberfläche. Eine Visualisierung erfolgt bei beiden Simulationswerkzeugen ausschließlich über generierte Screenshots zu definierten Zeitpunkten. Aufgrund der konzeptionellen Einschränkung auf Simulationen drahtloser Sensornetzwerke mit einer gemeinsamen Datenenke kommen die Anwendungen im Rahmen dieser Arbeit nicht zum Einsatz.

Weitere Simulationswerkzeuge wie *NCTUns*⁴ [189] oder der *OMNeT++*-Framework⁵ [186] bieten vielseitige grafische Nutzeroberflächen für die Interaktion mit dem Nutzer. NCTUns ist eine integrierte Netzwerk- und Verkehrssimulation und stellt einen sehr flexiblen Simulationskern zur Verfügung. Die Plattform fokussiert dabei die spezifischen Anforderungen im Bereich der *Inter-Vehicle* und *Car2Car*-Kommunikation. OMNeT++ ist eine weit verbreitete und leistungsfähige Software mit detaillierten Simulationsmodellen auf MAC-Protokollebene für *IEEE 802.11*, *Ethernet*, *FDDI*⁶ und *Token Ring*. Da OMNeT++ ursprünglich für statische Kommunikationsszenarien konzipiert wurde, stehen Simulationen dynamischer Netztopologien mit ausgewählten Bewegungsmodellen erst durch Erweiterungen wie bspw. *Castalia* [24] zur Verfügung. Durch diese Modularität kann OMNeT++ sehr vielseitig genutzt werden. Aber auch diese Simulationswerkzeuge beschränken sich in den möglichen Simulationsszenarien auf die Nutzung eines einheitlichen Funkstandards. Sowohl NCTUns als auch OMNeT++ unterstützen nur eine Kommunikationsschnittstelle pro Knoten.

Grundlegende, konzeptionelle Einschränkungen von OMNeT++ wirken sich ebenfalls auf die Skalierbarkeit aus. So kann die Simulationsumgebung nur sehr eingeschränkt für größere Netztopologien genutzt werden [34]. In Hinsicht auf eine mögliche Parallelisierung bietet OMNeT++ grundlegende Funktionalitäten auf der Basis von *MPI*⁷. Dabei gestaltet sich die Konvertierung bestehender Projekte auf eine parallele Berechnungsumgebung als schwierig und zeitaufwendig. Einige Veröffentlichungen versuchen daher komfortablere Lösungen zur Verfügung zu stellen [162]. Grundlegend basieren jedoch alle Funktionalitäten für parallele Simulationsszenarien in OMNeT++ auf konservativen, blockierenden Synchronisationsmechanismen, was im direkten Widerspruch mit den kommunikationsdominanten Aspekten von Netzwerksimulationen steht.

Andere Konzepte konnten die Einschränkungen durch eine einzelne Kommunikationsschnittstelle umgehen und ermöglichen den Übergang von einer homogenen Knotenlandschaft zu einer heterogenen Netzstruktur. Einer der bekanntesten Frameworks stellt dabei der *Network Simulator - version 2 (ns2)* [86] dar. ns2 ermöglicht die realistische Modellierung von Kommunikationseinheiten über alle Ebenen des jeweils genutzten Protokollstacks. Die Simulationsumgebung kann, ähnlich wie OMNeT++, um zusätzliche Module und Funktionalitäten erweitert werden. Aufgrund der starken Verbreitung von ns2 entstand ein Vielzahl unabhängiger Projekte zur Verbesserung des Funktionsumfangs. Mit dem Fokus auf heterogene, mobile Ad Hoc Netzwerke sind dabei drei wesentlich Erweiterungen zu nennen.

⁴National Chiao Tung University Network Simulator

⁵Open Source Module-based Network Simulator in C++

⁶Fiber Distributed Data Interface

⁷Message Passing Interface

5. Simulationsumgebung SimANet

*TeNS*⁸ [156] ist eine erweiterte Version von ns2 und ermöglicht die Nutzung mehrerer Netzwerkschnittstellen innerhalb eines Knotens. Zwar ist somit die Simulation von *Multi-Interface*-Szenarien möglich, doch auf MAC-Ebene unterstützt TeNS ausschließlich die IEEE 802.11 Spezifikation. In Folge dessen kann auch mit TeNS keine heterogene Netzstruktur mit mehreren, gleichzeitig arbeitenden Kommunikationsstandards realisiert werden.

Mit dem *MW-Node Projekt*⁹ [132] unterstützt ns2 sowohl *Multi-Interface*- als auch *Multi-Standard*-Szenarien und erlaubt die Koexistenz unterschiedlicher Funkstandards in einem Knoten. Die Simulation von Konzepten für die funkstandardübergreifende Kommunikation in MANETs ist dadurch realisierbar. Allerdings wird bei MW-Node jeweils ein Netzwerkinterface auf einen Protokoll-Stack mit den entsprechenden Algorithmen für Medienzugriff, Topologiekontrolle oder Routing abgebildet. Jede Schnittstelle wird folglich als eine unabhängige, dedizierte Kommunikationseinheit betrachtet. Konzepte, wie der im Rahmen dieser Arbeit entwickelte Lösungsansatz auf Basis einer intelligenten *Middleware*, sind daher nicht umsetzbar, da hier die Informationen aller Funkmodule in einer gemeinsamen Wissensbasis verwaltet werden. Auch darauf aufbauende Algorithmen, wie bspw. *EBCR*, können folglich nicht simuliert werden.

Die *ns2-MIRACLE*¹⁰ [29] Erweiterung erlaubt in jeder Protokollebene mehrere koexistente Instanzen und deckt den gesamten Bereich möglicher *Multi-Interface* und *Multi-Standard*-Szenarien ab. Die wesentlichen Nachteile von MIRACLE und allen anderen genannten Erweiterungen von ns2 korrelieren mit konzeptionellen Einschränkungen der Simulationsumgebung. Durch den komplexen Aufbau gestaltet sich die Umsetzung anwendungsspezifischer Szenarien als aufwendig und erfordert vom Nutzer eine relativ hohe Einarbeitungszeit. Für die Evaluierung und Analyse neuer Kommunikationskonzepte auf einer abstrakten und weitestgehend technologieunabhängigen Ebene ist ns2 zu aufwendig und eher ungeeignet. Weiterhin benötigt der ns2 entsprechende Hardwareressourcen in Form von Rechenleistung und Speicher.

Aufgrund der hohen Komplexität der Simulationsmodelle konzentriert sich ns2 in seiner ursprünglichen Form auf klassische, statische Netzwerkszenarien mit kleinen bis mittelgroßen Topologien [123]. Um die Leistungsfähigkeit zu steigern, stellt *PDNS*¹¹ [151] Methoden zur verteilten Simulation mit Hilfe von MPI zur Verfügung. Analog zu OMNet++ wird die Parallelisierung durch blockierende Synchronisationsmechanismen realisiert, was sich deutlich auf den erreichbaren *Geschwindigkeitszuwachs*, im weiteren Verlauf als *Speedup* bezeichnet, auswirkt. Desweiteren bringt die Einbindung der PDNS Erweiterung in den ns2-Framework weitere konzeptionelle Einschränkungen mit sich, welche bei der Umsetzung eigener Simulationsszenarien beachtet werden müssen.

Als Konsequenz aus den genannten Einschränkungen wurde im Rahmen dieser Arbeit eine Eigenentwicklung realisiert, welche die Anforderungen in Hinsicht auf Skalierbarkeit, Funktionsumfang und Bedienbarkeit erfüllen. Die implementierte Simulationsumgebung sowie die theoretischen Grundlagen werden nun detailliert vorgestellt.

⁸The Enhanced Network Simulator

⁹Module-based Wireless Node

¹⁰Multi-InteRfAce Cross Layer Extension

¹¹Parallel/Distributed NS

5.2. Theoretische Grundlagen

SimANet soll sowohl sehr effizient arbeiten als auch eine modulare Erweiterbarkeit gewährleisten. Diese Ziele schließen sich zunächst gegenseitig aus, da eine Modularisierung meist eine Verlängerung der internen Datenpfade im Verarbeitungsprozess mit sich bringt. Bei der Konzeption des Simulationsmodells müssen diese Rahmenbedingungen beachtet und ein hinreichend guter Kompromiss zwischen Effizienz und Modularität gefunden werden.

5.2.1. Simulationsmodell

Für die Modellierung der Simulationswelt und der darin enthaltenen Knoten wurde ein *periodisches Kommunikationsmodell* gewählt. Hierbei wird jeder Knoten nach Ablauf einer festgelegten Periode aktiv und startet seinen Verarbeitungszyklus. Anschließend wechselt er wieder in einen Wartezustand. Bei der Knoteninitialisierung wird Wartezeit innerhalb eines definierten Intervalls zufällig gewählt. Einhergehend mit der zeitlichen Diskretisierung der Verarbeitung bietet sich eine *eventbasierte Simulation* an. Eine *Event-Warteschlange (Event-Queue)* steuert dabei den zeitlichen Ablauf der simulierten Knoten. Desweiteren lassen sich mit Hilfe einer eventbasierten Simulation zeitgesteuerte Funktionen sowie benutzerdefinierte zeitliche Abläufe leicht umsetzen, da die jeweiligen Aktionen einfach in die bereits vorhandene Event-Queue integriert werden können.

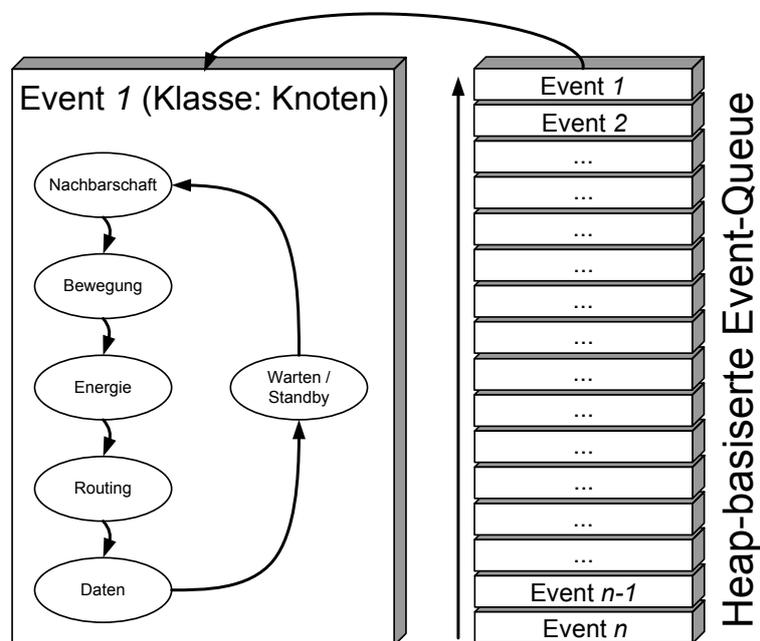


Abbildung 5.2.: Bearbeitungszyklus der Event-Queue am Beispiel eines Knoten-Events.

Die Aktivierung der jeweiligen Knoten repräsentiert gleichzeitig das *Alive-Signal (Heartbeat)* der Funkeinheit. Nur im aktiven Zustand erhalten die Knoten die Möglichkeit, eigene Aufgaben

5. Simulationsumgebung SimANet

abzuarbeiten. Diese Aktivitäten umfassen dabei die Nachbarschaftsverwaltung, Positionsaktualisierung, das Energiemanagement sowie Routing und Datenkommunikation (Abbildung 5.2). Auf die genannten Punkte soll nun detaillierter eingegangen werden.

5.2.2. Bewegungsmodelle

SimANet fokussiert die Simulation mobiler Ad Hoc und Sensornetzwerke. Dabei wird die Knotendynamik durch ein zugrunde liegendes Bewegungsmodell beschrieben. Innerhalb des eventbasierten Simulationskonzeptes werden dazu die Knotenbewegungen diskretisiert. Auf einige Ansätze soll im Folgenden näher eingegangen werden.

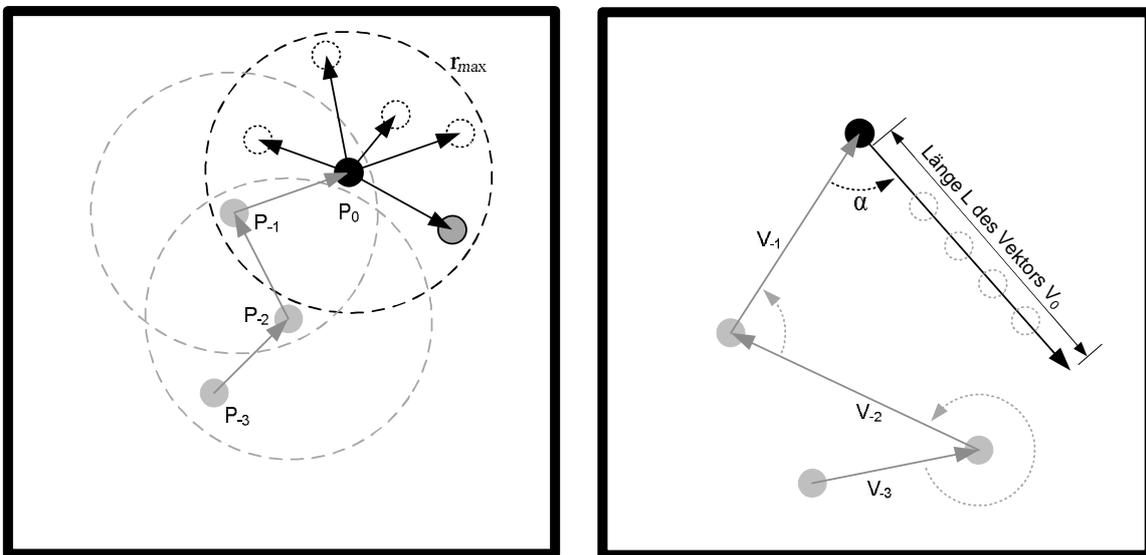


Abbildung 5.3.: Schematische Darstellung zweier unterschiedlicher Bewegungsmodelle: Das *Brownsche Modell* (links) mit zufälliger Bestimmung der neuen Position innerhalb eines definierten Radius r . P_{-3} bis P_{-1} repräsentieren vorhergehende Positionen, P_0 stellt die aktuelle Position dar. Rechts das *Random Waypoint Model* mit zufällig gewählten Bewegungsvektoren V_{-3} bis V_{-1} , welche sich aus dem Winkel α und der Länge L ergeben. Die so entstandenen Pfade werden anschließend mit einer definierten Geschwindigkeit abgefahren.

Brownsches Modell

Ein sehr einfaches Bewegungskonzept repräsentiert das *Brownsche Modell* (Abbildung 5.3 links). Jeder Knoten hat einen definierten Aktionsradius und wählt innerhalb dieser Grenzen eine neue Position zufällig aus. Nach dem Ablauf einer Periode wird die eigene Position in einer atomaren Aktion aktualisiert. Einhergehend mit den zufälligen Positionsänderungen ist die Knotenbewegung in keiner Weise vorhersehbar. Spezifische Aussagen zum Bewegungsmuster der Knoten sind nicht möglich.

Random Waypoint Model

Das *Random Waypoint Movement Model* (RWP - Abbildung 5.3 rechts) erweitert das vorgestellte Brownsche Modell. Jeder Knoten legt initial einen Bewegungspfad in Form eines Vektors fest. Die Länge des Vektors liegt dabei innerhalb eines definierbaren Intervalls und wird zufällig gewählt. Gleiches gilt für die Bewegungsrichtung, die randomisiert zwischen 0 und 359 Grad beträgt. Jeder Knoten bewegt sich mit der gewählten Geschwindigkeit entlang des definierten Vektors. Aus Geschwindigkeit und Richtung wird in jedem diskreten Simulationsschritt die Position entsprechend aktualisiert. Erreicht ein Knoten das Ende seines Bewegungspfades, wird ein neuer Vektor definiert. Aufgrund der gleichmäßigeren Bewegungen mittels RWP sind Vorhersagen deutlich besser möglich als im Brownschen Modell. Auch hier sind durch willkürliche Richtungswechsel Bewegungsmuster aus realen Anwendungsszenarien kaum erkennbar.

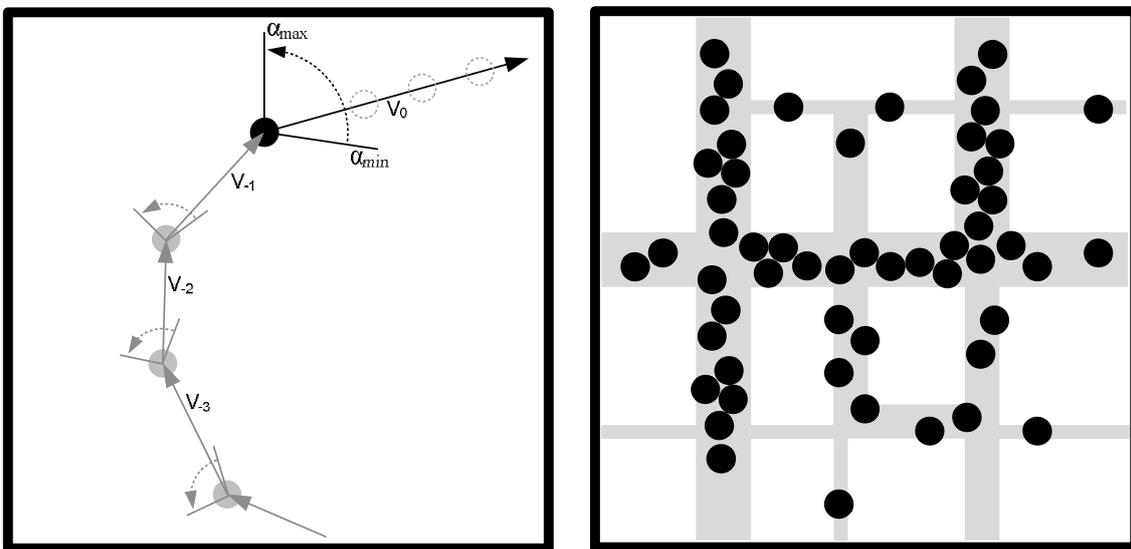


Abbildung 5.4.: Links das auf dem RWP basierende und verfeinerte *Markovian Waypoint Model* mit definierten Grenzen für den Winkel des Vektors. Referenzwert für die Intervallgrenzen ist jeweils der Winkel des vorhergehenden Bewegungsvektors. Das *Urban Mobility Model* wird rechts schematisch dargestellt und verdeutlicht den Bezug zu den gewichteten Kanten (Straßen), auf denen sich die Knoten bewegen dürfen.

Markovian Waypoint Model

Mit Hilfe des *Markovian Waypoint Model* (Abbildung 5.4 links), welches auf dem RWP aufsetzt, können deutlich präzisere Aussagen hinsichtlich individueller Bewegungsmuster getroffen werden. Dazu wird das RWP-Modell dahingehend erweitert, dass die Wahl der nächsten Zielposition nicht vollkommen zufällig erfolgt, sondern in Abhängigkeit des vorherigen Bewegungspfades. Für den Folgevektor wird die Richtungskomponente daher nur relativ zur Vorhergehenden innerhalb eines definierten Intervalls zufällig bestimmt. Unkoordinierte, chaotische Bewegungsfolgen, wie sie im Brownschen und im RWP Modell häufig auftreten, werden somit ausgeschlossen. Bedingt durch die Bewegungsmuster ist das Markovian Waypoint Modell

5. Simulationsumgebung SimANet

besonders für Simulationen im Automobilbereich interessant und sinnvoll.

Urban Mobility Model

Im Bereich der *Vehicle Ad Hoc Networks (VANETs)* oder *Car-to-Car-Communication (C2C)* stehen besonders Simulationen im urbanen Umfeld im Fokus der Betrachtungen. Daher emuliert das *Urban Mobility Model* (Abbildung 5.4 rechts) die schachbrettartigen Verkehrssysteme großer Metropolen. Knoten bewegen sich hierbei auf gewichteten Kanten, welche ressourcenbeschränkte Straßen darstellen. Auf diesen Bahnen können sich, je nach aktueller Auslastung, eine unterschiedliche Anzahl Knoten mit festgelegten Geschwindigkeiten bewegen. Auf der Basis des Urban Mobility Modells können somit besonders Stress- und Überlastsituationen emuliert werden.

5.2.3. Modellierung von Hindernissen

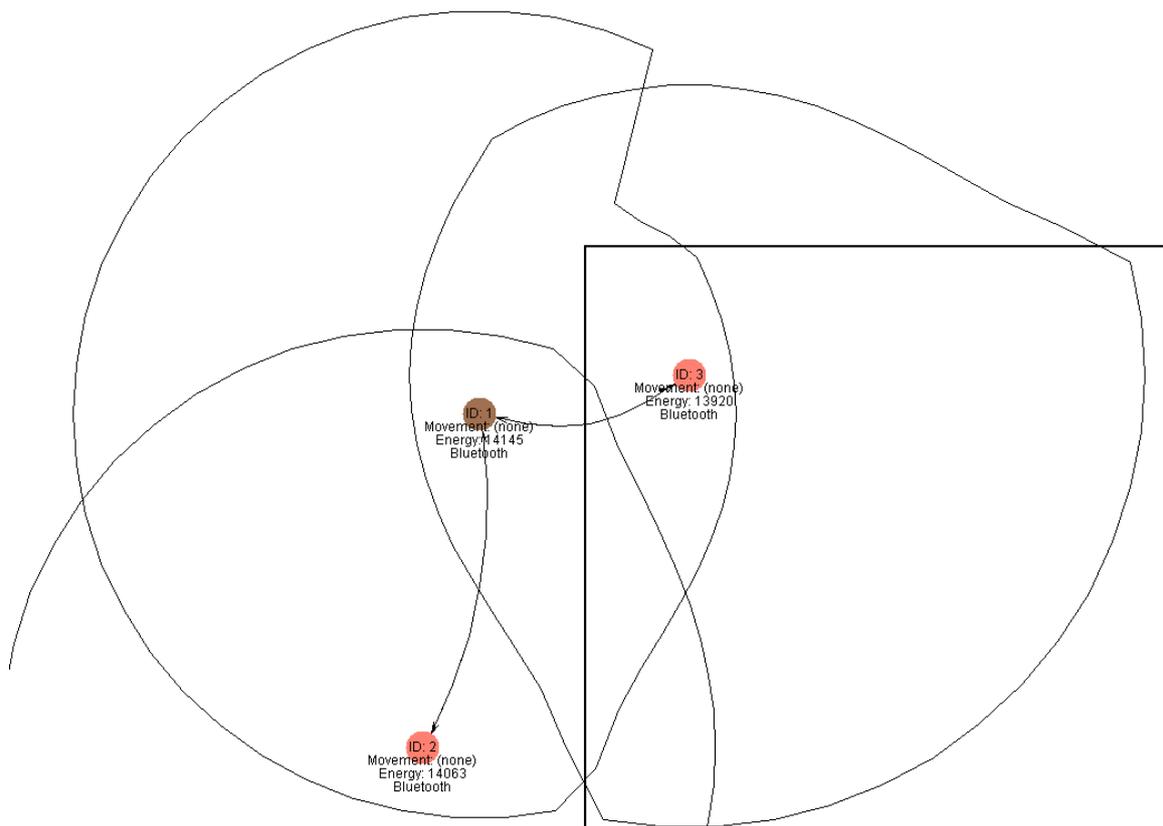


Abbildung 5.5.: Modellierung von Hindernissen innerhalb der Simulationsumgebung

Um realitätsnahe Anwendungsszenarien zu gestalten, ist die Modellierung von Hindernissen erforderlich. SimANet ermöglicht es dem Nutzer auf einfache Art und Weise geometrische Objekte mit Hilfe der grafischen Nutzeroberfläche oder per Skript in die simulierte Landschaft

zu integrieren. Dabei wird jedem erstellten Objekt ein individueller Dämpfungswert zugeordnet. Im Zuge einer Kommunikation durch das Hindernis hindurch wird die maximal mögliche Sendereichweite entsprechend des Dämpfungsfaktors angepasst. Alternativ ist auch eine paketorientierte Beeinflussung der Kommunikationseigenschaften realisierbar. Dabei entspricht der hinterlegte Dämpfungswert einer definierten Wahrscheinlichkeit für den Paketverlust. In Kombination mit der Sendedistanz wird dabei ein realitätsnahes Kommunikationsverhalten erzielt.

Da dieses Modellierungskonzept auf einer Heuristik basiert, ist das Verhalten jedoch schwer vorhersagbar. Daher wurde sich im Rahmen der vorgestellten Simulationsumgebung eine verbindungsorientierte, statische Anpassung der Sendereichweite mittels definierbarer Dämpfungswerte umgesetzt (siehe Abbildung 5.5).

Im derzeitigen Entwicklungsstand wurde noch kein physikalisch korrektes Wellenausbreitungsmodell über das Medium Luft integriert. Physikalische Phänomene wie Reflexion, Dämpfung oder Interferenzverhalten können somit nicht betrachtet werden. Nähere Ausführungen hierzu werden in Abschnitt 5.4 erläutert.

5.2.4. Energiemodell

SimANet integriert ein Modell zur dynamischen Anpassung der verfügbaren Energieressourcen. Dabei wird jede Aktion eines Netzwerkknotens in der Topologie durch abstrakte Energiekosten repräsentiert. Die Menge der Aktionen umfasst dabei Verarbeitungsprozesse für die Protokollkonvertierung, die Initialisierung eines neuen Kommunikationskanals innerhalb des gewählten Funkstandards sowie Send- und Empfangsprozesse. Desweiteren verursacht die Betriebsbereitschaft eines Netzwerkknoten fixe Basiskosten, was eine periodische Aktualisierung des Energielevels aller Knoten erforderlich macht. Die Modellierung der Ressource Energie wird im weiteren Verlauf des Kapitels bei den jeweiligen Simulationsreihen nochmals konkretisiert.

5.2.5. Nachbarschaftsverwaltung

Zur Verwaltung der simulierten Knotenlandschaft muss eine geometrische Datenstruktur genutzt werden, die es ermöglicht, eine große Anzahl von Umkreisanfragen effizient zu beantworten. Dabei liefert jede Anfrage eine Liste aller benachbarten Knoten innerhalb des definierten Funkradius zurück. Bei der Auswahl einer geeigneten Datenstruktur mussten unterschiedliche Kriterien beachtet werden:

1. Auswahl einer *geometrischen* Datenstruktur, in der jeder Punkt kartesische Koordinaten besitzt.
2. Auswahl einer dynamischen Datenstruktur, welche leistungsfähig genug ist, mehrere tausend Knoten zu verwalten.
3. Das Finden gespeicherter Punkte innerhalb eines gewählten Umkreisradius muss effizient

5. Simulationsumgebung SimANet

und ressourcenschonend umsetzbar sein.

Die auf der Datenstruktur ausgeführten Methoden beschränken sich dabei auf:

- Finden (*Find*): Suche eines Punktes
- Einfügen (*Insert*): Hinzufügen eines Punktes in die bestehende Datenstruktur
- Löschen (*Delete*): Entfernen eines gewählten Punktes und die anschließende Reorganisation der Datenstruktur
- Bewegen (*Move*): Positionsänderung eines Punktes und die anschließende Reorganisation der Datenstruktur
- Umkreissuche (*RangeSearch*): Rückgabe der Menge von Punkten, die sich in einem gewählten Radius r um eine definierte Position $P = (X, Y)$ befinden

Eine solche geometrische Datenstruktur stellt der *Quadtree* dar. Dieser bildet die Grundlage für mehrere, weiterentwickelte Datenstrukturen wie den *Compressed Quadtree* [32], den *Bucket Quadtree* [176] und schließlich dem *Skip Quadtree* [61], welcher im Rahmen von SimANet zur Anwendung kam.

Die Idee hinter einer Quadtree Baumstruktur ist die rekursive Aufteilung des Raumes in Quadranten. Die Anzahl der Quadranten beträgt $n = 2^d$ (d – Dimension). Solch ein Quadrant kann leer sein, genau einen Punkt enthalten oder wiederum einen Baum repräsentieren. Abbildung 5.6 zeigt einen Quadtree und dessen geometrische Interpretation.

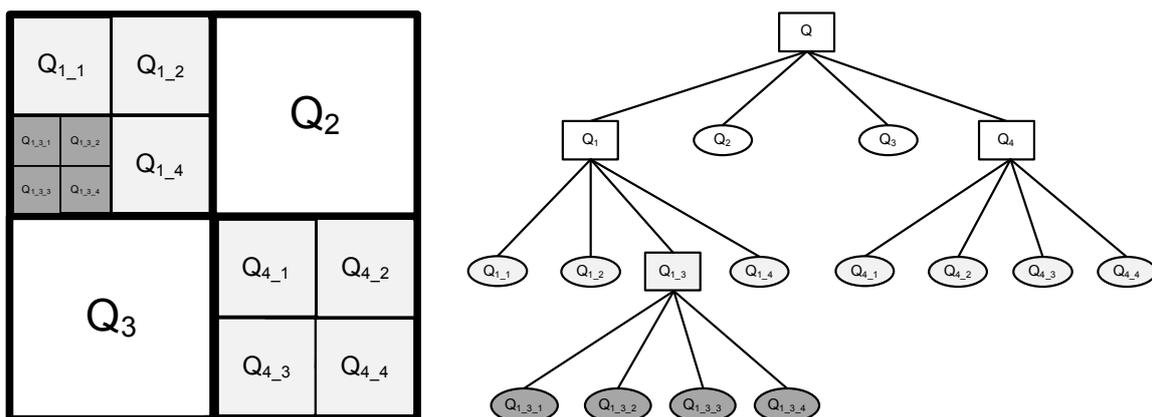


Abbildung 5.6.: Beispielhafter Quadtree: Aufteilung in jeweils vier Quadranten Q_1 bis Q_4 . Links die geometrische Repräsentation und rechts die Abbildung auf die Datenstruktur.

Die Datenstruktur hat dabei einen entscheidenden Nachteil. Die Tiefe des Baumes kann im Worst Case linear zur Anzahl der Knoten/Punkte ansteigen und ist stark abhängig von der räumlichen Verteilung dieser Punkte. In Folge dessen steigt auch der benötigte Speicherplatz

erheblich an und es kann bei einer Suchanfrage keine Antwortzeit von $O(\log n)$ ($n = \#Knoten$) garantiert werden.

Compressed Quadtree

Der *Compressed Quadtree* versucht, unnötige Verzweigungen in der Datenstruktur zu vermeiden indem er alle inneren Knoten der Baumstruktur, die nur einen Sohn besitzen einspart. Stattdessen verbindet er die eingehende Kante des Vaterknotens direkt mit dem jeweiligen Sohnknoten. Auf diese Weise kann die Tiefe des Baumes deutlich reduziert werden und die Antwortzeit bei Suchanfragen wird verringert (Abbildung 5.7).

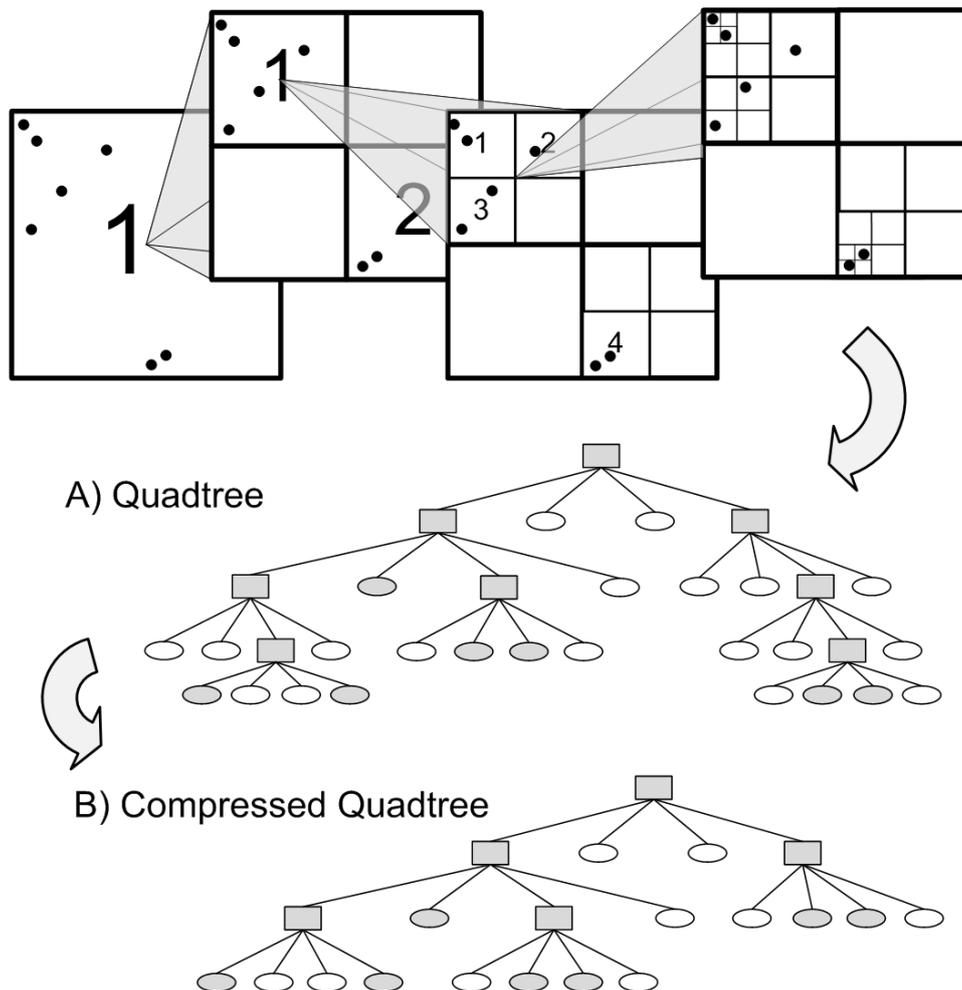


Abbildung 5.7.: Abbildung einer Punktemenge als Quadtree und Compressed Quadtree. Deutlich erkennbar ist die Reduzierung der Baumtiefe. Innere Knoten mit nur einem Sohn werden entfernt.

Definition: (*Interesting Square*) Ein **Interesting Square** ist die Wurzel eines *Quadtrees* oder ein innerer Knoten, der mindestens zwei nichtleere Söhne besitzt [61].

Anschaulich ausgedrückt bedeutet dies, dass ein Quadtree genau dann zu einem Compressed Quadtree mutiert, wenn alle inneren Knoten die kein *interesting Square* repräsentieren aus der

5. Simulationsumgebung SimANet

Datenstruktur entfernt werden. Betroffene Verbindungen werden anschließend entsprechend angepasst. Die Laufzeiten der einzelnen Operationen auf diese Datenstruktur betragen im Average Case $\log n$ ($n = \#Knoten$) [32].

Dennoch kann ein Worst Case Szenario mit linearer Baumtiefe ($O(n)$) provoziert werden und eine logarithmische Antwortzeit auf Suchanfragen ist nicht sichergestellt. Abbildung 5.8 stellt einen solchen generierten Fall dar.

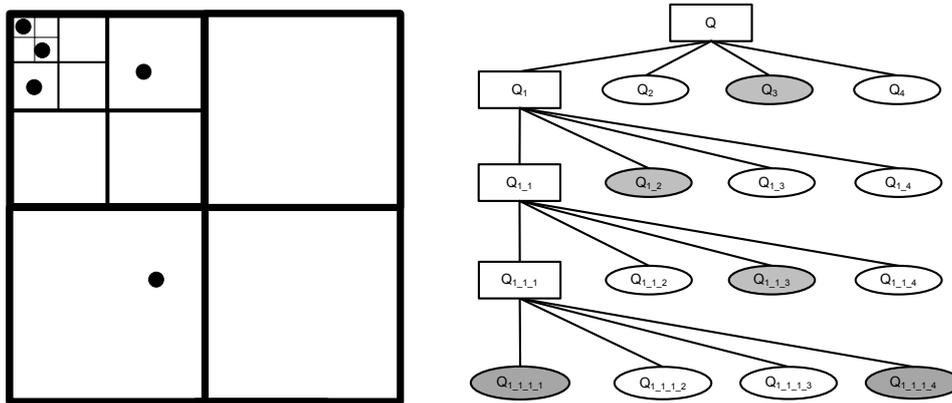


Abbildung 5.8.: Konstruierter Fall eines Worst Case Szenarios beim Compressed Quadtree. Die Baumtiefe ist linear zur dargestellten Knotenmenge.

Skip Quadtree

Als direkte Weiterentwicklung entstand schließlich der *Skip Quadtree*, welcher die Schwäche des Compressed Quadtree annähernd eliminiert. Eine Skip Quadtree Datenstruktur besteht dabei nicht mehr aus einem einzelnen Quadtree, sondern aus einer Menge verketteter, geometrischer Baumstrukturen. Er repräsentiert somit eine Liste von Quadtrees, genauer Compressed Quadtrees ($Q_0 \dots Q_k$) ($k = \#Quadtrees$).

Jeder Knoten s wird initial in den Quadtree Q_0 eingefügt. Die Knotenmenge S von Q_0 ($\Rightarrow S_0$) entspricht folglich der gesamten Netztopologie. Anschließend wird ein eingefügter Knoten mit einer festgelegten Wahrscheinlichkeit p (hier mit $p = 0.5$ definiert) von Q_i in den benachbarten Quadtree Q_{i+1} übernommen. Die Knotenmenge S_i ist also wie folgt definiert:

$$S_i = \{s \in S_{i-1} : \text{rand}() \geq 0,5\}$$

Somit sinkt mit ansteigendem i die in Q_i enthaltene Knotenmenge S_i bis schließlich in S_k ausschließlich ein Element enthalten ist.

Als direkte Schlussfolgerung ist die in der Liste enthaltene Anzahl Quadrees k nicht vorhersehbar sondern wird durch die zufällige Wahrscheinlichkeit bestimmt, wie oft ein Knoten von Q_i in Q_{i+1} kopiert wird.

Alle in einem Quadtree Q_i vorhandenen Quadrate (bestehend aus je vier Quadranten) werden,

sofern in ihren direkten Nachbarn Q_{i-1} und Q_{i+1} vorhanden, mit Hilfe von bidirektionalen Zeigern verlinkt.

Die folgende Abbildung 5.9 zeigt einen beispielhaften Skip Quadtree.

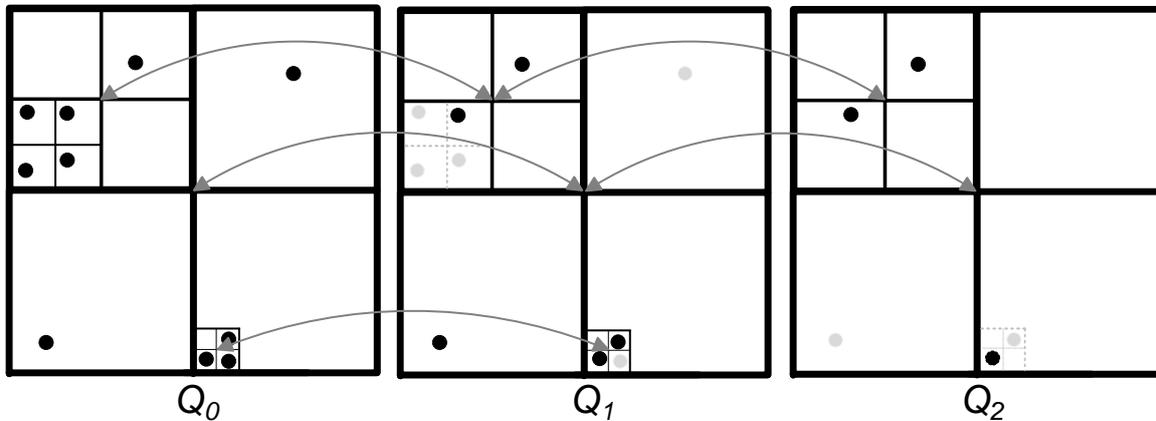


Abbildung 5.9.: *Randomised Skip Quadtree* Datenstruktur, bestehend aus drei doppelt verlinkten Compressed Quadrees (Q_0 , Q_1 , Q_2). Die grauen Pfeile verdeutlichen die Zeiger zwischen den jeweiligen Quadranten. Ein Knoten wird dabei mit einer definierten Wahrscheinlichkeit von 0.5 von Q_0 nach Q_1 und von Q_1 nach Q_2 kopiert. Somit sinkt in jedem Quadtree die Anzahl der repräsentierten Knoten und dadurch die Komplexität des Baumes. Nicht kopierte Knoten werden in diesem Beispiel grau dargestellt.

Alle Operationen können mit Hilfe des Skip Quadrees im Erwartungswert und mit sehr hoher Wahrscheinlichkeit in logarithmischer Laufzeit ($\log n$) ($n = \#Knoten$) ausgeführt werden.

Die Wahrscheinlichkeit für ein Worst Case Szenario ist im Vergleich zum Compressed Quadtree nochmals signifikant geringer. Theoretisch ist dieser genau dann konstruiert, wenn aufgrund der Wahrscheinlichkeit niemals ein Knoten vom initialen Quadtree Q_0 nach Q_1 kopiert werden würde. Folglich repräsentiert der Skip Quadtree dann einen einfachen Compressed Quadtree mit allen daran gebundenen Eigenschaften. In einem weiteren, konstruierten Worst Case wird ein Knoten mit der Wahrscheinlichkeit $p = 0.5$ immer wieder kopiert, wodurch die Anzahl der Quadrees und somit der Speicherbedarf unbegrenzt wächst. Dieses Szenario kann jedoch durch einfache Mechanismen eliminiert werden, bzw. durch eine Abbruchbedingung für das Kopieren eines Knotens, wenn der vorhergehende und aktuelle Quadtree (Q_{i-1} , Q_i , $i > 1$) bereits ausschließlich ein Element enthält. Alternativ kann durch eine zusätzliche Bedingung die Knotenmenge in Quadtree Q_{i+i} innerhalb eines definierten Prozentsatzes P ($25\% < P < 75\%$) der Knotenmenge in Q_i begrenzt werden.

Suchanfragen / Bereichssuche

Wie bereits ausführlich erläutert, bildet die Skip Quadtree Datenstruktur die simulierte Knotenmenge und deren räumliche Verteilung ab. Die Suche nach einem beliebigen Knoten im Raum startet zunächst mit einer Baumsuche im rechten Quadtree Q_{k-1} . Bleibt diese initiale Suche er-

5. Simulationsumgebung SimANet

folglos, ist der Knoten in diesem Quadtree nicht enthalten und die Suche springt über den Zeiger des kleinsten umgebenden Quadrats zum nächsten links verknüpften Quadtree Q_{k-2} . Dort kann die Suche in gleicher Form fortgesetzt werden. Folglich werden die Bäume von rechts nach links durchsucht.

Die Gesamtlaufzeit beträgt $\log n$ ($n = \#Knoten$) [61] in Erwartung und mit hoher Wahrscheinlichkeit, wobei die Wahrscheinlichkeit für ein Worst Case Szenario gegen 0 tendiert.

Das Einfügen und Löschen einzelner Knoten ist in der gleichen Laufzeit wie die Such-Operation möglich. Beim Einfügen in Q_0 ist der entsprechende Quadrant im Quadtree entweder bereits vorhanden oder es wird in konstanter Zeit genau ein innerer Knoten in der Datenstruktur angelegt. Jeder Knoten repräsentiert ein neues Blatt im Quadtree. Die Prozedur wiederholt sich je nach Wahrscheinlichkeit in den Quadtree $Q_1 \dots Q_{k-1}$. Da k mit sehr hoher Wahrscheinlichkeit $\leq \log n$ beträgt, resultiert daraus ebenfalls ein Erwartungswert für die Laufzeit von $\log n$ ($n = \#Knoten$). Das Löschen von Knoten aus der Skip Quadtree Datenstruktur läuft analog zum Einfügen ab. Verzeigerungen werden jeweils entsprechend angepasst.

Die Bereichssuche nach allen in einem definierten Radius befindlichen Knoten gestaltet sich aufgrund der kreisförmigen Struktur schwieriger. Der Skip Quadtree ist jedoch in der Lage, unscharfe Bereichsanfragen in einer logarithmischen Laufzeit ($\log n + \alpha$) zu beantworten. Der Faktor α hängt dabei direkt von dem gewählten Unschärfegrad ϵ ab. Abbildung 5.10 verdeutlicht die Problematik im Grenzbereich des Suchradius und die Bedeutung von ϵ . In Abschnitt 5.6 wird der Geschwindigkeitsvorteil durch die Einführung eines solchen Randbereiches anhand verschiedener Messreihen nachgewiesen. Grundsätzlich muss aber konzeptbedingt eine gewisse "Unschärfe" in Kauf genommen werden.

Während der Anfrage für eine Bereichssuche innerhalb eines Quadrees sind drei Arten von Knotenmengen zu unterscheiden:

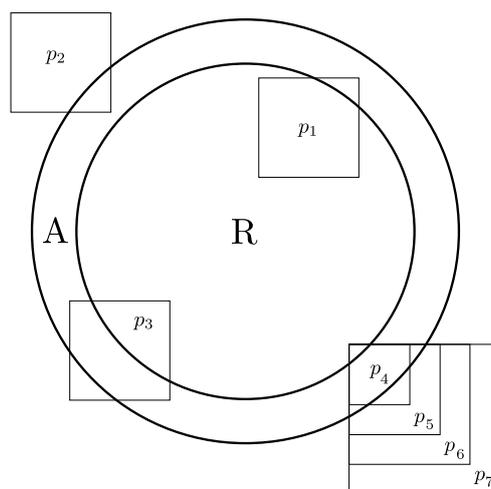


Abbildung 5.10.: Schematische Darstellung der unterschiedlichen Quadrattypen bei einer Bereichsanfrage (nicht maßstabsgetreu). Die Dicke der Fläche A repräsentiert den Unschärfegrad ϵ . p_1 (komplett enthalten), p_2 (komplett ausgeschlossen) und p_3 (partiell enthalten). $p_4 \dots p_7$ sind ebenfalls partiell enthalten, aber nur p_4 und p_5 sind kritisch.

- **Komplett enthalten** - Ein in der Datenstruktur enthaltener Quadrant ist vollständig in der Kreisfläche *inklusive* Grenzbereich enthalten (Abb. 5.10 - p_1).
- **Komplett ausgeschlossen** - Ein in der Datenstruktur enthaltener Quadrant liegt vollständig außerhalb der definierten Kreisfläche *exklusive* Grenzbereich (Abb. 5.10 - p_2).
- **Partiell enthalten** - Ein in der Datenstruktur enthaltener Quadrant liegt partiell in der definierten Fläche (Abb. 5.10 - p_3).

Definition: Ein partiell enthaltener Quadrant ist genau dann **kritisch**, wenn:

- die enthaltene Fläche größer als die ausgeschlossene Fläche ist
- seine Sohnknoten komplett enthalten sind

```

RANGESearch(p, r, ε)
  ▷ letzten non-critical Wurzelknoten suchen
1  Q ← Q0
2  while Q.right ≠ critical do
3    Q ← Q.right

  ▷ Eigentliche Bereichssuche starten
4  RecursiveRangeSearch(Q, p, r, ε)

RECURSIVERANGESearch(Q, p, r, ε)
  ▷ Quadrat nach IN, OUT, STABBING oder CRITICAL klassifizieren
1  switch classifyCritical(Q, p, r, ε)
2    case IN:
3      addToResult(Q.getAllPoints())
4      return
5    case OUT:
6      return
7    case STABBING:
8      for all quadrants q in Q do
9        RecursiveRangeSearch(q, p, r, ε)
10     return
11   case CRITICAL:
12     if Q.left ≠ NULL then
13       RecursiveRangeSearch(Q.left, p, r, ε)
14     return
15   else
16     ▷ kein Zeiger mehr nach links vorhanden, deswegen Tiefensuche
17     for all quadrants q in Q do
18       RecursiveRangeSearch(q, p, r, ε)
19     return

```

Abbildung 5.11.: Ablauf einer Bereichsanfrage in Quadtree Q um den Punkt p und im Radius r mit dem definierten Unschärfegrad ϵ (Pseudocode-Darstellung).

5. Simulationsumgebung SimANet

Folglich müssen für die Bereichsanfragen nur *komplett enthaltene* und *kritische, partiell enthaltene* Quadranten behandelt werden. Am Beispiel von Abbildung 5.10 demnach nur p_1 , p_3 , p_4 und p_5 . Die Suche nach kritischen Quadranten ist eine Erweiterung der Such-Operation in der Datenstruktur. Abbildung 5.11 zeigt den Ablauf einer solchen Bereichsanfrage als Pseudocode.

Leistungsmessungen in Abhängigkeit des gewählten ϵ Wertes für die Bereichssuche sowie Messungen bezüglich der Leistungsfähigkeit des Skip Quadrees im Vergleich zu anderen Datenstrukturen werden im weiteren Verlauf dieses Kapitels vorgestellt und analysiert.

5.2.6. Parallelisierung

Mit Hilfe einer Parallelisierung ist es möglich, geeignete Algorithmen durch Zuhilfenahme einer parallelen Berechnungsumgebung bzw. eines Rechnerverbundes zu beschleunigen. Das Verhältnis zwischen einfacher Berechnungszeit auf Basis einer einzelnen Recheneinheit (*Singleprozessor*) und der Berechnungszeit innerhalb eines Rechnerverbundes (*Multiprozessor / Multicore*) wird als *Speedup* s bezeichnet. Dieser *Geschwindigkeitsfaktor* kann dabei Werte im Bereich $0 \leq s \leq n$ annehmen ($s \in \mathbb{Q}$), wobei n die Anzahl der verfügbaren Recheneinheiten darstellt. Der erreichbare Speedup ist stark problemspezifisch und abhängig von der Eingabegröße.

Bei der Parallelisierung von Netzwerksimulationen treten konzeptionelle Schwierigkeiten auf. Diese resultieren aus der Tatsache, dass Simulationsanwendungen im Allgemeinen nicht rechenzeitintensiv sondern stark kommunikationslastig sind, was eine parallele Abarbeitung im klassischen Sinne verhindert. Somit ist die unabhängige Berechnung einzelner Teilkomplexe nicht möglich. Rechenintensive Operationen treten kaum auf und sind daher zu vernachlässigen. Betrachtet man die rechnerinternen Systembusse, ergibt sich das zentrale Problem aus dem vergleichsweise langsamen Verbindungsnetzwerk zwischen den Berechnungseinheiten. Die Kommunikation über das Netzwerk repräsentiert hierbei den Flaschenhals während der Simulation. Für eine 100%ig genaue Simulation muss in einer event-basierten Umgebung zu jedem Zeitpunkt jedes Ereignis im Netzwerk synchronisiert werden. Folglich ergibt sich aus den resultierenden Latenzen im Netzwerk mit hoher Wahrscheinlichkeit kein verwertbarer Speedup.

Desweiteren muss im Netzwerk eine einheitliche Zeitbasis geschaffen werden, um auf dieser Basis eine möglichst exakte Simulation zu gewährleisten. Dabei gestaltet sich die Synchronisation als nicht trivial. In Abschnitt 3.1.3 wurde bereits auf unterschiedliche Verfahren aus dem Bereich der Zeitsynchronisation eingegangen. Wird ein zentrales Zeitsignal in periodischen Abständen über das Netzwerk verbreitet, so entstehen periodische Spitzen in der Netzlast. Innerhalb der gewählten Periode können die Zeiten der einzelnen Simulationseinheiten auseinanderdriften. Abhängig vom definierten Intervall muss ein Kompromiss zwischen Zeitgenauigkeit und erzeugter Netzlast gefunden werden.

Unscharfe Simulation

Als direkte Konsequenz folgt daraus der Kompromiss zwischen Genauigkeit der Simulation (Netzwerksynchronisation) und erreichbarem Speedup. In Folge dessen müssen geeignete Realisierungsmodelle gefunden werden, die im Bereich paralleler Netzwerksimulation einen angemessenen Speedup erzielen und dabei die Anforderungen an die Simulationsgenauigkeit erfüllen. Für die Realisierung der SimANet Umgebung wurde ein *Sliding Time Window* Verfahren konzipiert und implementiert, dessen Grundlagen sich auf die *Sliding-Window* Ansätze im Bereich der Netzwerkkommunikation beziehen. Sie dienen dabei zur Flusskontrolle innerhalb des Übertragungsmediums und definieren ein Zeitfenster, in dem der Sender eine bestimmte Datenmenge übertragen darf, ohne dass hierfür vom Empfänger eine Bestätigung vorliegt. Daraus resultiert eine deutlich optimierte Latenz für die Kommunikation.

Diese Ansätze wurden auf die Problematik der parallelen Simulation abgebildet. Das hier entwickelte Verfahren nimmt für die simulierte Zeit eine definierbare *Unschärfe (Time Drift t_d)* in den einzelnen Rechenknoten bewusst in Kauf. Jede Berechnungseinheit verwaltet dabei eine lokale Zeitschranke und ein definiertes Zeitfenster, welches initial (Simulationszeit $t = 0$) jeweils auf den Wert t_d gesetzt wird. Ist dieses Fenster zur Hälfte abgelaufen, beginnt der jeweilige Rechenknoten, Informationen über die aktuellen Zeiten von allen direkt benachbarten Berechnungseinheiten anzufordern.

```

INITIALISATION:
1 long timeLimit ←  $t_d$ 
2 list listResponseTimes ←  $\emptyset$ 
3 int countResponsesNeeded ← 0
4 TIME_HALF ←  $\frac{t_d}{2}$ 
5 TIME_UP ←  $t_d$ 

IF TIME_HALF IS REACHED:
1 countResponsesNeeded ← #neighbour CPUs
2 listResponseTimes.clear()
3 for each neighbour CPU  $p$  do
4   Network.Send( $p$ , TIME_REQUEST, currentTime() +  $t_d$ )

IF TIME_UP IS REACHED:
  ▷ Simulation unterbrechen, wenn noch Antworten von Nachbarrechnern fehlen
1 if countResponsesNeeded  $\neq$  0 then
2   pauseSimulation()

```

Abbildung 5.12.: Darstellung des *Sliding Time Window* Algorithmus - Teil 1.

Aus allen Antworten wählt er die kleinste simulierte Zeit t_{min} aus (was der langsamsten Recheneinheit entspricht) und berechnet die neue Zeitschranke aus der Summe von t_d und t_{min} . Somit ist die eigene simulierte Zeit niemals um mehr als t_d größer als die des langsamsten benachbarten Berechnungsknotens. Die Aktualisierung der eigenen Zeitschranke wird bereits

5. Simulationsumgebung SimANet

nach Ablauf der ersten Hälfte des gesetzten Zeitfensters angestoßen. Folglich bleiben relativ große Zeitpuffer für den Empfang der Antwortpakete aller benachbarten Rechenknoten, wodurch Unterbrechungen in der Simulation weitestgehend vermieden werden. Die Abbildungen 5.12 und 5.13 verdeutlichen den Ablauf als Pseudocode.

```
ON INCOMING TIME_REQUEST:
  ▷ Nachbarrechner bittet, weitersimulieren zu dürfen
1  if timeLimit ≥ msg.requestedTime then
    ▷ Zeitlimit im Rahmen, Antwort sofort versenden
2    Network.Send(msg.source, TIME_RESPONSE, currentTime() + td)
  else
    ▷ Angefragte Zeit zu groß, Antwort verzögern
3    CreateDelayedTransmitEventWhenConditionIsReached()

ON INCOMING TIME_RESPONSE:
1  countResponsesNeeded ← countResponsesNeeded - 1
2  listResponseTimes.add(msg.time)
3  if countResponsesNeeded = 0 then
    ▷ Alle Nachbarrechner haben geantwortet
    ▷ Nun eigene Zeitschranke aktualisieren
4    timeLimit ← listResponseTimes.getMinimum()

    ▷ TIME_HALF und TIME_UP aktualisieren
5    TIME_HALF ← currentTime() +  $\frac{timeLimit - currentTime}{2}$ 
6    TIME_UP ← timeLimit
7    listResponseTimes.clear()
8    if simulation is paused then
9      continueSimulation()
```

Abbildung 5.13.: Darstellung des *Sliding Time Window* Algorithmus - Teil 2.

In Hinsicht auf die praktische Umsetzung des Verfahrens stellt die optimale Steuerung von t_d das wesentliche Problem dar. Im Falle eines zu groß gewählten t_d sinkt zwar der Overhead für die Synchronisation über das Netzwerk auf ein Minimum. Gleichzeitig wird die Simulation aber zu ungenau für verwertbare Aussagen. Bei einem zu klein gewählten t_d wird sich die hohe Simulationsgenauigkeit durch einen extremen Synchronisationsaufwand erkauft. Damit einhergehend wird auf einen Geschwindigkeitsvorteil durch die Parallelisierung verzichtet. Wird t_d mit der durchschnittlichen Latenzzeit der Netzwerkstruktur definiert, treten im optimalen Fall keine Unterbrechungen der Simulation auf. Da die Netzwerklast jedoch stark variieren kann, ist das Verhalten nicht vorhersagbar.

Partitionierung

Die Aufteilung der Simulationsfläche mit den darin enthaltenen Knoten auf die verfügbare, parallele Rechenlandschaft stellt ein nicht triviales Problem dar. Eine optimale Aufteilung ist dabei nur in statischen Zuständen einer gegebenen Netzwerktopologie möglich, indem neben der Rechenleistung der jeweiligen Berechnungseinheiten auch das Kommunikationsverhalten berücksichtigt werden muss. In Folge der Knotenbewegung und der sich damit ständig verändernden Netzwerktopologie ist dies nicht möglich. Somit müssen Verfahren gefunden werden, welche einen Kompromiss zwischen Berechnungsaufwand und Güte des Partitionierungsergebnisses finden und eine dynamische Verteilung zur Laufzeit der Simulation zulassen.

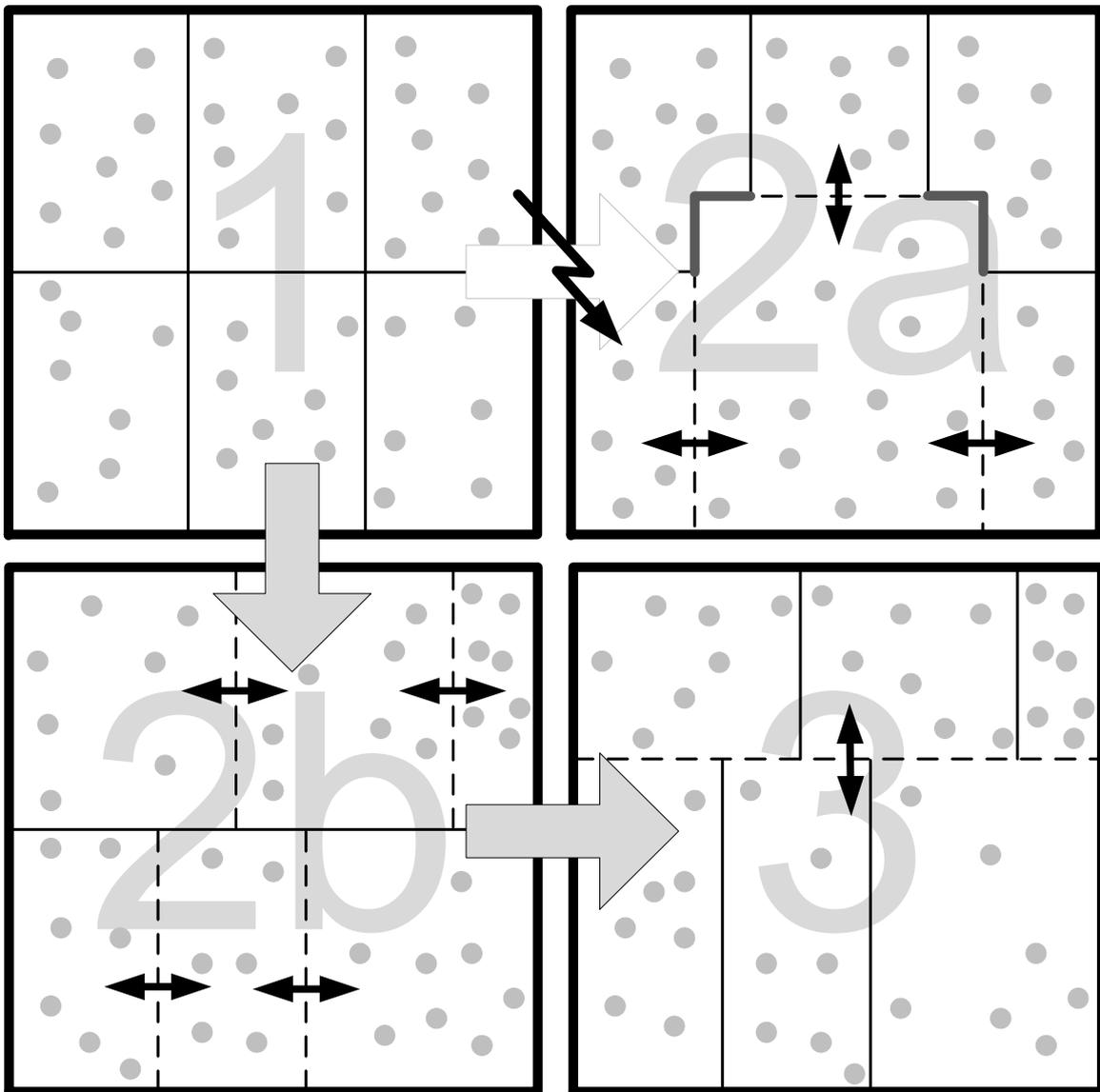


Abbildung 5.14.: Parallele Simulation: Dynamische Adaption der Sektorgrenzen je nach Rechenlast innerhalb der jeweiligen Recheneinheiten. Konstellationen, die von der normalen Rechteckform abweichen (Polygone) müssen dabei vermieden werden (oben rechts). Der Grenzverschiebungsalgorithmus passt dabei zunächst die vertikalen Grenzen einzelner Sektoren an (links unten). Im Anschluss wird die Verteilung zeilenweise über die horizontalen Grenzen nochmals optimiert (rechts unten).

5. Simulationsumgebung SimANet

Im Rahmen dieser Arbeit wurde eine geometrische Partitionierung umgesetzt, welche Knoten in Abhängigkeit ihrer Positionsdaten den jeweiligen Berechnungseinheiten zuteilt. Durch diese zur Laufzeit durchgeführte, dynamische Grenzverschiebung wird die Aufteilung der Knoten auf die verfügbaren Recheneinheiten optimiert und der Knotenbewegung innerhalb der Simulation Rechnung getragen.

```
EQUALISE-LOAD()
1  while TRUE do
    ▷ Load innerhalb einer Reihe angleichen (horizontal benachbarte Knoten)
2  if rechter Nachbar rn existiert then
3       $l_1 \leftarrow \text{getLoad}()$ 
4       $l_2 \leftarrow rn.\text{getLoad}()$ 
5      if  $\left|1 - \frac{l_1}{l_2}\right| > \text{THRESHOLD}_X$  then
6           $x_1 \leftarrow \text{getMiddlePosition}().x$ 
7           $x_2 \leftarrow rn.\text{getMiddlePosition}().x$ 
8           $dx \leftarrow x_2 - x_1$ 
9           $\text{moveRightBorder} \left( x_1 + dx \cdot \frac{l_2}{l_1+l_2} \right)$ 
10          $\text{commitBordersGlobally}()$ 

    ▷ Load der benachbarten Reihen angleichen
    ▷ Nur ein Knoten pro Reihe (Rowmaster) darf das
11 if Rowmaster and unterer Nachbar un existiert then
12      $l_1 \leftarrow \text{row.getLoad}()$ 
13      $l_2 \leftarrow un.\text{row.getLoad}()$ 
14     if  $\left|1 - \frac{l_1}{l_2}\right| > \text{THRESHOLD}_Y$  then
15          $y_1 \leftarrow \text{getMiddlePosition}().y$ 
16          $y_2 \leftarrow un.\text{getMiddlePosition}().y$ 
17          $dy \leftarrow y_2 - y_1$ 
18          $\text{moveBottomBorder} \left( y_1 + dy \cdot \frac{l_2}{l_1+l_2} \right)$ 
19          $\text{commitBordersGlobally}()$ 

    ▷ Gesamte Schleife regelmäßig ausführen
20      $\text{sleep}(t_{\text{sleep}})$ 
```

Abbildung 5.15.: Darstellung des Algorithmus zur dynamischen Grenzverschiebung im Zuge der Partitionierung einer verteilten Simulation auf die verfügbaren Berechnungseinheiten.

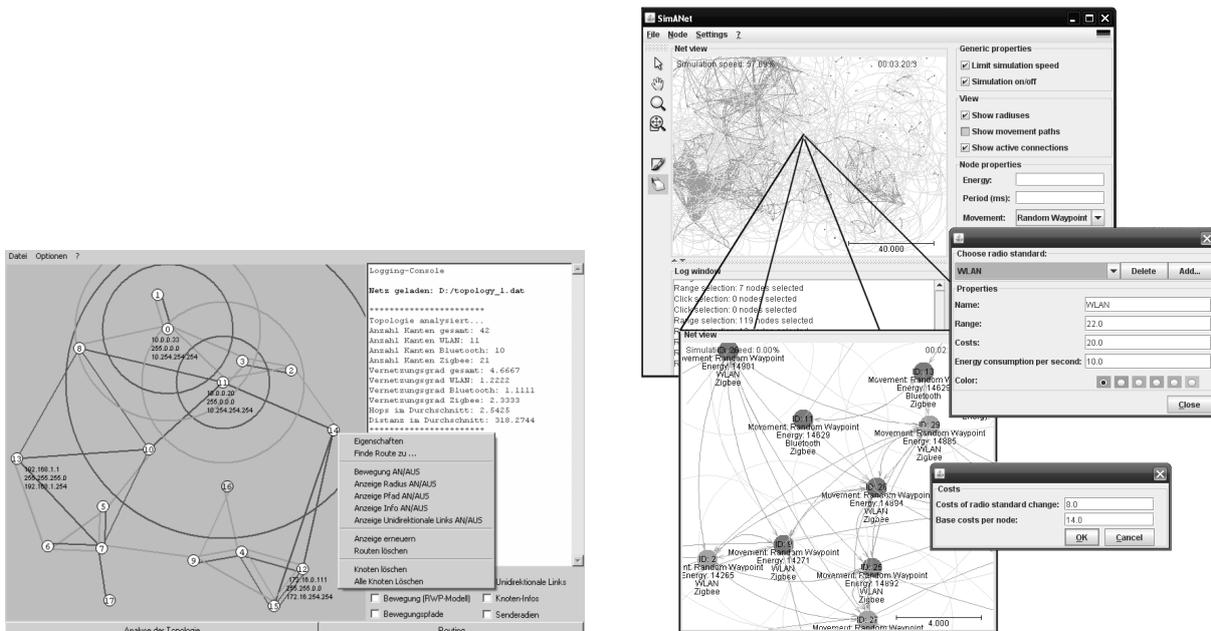
Grundlage dieses adaptiven Partitionierungsverfahrens ist die Aufteilung der Simulationsfläche auf $n \times m$ ($n, m \in \mathbb{N}$) Recheneinheiten. Bei der anschließenden Restrukturierung der geometrischen Fläche müssen mehrere Rahmenbedingungen beachtet werden. Zum einen sind Abweichungen der Flächen von der Rechteckform zu vermeiden, da in Folge dessen der administrative und kommunikative Aufwand erheblich ansteigt. Weiterhin darf die Reallokation nicht zu stark frequentiert stattfinden, da sonst zu viel Rechenzeit für die Optimierung aufgewendet werden

muss. Als initiale Partition wird eine gleichmäßige Verteilung gewählt. Zunächst prüft der Algorithmus dann zeilenweise jeden Sektor hinsichtlich seiner Auslastung und passt die vertikalen Sektorgrenzen bei Bedarf an.

Im Anschluss werden dann die kompletten Zeilen aus Sektoren miteinander verglichen und entsprechend ausbalanciert. Die horizontalen Sektorgrenzen werden dabei nicht individuell verschoben sondern einheitlich über die jeweilige Sektorzeile gesetzt. Anderenfalls würden an dieser Stelle komplexere Flächen entstehen, welche es zu vermeiden gilt. Abbildung 5.14 verdeutlicht das Verfahren anhand einer beispielhaften Netzwerktopologie, die sich auf sechs Berechnungseinheiten verteilt.

Der Ablauf des Partitionierungsverfahrens wird in Abbildung 5.15 als Pseudocode erläutert.

5.3. Funktionsumfang und Implementierung



(a) SimANet Verion 1 auf Basis der Skriptsprache Python.

(b) SimANet Version 2 als komplette Neuimplementierung basierend auf Java 6.

Abbildung 5.16.: Grafische Oberfläche von SimANet Version 1 und 2 im direkten Vergleich.

Die Simulationsumgebung SimANet entstand in der ersten Version auf Basis von Python. Dabei repräsentierte jeder simulierte Knoten einen eigenständigen Thread mit eigenen Kommunikationsroutinen. Folglich stieß die Simulation bei einer steigenden Knotenanzahl schnell an ihre Leistungsgrenzen, da der Synchronisationsaufwand unter den Thread einen zentralen Flaschenhals bildete. Möglichkeiten für eine verteilte Simulation waren ebenfalls nicht vorhanden. Zusätzlich erschwerten konzeptionellen Einschränkungen eine modulare Erweiterung mit den neuen Funktionalitäten und Simulationsmodellen. Als Konsequenz entstand eine vollkom-

5. Simulationsumgebung SimANet

men neu implementierte Version von SimANet, welche diese Nachteile umgeht und eine leistungsfähige Plattform für die Analyse neuer Kommunikationskonzepte im Bereich funkstandardübergreifender, mobiler Ad Hoc Netzwerke zur Verfügung stellt.

Um die Leistungsdefizite der Skriptsprache Python zu umgehen und gleichzeitig weiterhin eine plattformunabhängige Software zur Verfügung zu stellen, wurde SimANet - Version 2 auf Basis von *Java 6* entwickelt [17]. Für das grafische Frontend kam der Grafiktoolkit *Swing* zum Einsatz. Somit ist SimANet sowohl unter Windows als auch unter Linux Betriebssystemen problemlos einsetzbar. Dazu gehört auch der Hochleistungsrechencluster *CHiC*¹² der TU Chemnitz, der im Rahmen dieser Arbeit als Experimentalhardware für den parallelen Betrieb der Simulationsumgebung diente. Abbildung 5.16 zeigt SimANet in der ursprünglichen und aktuellen Version.

5.3.1. Architektur

Bei der Umsetzung von SimANet erfolgte eine konsequente Trennung des funktionalen Simulationskerns (*Backend*) und der grafischen Nutzeroberfläche (*Frontend*). Dabei können unterschiedliche *Entwurfsmuster* aus dem Bereich der Softwaretechnologie [68, 141] zum Einsatz kommen, welche komplexe Softwareanwendungen und die damit verbundenen Entwicklungsprozesse beherrschbar machen sollen und eine spätere Weiterentwicklung sowie Modifikationen der Software erleichtern. Die Implementierung des SimANet Frameworks erfolgte auf der Basis des *Model-View-Controller (MVC) Design Patterns* [121], bei dem zwischen Anwendungskern mit Steuerung (Controller), genutzten Datenstrukturen (Model) und grafischer Oberfläche/*Frontend* (View) unterschieden wird. Abbildung 5.17 verdeutlicht die Struktur des MVC Entwurfsmusters.

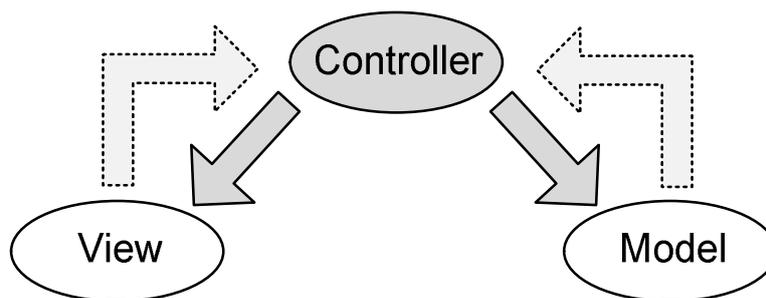


Abbildung 5.17.: Schematische Darstellung des *MVC Design Patterns* und der logischen Kommunikationspfade.

Neben Vorteilen für die Projektübersicht und eine strukturierte Weiterentwicklung bringt das Konzept aber auch einige Nachteile mit sich. Mit dem Ziel, die Software und somit auch den Entwicklungsprozess semantisch zu gliedern, müssen Einschränkungen hinsichtlich der Effizienz in Kauf genommen werden. Sämtliche Datenflüsse laufen über den Controller ab, was

¹²Chemnitzer Hochleistungs-Linux-Cluster

sich durch die verlängerten Kommunikationspfade negativ auf die Leistungsfähigkeit des Simulationskerns und folglich der gesamten Simulationsumgebung auswirkt. Dies bezieht sich konkret auf Operationen für die Datenkonvertierung, Dereferenzierungen sowie die Verwaltung zusätzlicher Stackframes.

Aus diesem Grund wurde bei der Umsetzung von SimANet an einigen zentralen Stellen das MVC Design-Pattern umgangen, um die dargelegten Flaschenhalse des Simulationskerns zu eliminieren. Diese umfassen im Wesentlichen den Zugriff der Zeichenroutine (innerhalb der View-Komponente) auf die darzustellende Datenstruktur. Hier darf lesend direkt auf die Model-Komponente zugegriffen werden, ohne den aufwendigen Umweg über den Controller.

5.3.2. Simulationskern

Im Abschnitt 5.2 wurde bereits näher auf eventbasierte Simulationsmodelle eingegangen. Für den implementierten Simulationskern kommt eine generische Prioritätswarteschlange zum Einsatz, welche die unterschiedlichen Events mit den dazugehörigen Parametern abarbeitet. Um die Verwaltung aller Events effizient zu gestalten, wurde eine *heap-basierte* Datenstruktur realisiert, welcher alle atomaren Operationen in einer Laufzeit von $O(\log n)$ ermöglicht. Die Prioritätswarteschlange beinhaltet die Menge aller möglichen Simulationsereignisse, welche sich wie folgt klassifizieren lassen:

- *Knoten-Kommandos* (einfügen, bewegen, löschen, modifizieren, etc.)
- *System-Kommandos* (starten der Simulation, unterbrechen, globale Ereignisse, Parallelisierung, etc.)
- *Export-Kommandos* (speichern der aktuellen Datenstrukturen / des Simulationszustandes, Screenshots, etc.)
- *Umgebungs-Kommandos* (Erzeugen von Hindernissen, Modifikation und Bewegung von Hindernissen, Eigenschaften des Übertragungsmediums, etc.)

Neue Ereignisse für den Simulationsprozess können zu jedem Zeitpunkt via GUI oder auch konsolengesteuert in ein bestehendes Szenario eingefügt werden. Ein automatisierter Testablauf ist ohne weiteres möglich und wird in Abschnitt 5.3.4 nochmals beschrieben.

Für die Ausführung von Simulationen beschränkt sich SimANet auf zwei wesentliche Threads für die GUI sowie für den Simulationskern. Folglich reduziert sich auch die Anzahl notwendiger Synchronisationskonstrukte, welche die Abarbeitungsgeschwindigkeit der Simulationsumgebung deutlich beeinflussen. *Kritische Abschnitte*, wie sie bei der Aktualisierung der GUI mit den aktuellen Datenstrukturen auftreten, können auf diese Weise minimiert werden. Synchronisationsprobleme durch das parallele Arbeiten einer Vielzahl unabhängiger Simulationsthreads werden in Folge dessen weitestgehend eliminiert.

5. Simulationsumgebung SimANet

Für die Nachbarschaftsverwaltung kommt die bereits vorgestellte Skip Quadtree Datenstruktur zum Einsatz. Sie ermöglicht eine effiziente Nachbarschaftssuche in logarithmischer Laufzeit und leistet folglich einen wesentlichen Beitrag zur Leistungsfähigkeit von SimANet.

5.3.3. Skalierbarkeit und Parallelisierung

Um die Leistungsfähigkeit zu erhöhen, kamen die in Abschnitt 5.2.6 eingeführten Verfahren zum Einsatz. Die automatischen Partitionierungsverfahren zur dynamischen Lastverteilung im parallelen Simulationsumfeld sowie die Nutzung des *Sliding Time Window* Algorithmus ermöglichen eine hervorragende Skalierbarkeit von SimANet. Im derzeitigen Entwicklungsstand ist dabei der erlaubte Time-Drift t_d statisch definiert und kann somit nicht zur Laufzeit angepasst werden. Dies bietet Raum für weitere Optimierungen, welche im weiteren Verlauf des Kapitels unter 5.5 näher erläutert werden.

Im Zuge der Parallelisierung von SimANet kam MPI zum Einsatz. Der MPI Standard [119, 120] beschreibt dabei den Nachrichtenaustausch innerhalb paralleler Berechnungen in verteilten Systemen. Die Programmierschnittstelle besteht aus hochoptimierten Bibliotheken, die für nahezu alle Computerarchitekturen, Betriebssysteme und Rechencluster zur Verfügung stehen. Bedingt durch den hohen Optimierungsgrad sind MPI Bibliotheken, sowohl in der Spezifikation 1.0 als auch in der aktuellen Version 2.0, ausschließlich in C programmiert. Um diese Funktionalität in die java-basierende SimANet Implementierung zu integrieren, kann entweder eine reine Java-Implementierung oder die Abstraktion mittels *JNI*¹³ genutzt werden (Abbildung 5.18). Mit JNI bietet Java eine standardisierte Programmierschnittstelle (API) für den Zugriff auf plattformspezifische Funktionen, wodurch auch andere Programmiersprachen und Bibliotheken verwendet werden können. Allerdings ist durch diese Flexibilität eine Plattformunabhängigkeit nur noch bedingt gewährleistet. Der Konfigurationsaufwand ist entsprechend aufwendiger, da die zusätzlichen, hardware-spezifischen Softwarebibliotheken für die Ausführung der Anwendung zur Verfügung stehen müssen.

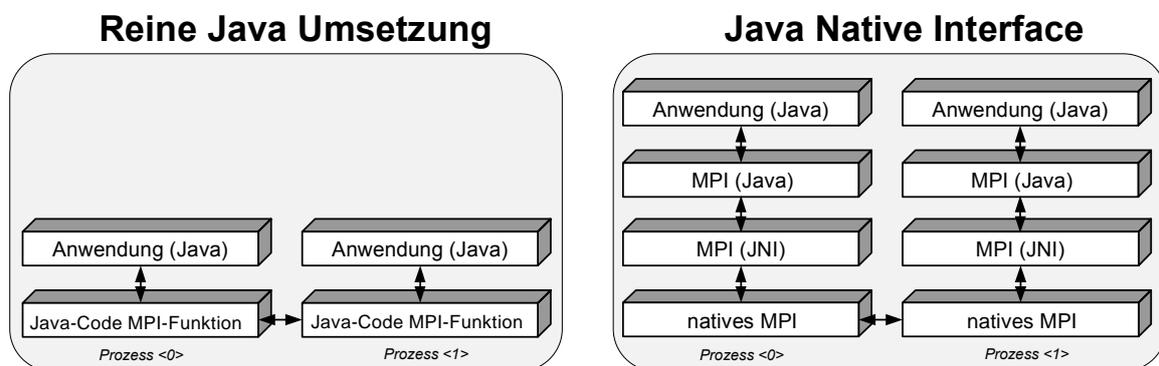


Abbildung 5.18.: Unterschiedliche Ansätze für MPI Umsetzungen in Java. Links eine reine Java Umsetzung, welche auf direktem Weg native MPI Funktionalitäten nutzt und rechts eine Realisierung über JNI.

¹³Java Native Interface

Das im Rahmen der hier durchgeführten Simulationen genutzte *jMPI* [118] ist eine reine Java-Implementierung, welche zwar plattformunabhängig einsetzbar ist, aber seit 2003 nicht mehr weiterentwickelt wurde. In Folge dessen wurden Erweiterungen der MPI 2.0 Spezifikation nicht integriert. Als reine Java-Umsetzung wurden alle MPI-Funktionen auf Basis von Java nachprogrammiert. Da Java vor allem durch sein Speicherkonzept und durch die Hardwareabstraktion auf virtuelle Maschinen deutlich weniger Optimierungsmöglichkeiten bereitstellt als die native Programmierung in C, ergeben sich an dieser Stelle erhebliche Leistungsdefizite. Die geschriebenen Java-Funktionen sind in Hinsicht auf die Leistungsfähigkeit nicht mit den hochoptimierten C-Bibliotheken vergleichbar und erzeugen weitaus mehr Overhead. Desweiteren bewirkt auch die TCP-basierte Kommunikation einen Anstieg der Kommunikationszeit gegenüber einer verbindungslosen UDP Variante.

Sinnvoll ist daher die Implementierung einer MPI-Umgebung für Java, welche die nativen C-Bibliotheken von MPI nutzt. Hier wäre *mpiJava* [28] als eine auf JNI basierte Implementierung deutlich leistungsfähiger. Jedoch wurde auch *mpiJava* seit 2003¹⁴ nicht mehr weiterentwickelt und die Kompatibilität zu den verfügbaren Rechnerarchitekturen ist nur noch eingeschränkt gewährleistet. Es existiert keine 64Bit Unterstützung, wodurch ein Einsatz auf aktuellen Server- und Rechencluster-Systemen nur teilweise möglich ist. Um den Funktionsumfang des aktuellen MPI 2.0 Standards integrieren zu können, ist auch hier eine umfassende Anpassung bzw. Erweiterung der verwendeten Wrapper notwendig.

Auf das Optimierungspotential durch eine aktuelle und leistungsfähige Java-Anbindung an MPI wird in Abschnitt 5.5 detailliert eingegangen. Grundsätzlich kann davon ausgegangen werden, dass nur hinreichend große Simulationsszenarien effizient parallelisierbar sind.

5.3.4. Ein- und Ausgabe

Ein wesentliches Bewertungskriterium für den Einsatz von Simulationswerkzeugen ist die Bedienbarkeit und die möglichen Funktionalitäten für die Datenein- und -ausgabe. Auf die zentralen Merkmale von SimANet wird nun näher eingegangen.

Benutzeroberfläche

Standardmäßig nutzt SimANet eine grafische Nutzeroberfläche zur Visualisierung der Simulation. Dabei erfolgt die Ausgabe der Datenstruktur in einem dedizierten Thread, wodurch die Bedienbarkeit auch bei großen Datenmengen und komplexen Netztopologien gewährleistet werden kann. An dieser Stelle wurde zu Gunsten der Leistungsfähigkeit auch das MVC Entwurfsmuster umgangen. Um die relativ hohe Rechenzeit für die Darstellungsprozedur zu optimieren, greift diese direkt auf die Datenstruktur zu und verzichtet auf explizite Synchronisationsmechanismen. Das Aktualisierungsintervall beträgt statisch 100 ms.

Die dargestellten Knoten werden je nach aktuellem Zoom-Grad mit entsprechenden Zusatzin-

¹⁴nach [28], Stand Januar 2009

5. Simulationsumgebung SimANet

formationen (ID, Routing-Informationen, Ladezustand, etc.) dargestellt. Aktive Verbindungen zwischen Knoten wurden als Bezier-Kurven realisiert, wodurch mehrere Verbindungskanten zwischen zwei Knoten auf Basis unterschiedlicher Kommunikationsstandards differenziert werden können. Hindernisse können jederzeit eingefügt oder entfernt werden.

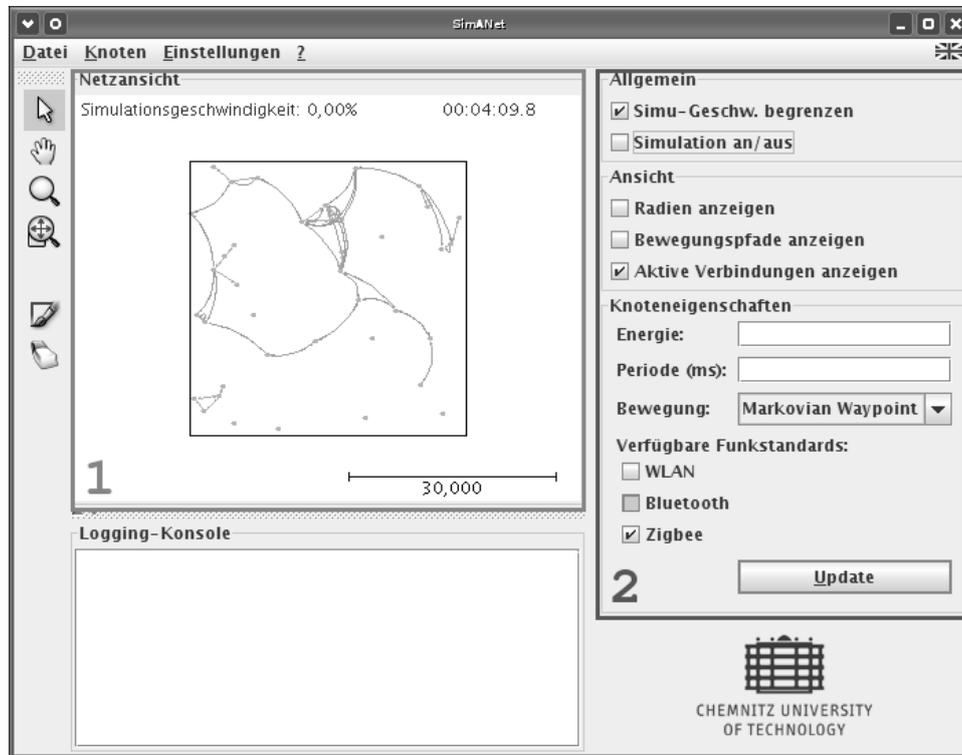


Abbildung 5.19.: Funktionale Aufteilung der grafischen Benutzeroberfläche von SimANet. Diese untergliedert sich in zwei Bereiche: die grafische Darstellung des Simulationsszenarios (1) sowie die Eigenschaftsleiste mit globalen Einstellmöglichkeiten für die Simulationsumgebung (2).

Im rechten Bereich der Benutzeroberfläche befinden sich die Konfigurationsmöglichkeiten der Simulationsumgebung. Diese umfassen Möglichkeiten zur Erstellung und Modifikation verfügbarer Funkstandards, Optionen für die zu nutzenden Bewegungsmodelle sowie Routing- und Energieeinstellungen.

SimANet ist mehrsprachig ausgelegt und kann durch die Erstellung weiterer Sprachdateien je nach Bedarf erweitert werden. Derzeit stehen die Sprachen Deutsch und Englisch zur Verfügung.

Testautomatisierung

SimACon - die *Consolen* Steuerung von SimANet - stellt eine eigenständige Skriptsprache dar, welche die Automatisierung von Simulationsszenarien sowohl im grafischen als auch im textbasierten Modus ermöglicht. Einzelne Zeilen in den SimACon-Skripten repräsentieren Ereignisse in der simulierten Zeit. Beim Laden werden diese in die *Event-Queue* eingefügt und somit zum definierten Zeitpunkt ausgeführt. Der Funktionsumfang von SimACon umfasst alle Arten von

Ereignissen für Knoten und Simulationsumgebung. Somit sind sämtliche internen und externen Funktionsaufrufe von SimANet verfügbar.

Die SimACon Skriptsprache wird in Anhang A genauer vorgestellt und die Syntax anhand eines Beispiels erläutert.

Datenaufbereitung

Für den Datenimport und Export wurde ein *XML*¹⁵ basiertes Format gewählt. Somit ist das Speichern der gesamten Netzwerktopologie inklusive der gewählten Konfigurationsdaten auf einfache Weise möglich. Die Speicherabbilder können durch den Einsatz von XML auch nachträglich noch analysiert und ggf. modifiziert werden.

Auch der Export des Anzeigebereichs als Rastergrafik-Datei ist möglich. Die Auflösung wird dabei durch den Nutzer definiert. Eine Konvertierung in die gewünschten Ausgabeformate realisieren java-eigene Bibliotheken. Standardmäßig stehen für den Grafik-Export die Formate *JPG*, *PNG*, *BMP* und *GIF* zur Verfügung.

Im konsolenbasierten Betrieb sind die vorgestellten Export-Funktionalitäten die zentralen Mittel zur Ausgabe der jeweiligen Simulationsabschnitte. Mit Hilfe der bereits vorgestellten SimACon Skriptsprache kann ein automatisierter Export ereignisgesteuert oder auch periodisch erfolgen.

5.4. Einschränkungen

SimANet fokussiert die Validierung neuer Kommunikationskonzepte im Bereich funktionsübergreifender, mobiler Ad Hoc Netzwerke auf einer abstrakten Ebene. Technologiespezifische Rahmenbedingungen einzelner Kommunikationsstandards werden in ein abstraktes Kostenmodell überführt, welches jede Form von Kommunikationsereignissen auf definierte Kostenwerte abbildet. Die Modellierung physikalisch korrekter und detaillierter Umweltbedingungen ist konzeptionell vorgesehen, jedoch im aktuellen Entwicklungsstand nicht vollständig realisiert.

Das Interferenzverhalten auf dem PHY-Layer wird im aktuellen Entwicklungsstand des Simulators vernachlässigt. Hier ist derzeit kein korrektes physikalisches Modell hinterlegt, wodurch Störungen durch Wellenüberlagerungen nicht simuliert oder detektiert werden können. Die Problematik beinhaltet folglich auch die physikalisch korrekte Berechnung von Dämpfung, Beugung und Reflexion sich ausbreitender Funkwellen im Raum. Erstellte Hindernisse können zwar mit abstrakten Eigenschaften für die Signaldämpfung definiert werden, diese beruhen jedoch nur auf einer individuell festgelegten Wahrscheinlichkeit für den Paketverlust oder alternativ auf einer definierten Reduzierung der Sendereichweite. Das Verhalten basiert dabei nicht auf einer physikalisch korrekten Berechnung der Eigenschaften des zu durchdringenden Mediums.

¹⁵Extendable Markup Language

5. Simulationsumgebung SimANet

Für die Übertragung mit SimANet wird ein störungsfreier Kommunikationskanal auf MAC Ebene angenommen. Technologiespezifische MAC-Layer Modelle sind in der derzeitigen Simulationsumgebung nicht enthalten. Die Übertragungscharakteristik eines abstrakt definierten Funkstandards wird auf eine Menge ausgewählter Kommunikationsparameter abgebildet. Diese beinhalten nur indirekt das Verhalten auf MAC Ebene.

Weiterhin stellt SimANet derzeit ausschließlich eine zweidimensionale Umgebung zur Verfügung. SimANet wurde aber bereits für eine Erweiterung um eine dritte Dimension vorbereitet. Alle verwendeten Datenstrukturen wurden generisch definiert so eine korrekte Funktion im 3D-Betrieb. Dabei muss zum einen die Zeichenroutine entsprechend angepasst werden, um die höheren Dimensionskoordinaten zu visualisieren. Die Implementierung einer einfachen, isometrischen 3D-Darstellung ist aber problemlos möglich. Die Umsetzung der Bereichssuche als zentrale Operationen auf der Skip Quadtree Datenstruktur gestaltet sich zeitintensiver. Während sich die mathematische Problematik im zweidimensionalen Raum auf die Berechnung einer Schnittfläche zwischen Quadrat und Kreis reduziert, entsteht im 3D-Raum die Durchdringung von Kugel und Würfel. Der Aufwand für die Berechnungen steigt folglich erheblich an und steht in keiner Relation zu der Problematik in zwei Dimensionen.

Im Rahmen paralleler Simulationen ist ein Betrieb der grafischen Nutzeroberfläche nicht möglich. Durch die fehlende Interaktion reduziert sich die Funktionalität von SimANet auf den konsolenbasierten Modus. Dazu kommt die bereits vorgestellte SimACon Skriptsprache für die Testautomatisierung zum Einsatz.

5.5. Optimierungsansätze

Mögliche Optimierungsansätze resultieren direkt aus den zuvor genannten Einschränkungen. Auf Einige soll nun aber nochmals eingegangen werden.

5.5.1. Übertragungsmodell

Einen primären Verbesserungspunkt stellt die derzeit umgesetzte Modellierung der Datenübertragung dar. Eigenschaften der Antennensysteme können nur durch abstrakte Werte für die Sendereichweite dargestellt werden. Gerichtete Antennensysteme, wie in Kapitel 3.2.3 (Smart Antennas) beschrieben, sind nicht realisierbar.

Weiterführend ist somit die physikalisch korrekte Modellierung des Übertragungsmediums sowie die Einbindung exakter Modelle der jeweiligen Kommunikationsstandards sinnvoll. Das Zeitverhalten zur Laufzeit wird im aktuellen Stand auf die abstrakte Kostenfunktion abgebildet und ist somit implizit auf algorithmischer Ebene enthalten. Es wurde somit kein separates, datenvolumenbasiertes Zeitmodell auf Netzwerkebene gewählt. Für den Fokus der wissenschaftlichen Betrachtungen in dieser Arbeit spielt diese Einschränkung keine Rolle. Für die Weiterverwendung des Simulators muss dieses Modell ggf. angepasst und erweitert werden.

5.5.2. Topologieoptimierung / Routing

Der Fokus von SimANet richtet sich auf die Verwaltung mehrerer Funkmodule pro Knoten und der damit verbundenen Steigerung der Erreichbarkeit. Somit spielen angepasste Routing-Verfahren eine zentrale Rolle. Um dabei den Energiebedarf zu optimieren, sind unterschiedliche Ansätze aus dem Bereich der Topologiekontrolle / Topologieoptimierung vorstellbar. Erweiterungen von SimANet mit entsprechenden Modulen zur dynamischen Anpassung der Sendeleistung oder zur selektiven Vernetzung sind in jedem Fall sinnvoll. Zusätzlich können in Verbindung mit physikalisch korrekten Übertragungsmodellen weiterführende Verfahren zur Detektierung und Vermeidung von Interferenzen betrachtet werden.

5.5.3. Sliding Time Window

Im aktuellen Entwicklungsstand wird der mögliche Zeitversatz t_d zwischen den jeweiligen Berechnungseinheiten statisch definiert. Der optimale Wert für diesen Zeitdrift ist jedoch stark von der Größe des Simulationsszenarios beeinflusst. Auch andere Faktoren, wie bspw. die Latenzen in der Netzinfrastruktur spielen eine Rolle. Andere Prozesse des Betriebssystems, welche um nötige Betriebsmittel konkurrieren, sowie die betriebssystem-internen Scheduling-Algorithmen sind ebenfalls relevant.

Folglich muss t_d je nach parallelem Berechnungsumfeld manuell angepasst und optimiert werden. Jedoch wäre an dieser Stelle eine dynamische Anpassung zur Laufzeit in Abhängigkeit von ausgewählten Faktoren der Hard- und Software-Umgebung sinnvoll.

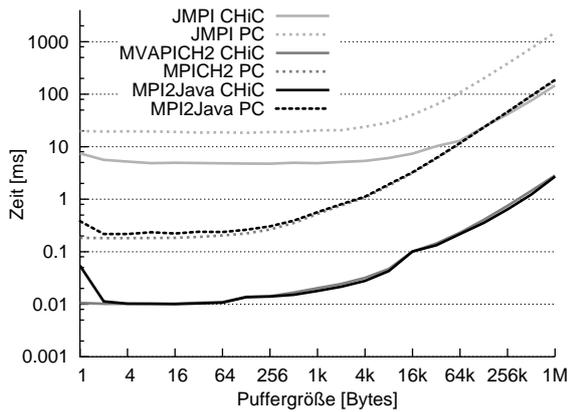
Neben der aktuellen Rechenlast könnten unter anderem die benötigten Unterbrechungszeiten für die Aktualisierung des Zeitfensters als Referenzwert dienen. Überschreitet die Häufigkeit der Unterbrechungen einen definierten Wert, wird t_d erhöht. Detektiert der Sliding Time Window Algorithmus über einen längeren Zeitraum keine Unterbrechung, wird im Gegenzug der Wert für t_d entsprechend verringert.

5.5.4. Parallelisierung

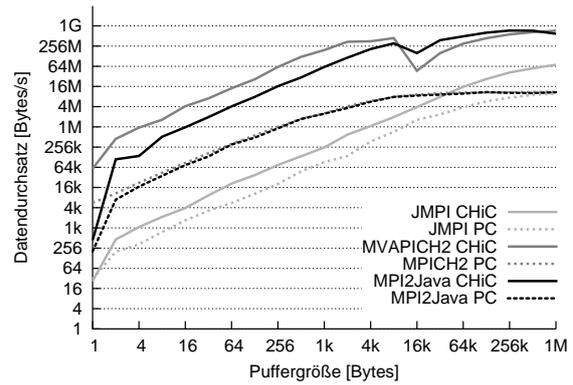
Im Rahmen der Diplomarbeit *Java-Anbindung von MPI-Bibliotheken für hochperformante Parallelanwendungen* [18] wurde eine umfassende Neuimplementierung der MPI-Schnittstelle für Java durchgeführt. Im Gegensatz zur bisher genutzten, reinen Java-Implementierung JMPI basiert das entwickelte *MPI2Java* auf JNI und nutzt die hochoptimierten C-Bibliotheken der aktuellen MPI 2.0 Spezifikation. *MPI2Java* ist vollständig 64Bit-fähig und weiterhin plattformunabhängig nutzbar.

Durch die verzögerte Entwicklung von *MPI2Java* kommt die neue Schnittstelle in den durchgeführten Simulationsszenarios jedoch nicht zum Einsatz. Dennoch seien an dieser Stelle einige Leistungsmessungen aufgeführt, welche das zusätzliche Optimierungspotential von SimANet durch die neue Schnittstelle verdeutlichen.

5. Simulationsumgebung SimANet

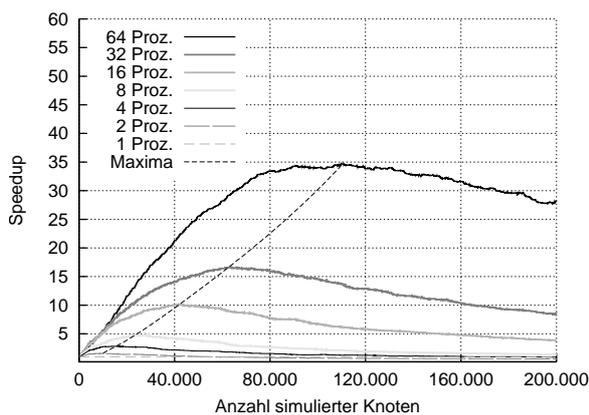


(a) Testreihen MPI2Java - gemessene Zugriffszeiten mittels Pingpong-Test, mit dessen Hilfe die Latenzen während der Datenübertragung analysiert wurden.

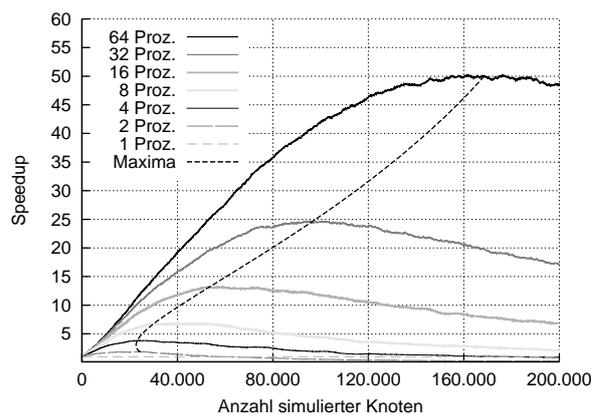


(b) Testreihen MPI2Java - Datendurchsatzmessungen.

Abbildung 5.20.: Optimierungspotential durch MPI2Java. Messungen jeweils auf dem CHiC-Rechencluster der TU Chemnitz. Da mpiJava nicht 64Bit-fähig ist, konnten entsprechend keine Werte aufgenommen werden. Rahmenbedingungen: exklusive Nutzung der beteiligten CPU-Knoten, Darstellungen entsprechen Mittelwerten aus 10 unabhängigen Durchläufen.



(a) JMPI - Dynamische Topologie mit einer unterschiedlicher Anzahl Prozessoren.



(b) MPI2Java - Dynamische Topologie mit einer unterschiedlicher Anzahl Prozessoren.

Abbildung 5.21.: Direkter Vergleich der bisher genutzten, veralteten JMPI Umsetzung und MPI2Java, welches nun MPI2.0 unterstützt und auf JNI basiert.

Erste Messreihen der MPI2Java Umsetzung innerhalb der SimANet-Umgebung zeigen die Leistungssteigerungen durch eine effiziente Parallelisierung. Abbildung 5.20 und 5.21 verdeutlicht die enormen Unterschiede der reinen Java-Implementierung JMPI und der auf JNI basierten MPI2Java Umsetzung. Die erreichten Speedup-Werte zeigen das Optimierungspotential durch die MPI2.0 Funktionserweiterung und die JNI-Anbindung.

Ein Ansatzpunkt für Verbesserungen stellt der Datenexport im parallelen Umfeld dar, der unter Umständen einen Flaschenhals repräsentiert. Die periodische Übermittlung aller Informationen an eine definierte Datensinke führt unweigerlich zu Blockierungen während der Synchronisationszeitpunkte. Diese Problemstellung kann auch nicht ohne Weiteres beseitigen werden und

erfordert und Umständen ein erhöhten Implementierungsaufwand.

5.5.5. Partitionierung

Die dynamische Partitionierung im parallelen Simulationsumfeld erfolgte bisher nur in Abhängigkeit von Knotenmengen auf den einzelnen Berechnungseinheiten. Wurden definierte Referenzwerte über- oder unterschritten, erfolgte die Reallokation der betroffenen Sektorgrenzen. Eine Bewertung auf Basis des Kommunikationsverhaltens würde hier eine deutlich bessere Bewertungsmethode darstellen und die Leistungsfähigkeit nochmals erheblich verbessern.

5.6. Simulation und Analyse

Auf Basis der vorgestellten Simulationsumgebung SimANet können nun die in Kapitel 4 vorgestellten Konzepte simuliert und entsprechend bewertet werden. Die durchgeführten Messungen und die dabei gewählten Rahmenbedingungen werden in den jeweiligen Abschnitten beschrieben. Neben Testreihen in Bezug auf die Leistungsfähigkeit von SimANet im Vergleich zu anderen bekannten Simulationswerkzeugen stehen im Wesentlichen die Faktoren *Erreichbarkeit*, *Energiebedarf* und *Routing* im Fokus der Betrachtungen. Die Ergebnisse ermöglichen qualitative Aussagen über das vorgestellte Konzept und stützen die Thesen dieser Arbeit.

5.6.1. Leistungsfähigkeit des Simulationskern

Um die Leistungsfähigkeit von SimANet selbst zu ermitteln, wurden zunächst mehrere Testreihen bezüglich des Simulationskerns und des Ressourcenverbrauchs durchgeführt. Dabei ist auch die Effizienz in einem parallelen Berechnungsumfeld von Relevanz.

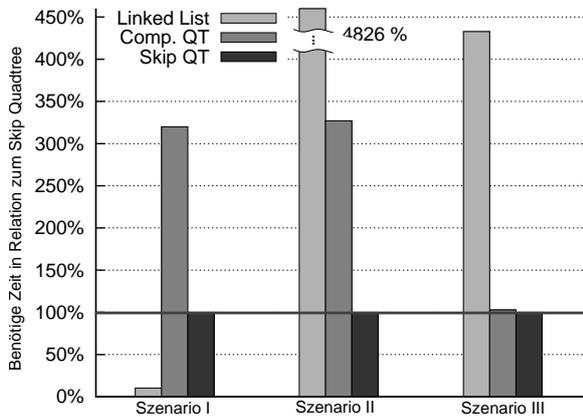
Datenstruktur und Objektlokalisierung

Der Skip Quadtree stellt eine zentrale Datenstruktur in SimANet dar und steht folglich in direktem Zusammenhang mit der Leistungsfähigkeit. Abbildung 5.22 zeigt dazu zwei Diagramme.

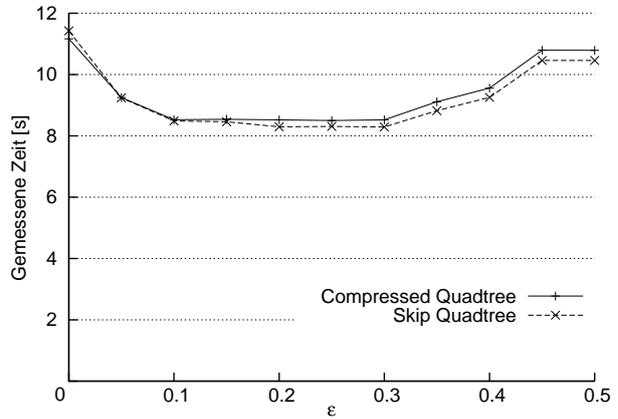
In der ersten Testreihe wurde die benötigte Zeit für das Durchführen von 300.000 zufälligen Such-Operationen gemessen. Der verwendeten Skip Quadtree Datenstruktur sind dabei der herkömmliche Compressed Quadtree und eine triviale verkettete Liste gegenübergestellt. Durchgeführt wurden drei Szenarien: *Szenario 1* repräsentiert einen künstlich konstruierten *Best Case* für den Skip Quadtree. Die erzeugte Knotenlandschaft enthält 60 Knoten, bei denen sich der Abstand zwischen zwei Punkten P_i und P_{i+1} jeweils verdoppelt. In Folge dessen entsteht für den Compressed Quadtree ein *Worst Case*, da eine Baumtiefe von n und somit ein Laufzeitverhalten von $O(n)$ ($n = \#Knoten$) provoziert wird. Da die Anzahl Knoten in diesem Fall sehr gering ist, benötigt hier die einfache Liste am wenigsten Zeit und schlägt beide

5. Simulationsumgebung SimANet

Baumimplementierungen. *Szenario II* repräsentiert eine Kombination aus *Szenario I* und einer zusätzlichen, zufälligen Verteilung von 5.000 Knoten. Erwartungsgemäß bleibt der Abstand zwischen *Compressed Quadtree* und *Skip Quadtree* auf gleichem Niveau. Durch die deutlich größere Netztopologie werden nun aber die Nachteile der Listenimplementierung sehr deutlich. In *Szenario III* wurde eine sehr realitätsnahe *Hop Spot* Knotenverteilung gewählt, in dem die Knotendichte in verschiedenen Bereichen stark variiert.



(a) Vergleich drei verschiedener Datenstrukturen für die Objektlokalisierung: einfache Liste mit X/Y-Koordinaten, *Compressed Quadtree* und *Skip Quadtree*. Die Messergebnisse des *Skip Quadtree* stellen dabei den Referenzwert dar. Die anderen Messergebnisse werden relativ zu diesem Referenzwert interpretiert.



(b) Vergleich von *Compressed Quadtree* und *Skip Quadtree* in Abhängigkeit des gewählten ϵ . Erkennbar ist das erreichte Optimum bei $0.2 \leq \epsilon \leq 0.3$. Die Laufzeit des *Skip Quadtrees* liegt ab einem $\epsilon \geq 0.1$ kontinuierlich unter der des normalen *Compressed Quadtrees*.

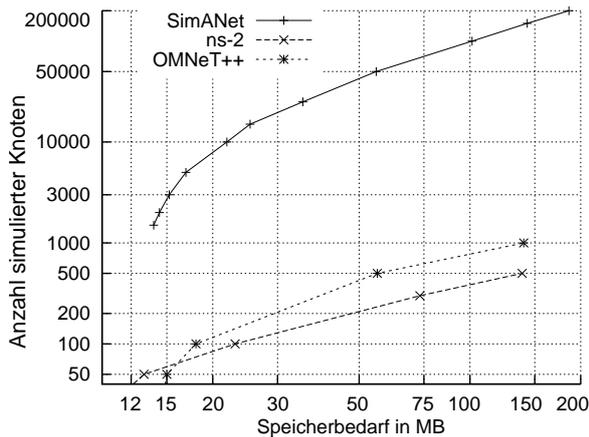
Abbildung 5.22.: Leistungsfähigkeit verschiedener Datenstrukturen für die Objektlokalisierung.

Im rechten Diagramm wurden Messungen für die Bereichssuche dargestellt. Wie bereits erwähnt, muss hier ein Kompromiss zwischen Genauigkeit und Antwortzeit gefunden werden. Um einen optimierten Wert für den Randbereich zu finden, wurden Bereichssuchen mit unterschiedlichen Ausdehnungen für den Rand (ϵ von 0.0 bis 0.5, Schrittweite 0.05) durchgeführt. Ziel war es, diesen Wert ϵ , also die Größe des Radius im Verhältnis zum Suchradius, dahingehend zu optimieren, dass eine minimale Antwortzeit erzielt wird. Die Testumgebung umfasst dabei 50.000 zufällig verteilte Knoten und gemessen wurde die Zeit für 1.000 zufällige Umkreisanfragen. Die durchschnittlich zurückgelieferte Knotenmenge umfasste 10.000 Knoten. Das Diagramm zeigt deutlich das ermittelte Optimum bei $\epsilon \approx 0.3$.

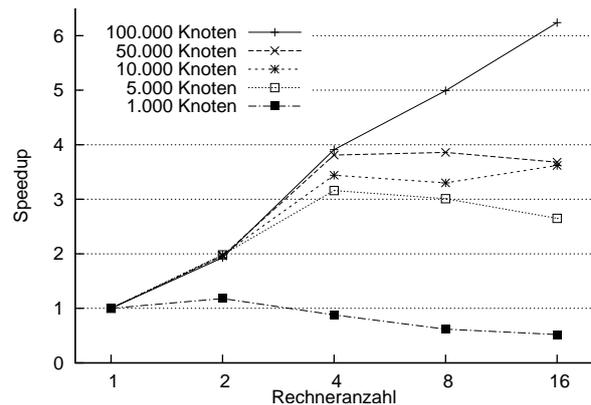
Ressourcenverbrauch und Parallelisierung

Im Folgenden wurden mehrere Testreihen für den Speicherverbrauch und die Leistungsfähigkeit bei verteilten Simulationen durchgeführt. Abbildung 5.23 zeigt links den Ressourcenbedarf von SimANet im Vergleich zu den bekannten Simulationswerkzeugen ns2 und OMNet++. Schon bei einer geringen Knotenmenge werden die Relationen deutlich. Wie bereits in Abschnitt 5.1 ausführlich erläutert, bieten weder ns2 noch OMNet++ Möglichkeiten für die Simulation großer

Netzwerktopologien mit einer akzeptablen Geschwindigkeit [123]. Bereits bei einem mittelgroßen Anwendungsszenario mit 500 Knoten benötigt OMNet++ über 50 MB Arbeitsspeicher. ns2 in der originalen Version belegt hierbei annähernd 150 MB RAM (nach [123] 138.7 MB). Trotz mehrerer Optimierungsansätze sind Szenarien mit über 3.000 Knoten nur sehr eingeschränkt realisierbar. Neben den Anforderungen an Arbeitsspeicher stellt dabei vor allem die extrem lange Simulationszeit einen limitierenden Faktor dar. Praktische Tests mit OMNet++ zeigen schon bei Topologien mit 200 Knoten Wartezeiten von weit über 30 Minuten, wodurch diese Simulationswerkzeuge bereits an ihre Leistungsgrenzen stoßen.



(a) Der *SimANet* Simulationskern im Vergleich mit ns2 und OMNet++. Der Speicherbedarf von SimANet beträgt dabei nur einen Bruchteil von den benötigten Ressourcen bei ns2 und OMNet++. Somit ist nur SimANet in der Lage, große Netzwerktopologien mit weit über 10^5 Knoten zu simulieren. Der Funktionsumfang kann dabei je nach Bedarf um die jeweiligen Berechnungsmodelle bzw. Plugins erweitert werden.



(b) Erzielter Speedup bei der Parallelisierung in Abhängigkeit von verschiedenen Problemgrößen. Wie deutlich zu erkennen ist, können nur hinreichend große Netzwerktopologien effizient parallelisiert werden. Bei Anwendungsszenarien mit 1.000 oder weniger Knoten wird die Simulation in einer verteilten Berechnung bedingt durch den Synchronisationsaufwand sogar ausgebremst.

Abbildung 5.23.: Direkter Vergleich von SimANet mit den bekannten Simulationswerkzeugen ns2 und OMNet++ in Hinblick auf die Skalierbarkeit (a). Rechts der erreichte Leistungszuwachs bei verteilten Simulationen (b).

Eine Parallelisierung kann die Leistungsfähigkeit nochmals verbessern. Neben den konkreten Anwendungsszenarien haben dabei vor allem der Simulationskern und die verwendeten Mechanismen für die verteilte Simulation einen entscheidenden Einfluss auf den erreichbaren Speedup. Die Probleme im Bereich der Netzwerksimulation sowie die Nachteile traditioneller, blockierender Parallelisierungsmechanismen wurden im Verlauf dieses Kapitels bereits verdeutlicht. Abbildung 5.23 zeigt rechts Leistungsmessungen des im Rahmen von SimANet implementierten Parallelisierungsverfahrens. Dieses umfasst neben dem Sliding Time Window Algorithmus auch die dynamische Aufteilung und Anpassung der Simulationsfläche auf die verfügbare Berechnungsumgebung.

Im Gegensatz zu blockierenden Verfahren erlaubt die Einführung eines definierten Zeitdrifts eine nahezu optimale Beschleunigung im parallelen Umfeld. Die Ergebnisse sind dabei wie erwartet, stark abhängig von der Problemgröße. Bei hinreichend großen Topologien mit 100.000

5. Simulationsumgebung SimANet

und mehr Knoten kommt das Verfahren in vollem Umfang zum Tragen. Bei kleineren Topologien sinkt der Speedup erheblich, primär bei vielen parallel arbeitenden Recheneinheiten. Ebenfalls vorhersehbar sind die Ergebnisse bei kleinen Netzwerktopologien. Für Knotenmengen ≤ 1.000 bremst die Synchronisation über das Netzwerk die Simulation sogar aus.

Die vorgestellten Algorithmen für die dynamische Grenzverschiebung und das Sliding Time Window im Rahmen paralleler Simulationen erzielen sehr gute Werte hinsichtlich des Speedups. Trotz des kommunikationslastigen Charakters von Netzwerksimulationen erlaubt das umgesetzte Konzept eine fast optimale Beschleunigung bei hinreichend großen Knotentopologien. Die Ergebnisse sind dabei nur auf Basis tolerierter Zeitdrifts möglich, welche die auftretenden Netzwerklatenzen überbrücken und ausbalancieren.

5.6.2. Testreihen EAN - Topologische Kenngrößen

In den folgenden Simulationsszenarien soll der konzeptionelle Vorteil hinsichtlich einer Steigerung der Erreichbarkeit untersucht werden. Im Bereich der Graphentheorie existiert hierzu eine analytische Betrachtung dieser Problemstellung. *Random Geometric Graphs (RGG)* [51] ermöglichen dabei unter bestimmten Rahmenbedingungen eine numerische Berechnung des notwendigen Vernetzungsgrades für eine vollständig erreichbare Topologie. Dabei wird jedoch davon ausgegangen, dass die gesamte Topologie bei maximaler Sendeleistung erreichbar ist. Der Lösungsansatz beschränkt somit die Erreichbarkeitsproblematik auf eine reine Topologieoptimierung. Eine vollständig erreichbare Topologie wird vorausgesetzt und es soll ausschließlich der minimale Vernetzungsgrad für eine vollständige Konnektivität ermittelt werden.

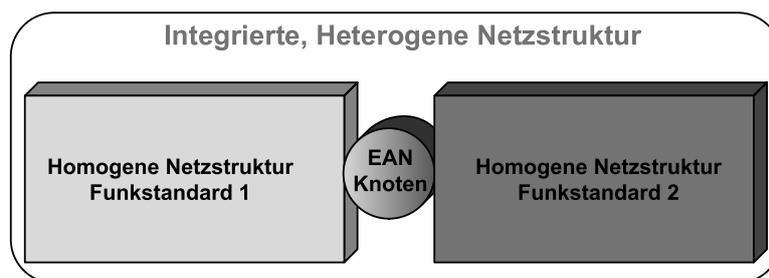


Abbildung 5.24.: Zwei Netzwerkstrukturen können durch einen EAN-konformen Knoten integriert werden.

Das EAN Konzept betrachtet im Gegensatz dazu auch disjunkte Netzstrukturen, welche aufgrund ungünstiger Knotenverteilungen oder mangelnder Sendeleistung isoliert sind. Hier soll nun mit Hilfe EAN-konformer Brücken-Knoten eine Integration ermöglicht werden. Dazu wird in den folgenden Simulationen sukzessiv der Anteil EAN-konformer Multistandard-Knoten erhöht. Durch das zweite Kommunikationsinterface mit anderen Übertragungseigenschaften erhöht sich der Vernetzungsgrad in den einzelnen Knoten. Ziel dabei ist eine vollständig erreichbare, funkstandardübergreifende Netzinfrastruktur bei möglichst geringem EAN Knotenanteil.

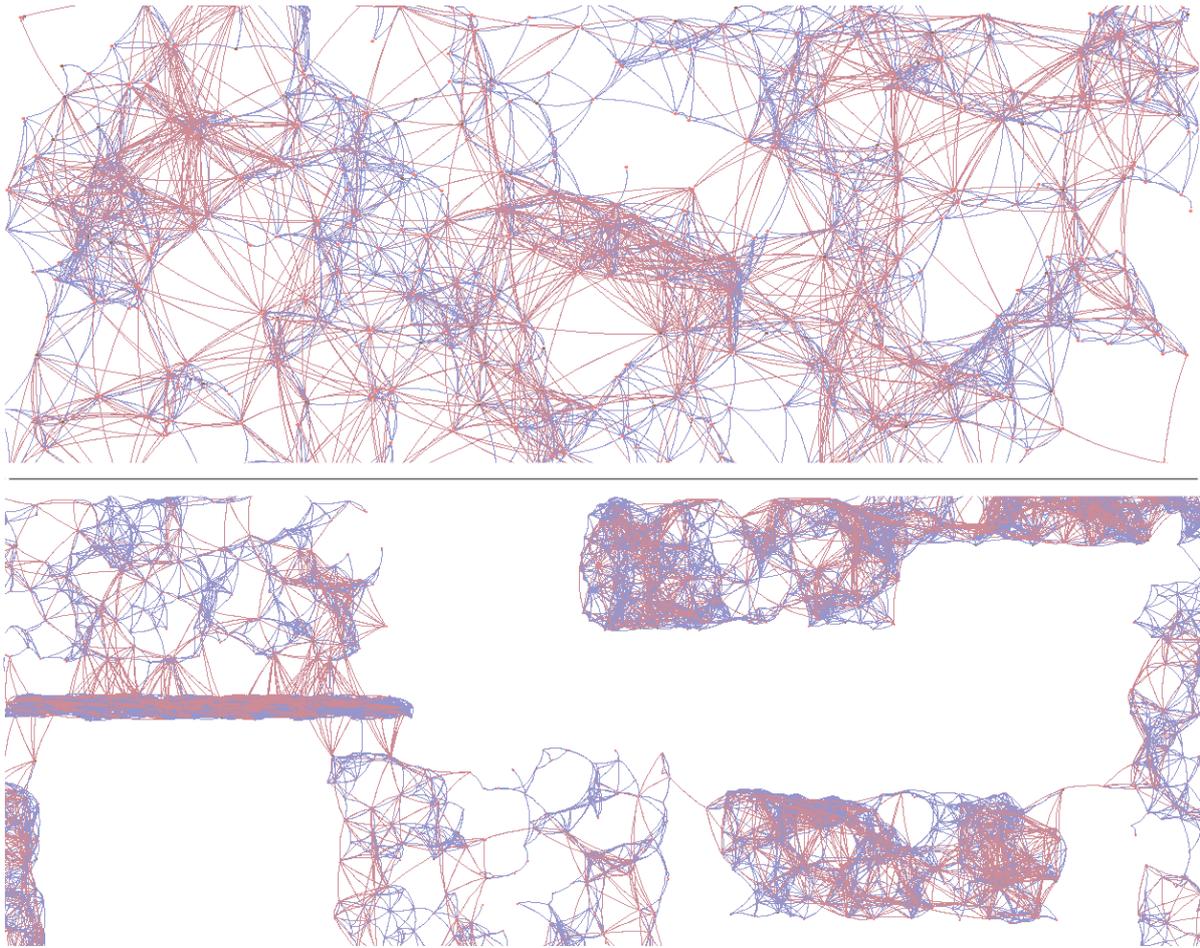


Abbildung 5.25.: Zwei spezifische Netzwerktopologien: eine zufällige Knotenverteilung (oben) sowie eine Topologie mit stark variierender Knotendichte (unten).

Abbildung 5.24 zeigt dazu ein Minimalbeispiel mit zwei disjunkten Netztopologien. Die beiden homogenen Teilmengen werden mit Hilfe eines einzelnen eingestreuten EAN-Knotens in eine heterogene Gesamtstruktur integriert. Die Erreichbarkeit steigt hierbei von 50% auf 100%. Abbildung 5.25 verdeutlicht die Problematik an zwei Testszenarien. Die beiden Farben der Kanten repräsentieren zwei Kommunikationstechnologien. Es wird deutlich, dass die EAN Knoten auf zwei Ebenen als *Brückenknoten* fungieren. Zum einen dienen sie der Funkstandardintegration und gleichzeitig ermöglichen sie die Erreichbarkeit bisher isolierter Teilbereiche der Topologie.

Um im Rahmen der durchgeführten Messreihen statistische Aussagen treffen zu können, ist zunächst eine globale Sicht auf die Testszenarien notwendig. Aus diesem Grund verfügt SimANet über eine zusätzliche, globale Topologiekontrollschicht, welche die Analyse und Auswertung durchgeführter Testszenarien vereinfacht. Diese Schicht hat keinen Einfluss auf die zu testenden Messgrößen. Auf Analyseebene stehen alle aktuellen Knoteninformationen wie bspw. Vernetzungsgrad, Energieressourcen, Routingeinträge oder aktive Kommunikationsstandards zur Verfügung, wodurch die Ad Hoc Topologie jederzeit ausgewertet werden können.

Alle Rahmenbedingungen für die nachfolgenden Simulationen wurden in Tabelle 5.1 zusammengefasst:

5. Simulationsumgebung SimANet

<i>Anzahl Knoten</i>	500-5000
<i>Knotenverteilung</i>	zufällig / hop-spot
<i>Energiemodell</i>	ohne Relevanz
<i>Netzlast</i>	netzlastunabhängig
<i>Knotenbewegung</i>	statische Topologie
<i>Kommunikationsstandards</i>	RS_1 Low-Power RS_2 Mid-Range
<i>Anteil EAN Knoten</i>	schrittweise von 0% bis 100% EAN ; Restmenge je nach Szenario unterteilt in RS_1 / RS_2
<i>Routing</i>	ohne Relevanz
<i>Eingabegröße</i>	Anteil EAN-konformer Knoten
<i>Kenngrößen</i>	Grad der Erreichbarkeit Anzahl disjunkter Teilmengen in der Topologie

Tabelle 5.1.: Übersicht über die Rahmenbedingungen in den folgenden Simulationen.

Um die konzeptionellen Ziele auf ihre Umsetzbarkeit zu testen, wurde zunächst verschiedene Testszenarien mit zwei fiktiven Funkstandards RS_1 und RS_2 umgesetzt. Die maximale Sendereichweite T_1 des ersten Funkstandards beträgt 50% des zweiten Standards. T_2 entspricht folglich $2 \cdot T_1$.

Ausgangspunkt jeder Testreihe ist eine zufällig generierte Netztopologie mit einer vordefinierten Anzahl Knoten und Knotenverteilung. Je nach Testszenario verfügen die Knoten über jeweils einen der Funkstandards RS_1 oder RS_2 . Im Anschluss werden nach einem zufälligen Muster einzelne Knoten durch EAN-konforme Multistandard-Knoten ersetzt, welche über ein zweites, unabhängiges Funkmodul verfügen.

Gemessen wird die Anzahl disjunkter/isolierter Knotenmengen sowie die prozentuale Erreichbarkeit der Topologie. Während der Analyse wird dazu durch die SimANet Topologiekontrollschicht eine Breitensuche beginnend von jedem Knoten initialisiert. Die Anzahl erreichbarer Elemente wird anschließend in Relation zur Gesamtknotenmenge gesetzt. Der errechnete Mittelwert repräsentiert die durchschnittliche Erreichbarkeit in der Topologie.

Die dargestellten Diagramme beinhalten Durchschnittswerte von 10 unabhängigen Testreihen unter gleichen Voraussetzungen und Rahmenbedingungen.

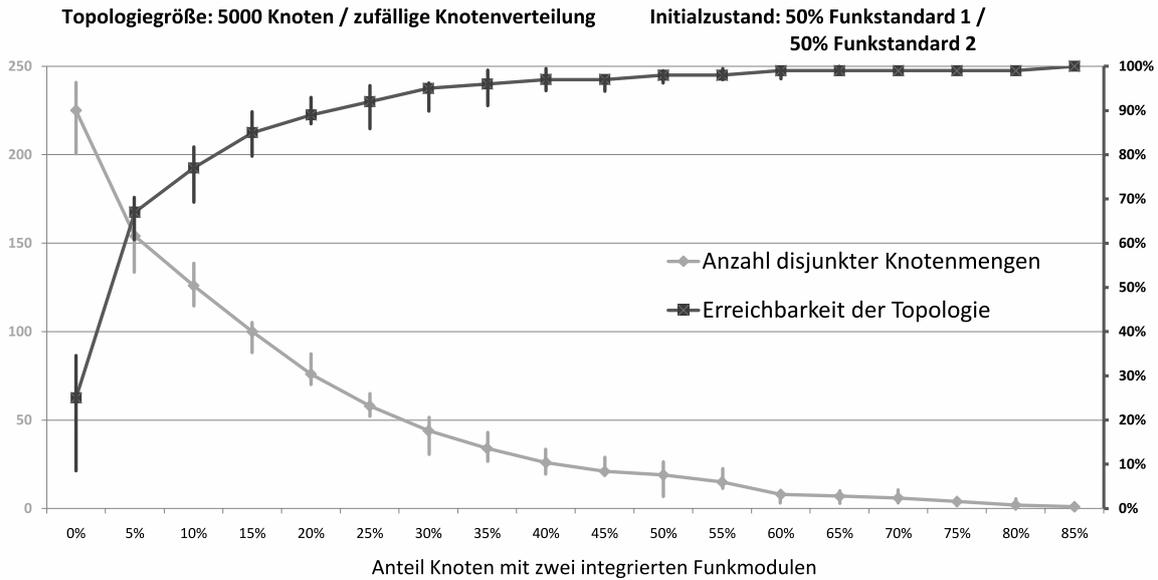


Abbildung 5.26.: Netzwerktopologie mit zunächst zwei separaten Funkstandards. Durch das sukzessive Ersetzen einfacher Knoten durch EAN-konforme Systeme mit zwei simultan betriebenen Funkmodulen wird die Integration unterschiedlicher Kommunikationsstandards ermöglicht. Zusätzlich erfolgt die Einbindung bisher isolierter Knoten bzw. Knotenmengen durch die geschaffene, heterogene Infrastruktur. Das Diagramm zeigt dabei Werte hinsichtlich der Erreichbarkeit der Knotentopologie sowie die Menge isolierter Knotenbereiche in Abhängigkeit vom Anteil EAN-konformer Knoten. Die jeweiligen Konfidenzintervalle wurden mit integriert.

Diagramm 5.26 zeigt eine Messreihe mit 5.000 zufällig verteilten Knoten, wobei initial 50% der Knoten mit Funkstandard RS_1 und 50% mit RS_2 ausgestattet wurden. Die beiden Messkurven verdeutlichen den Zusammenhang der beiden Messgrößen. Die Ergebnisse zeigen bereits bei einer EAN Einstreuung von 20% signifikante Verbesserungen beider Werte.

Die folgenden Diagramme 5.27 sowie 5.28 ermöglichen einen direkten Vergleich der Optimierungsergebnisse in Abhängigkeit von den definierten Rahmenbedingungen hinsichtlich Topologiegröße, Knotendichte sowie Knotenverteilung.

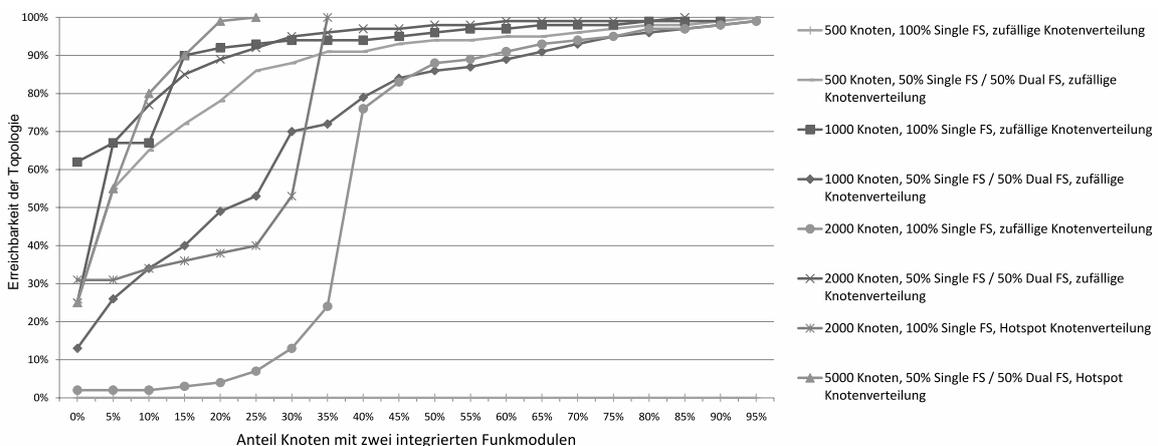


Abbildung 5.27.: Abhängigkeit der Optimierungsergebnisse von der Topologiegröße, der Knotenverteilung sowie der Initialbelegung aller Funkstandards.

5. Simulationsumgebung SimANet

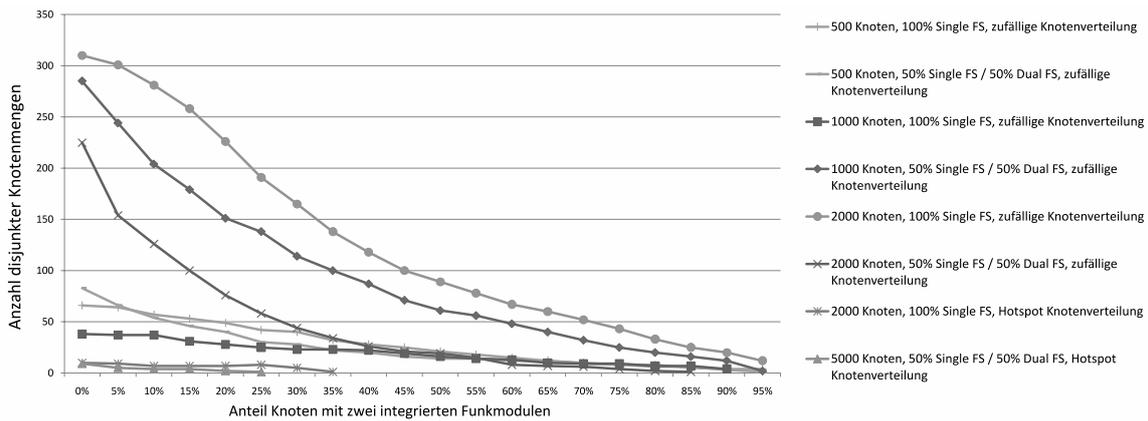


Abbildung 5.28.: Verringerung isolierter Knotenmengen durch Einstreuung EAN-konformer Knoten.

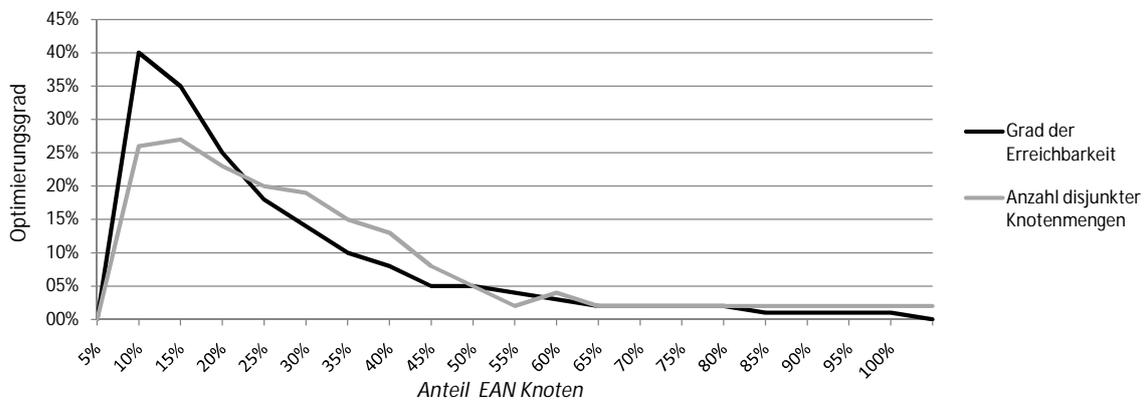


Abbildung 5.29.: Darstellung des durchschnittlichen Optimierungspotentials über alle durchgeführten Messungen in Abhängigkeit vom EAN Knotenanteil. Innerhalb der ersten 30% eingestreuter EAN Knoten zeigen sich die besten Optimierungswerte.

Resultierend aus den gezeigten Messergebnissen wird deutlich, dass unabhängig von den gewählten Rahmenbedingungen ein erhebliches Optimierungspotenzial ermöglicht wird. In jedem Testszenario erheben sich innerhalb der ersten 30% Einstreuung die deutlichsten Verbesserungen (siehe Abbildung 5.29). Der Vernetzungsgrad und folglich die Erreichbarkeit steigen hier durchschnittlich um 40% und erreichen Absolutwerte von ca. 80%. Die Testergebnisse zeigen somit die Vorteile des präsentierten EAN Ansatzes in Bezug auf die These einer Steigerung der Erreichbarkeit. Dabei nimmt die Topologiegröße nur eine untergeordnete Priorität ein.

5.6.3. Testreihen EAN - Worst Case Pfadkosten

Nachdem das Optimierungspotential des EAN Konzeptes bezüglich einer Steigerung der Erreichbarkeit verdeutlicht wurde, soll nun auf die damit verbundenen Kommunikationskosten eingegangen werden. Dabei erhöhen sich einerseits die Betriebskosten der Knoten durch mehrere simultan agierende Funkmodule. Andererseits ergeben sich daraus Optimierungsansätze für die Routenpfade.

Grundlage für jede Bewertung ist eine geeignete Kostenfunktion. Diese repräsentiert den Aufwand einer Übertragung mit dem gewählten Funkstandard durch abstrakte Parameter. Die gewählten Kostenwerte basieren dabei auf den Spezifikationen verfügbarer Kommunikationsstandards sowie auf den ermittelten Messwerten der *PLANet* Hardwarekomponenten (Kapitel 6). Für die durchgeführten Simulationen wird der Kostenvektor in die folgenden Parameter aufgeschlüsselt:

KATEGORIE	PARAMETER	RS_1	RS_2
Energiebedarf:	<i>Basiskosten pro Hop</i> C_{base}	20	20
	<i>Sendeleistung</i> C_{tpow}	0.8-8	5-50
zeitliche Aspekte:	<i>Verbindungsinitialisierung</i> C_{init}	4	8
	<i>Übertragungsdauer</i> C_{ttime}	2-20	0.1-1
	<i>Übertragungslatenzen</i> C_{lat}	20	10
Cross-Layer Parameter:	<i>aktueller Ladezustand</i> $C_{e.level}$	1-10	1-10
	<i>Funkstandardwechsel</i> C_{swap}	10	10

Tabelle 5.2.: Definition der beiden Funkstandards RS_1 und RS_2 mit der Aufschlüsselung aller abstrakter Kostenparameter.

Alle Parameter für die Funkstandards RS_1 und RS_2 werden in Tabelle 5.2 beschrieben. Die Tabelle ist in mehrere Bewertungskategorien untergliedert. Um Aussagen hinsichtlich der Güte eines Routenpfades treffen zu können, müssen nun diese unterschiedlichen Parameter in eine Gesamtfunktion integriert werden. Zum besseren Verständnis hier nochmals der bereits in Kapitel 4.4.4 vorgestellte Kostenvektor und dessen Herleitung:

$$C_{e.level} = \frac{1}{P_{current}}, 0.1 \leq P_{current} \leq 1.0$$

$$C_{tpow} = 1 + (C_{tpow_{max_{RS_x}}} \cdot P_{transmit}^2)$$

$$0.1 \leq P_{transmit} \leq 1.0$$

$$C_{ttime} = C_{ttime_{max_{RS_x}}} \cdot P_{transmit}$$

$$0.1 \leq P_{transmit} \leq 1.0$$

$P_{current}$... aktueller Ladezustand

$P_{transmit}$... zur Übertragung nötige Sendeleistung

5. Simulationsumgebung SimANet

Die Definition der Gesamtkostenfunktion $C_{overall}$ lautet:

$$\begin{aligned}
 C_{overall} &= \sum_{i=1}^{\#Hops} \left(\sum_{j=1}^n (w_j \cdot C_j) \right) \\
 &= \sum_{i=1}^{\#Hops} (w_{base} \cdot C_{base} + w_{e_level} \cdot C_{e_level} \\
 &\quad + w_{init} \cdot C_{init} + w_{tpow} \cdot C_{tpow} + w_{ttime} \cdot C_{ttime} \\
 &\quad + w_{lat} \cdot C_{lat} + w_{swap} \cdot C_{swap})
 \end{aligned}$$

n ... Anzahl Parameter

w_x ... Gewichtung des Parameters x

Gewichtung des Kostenvektors:

Um die jeweiligen Parameter in Abhängigkeit vom gewünschten Anwendungsszenario in Relation setzen zu können, kommt eine Gewichtungsfunktion zum Einsatz. Durch diese lassen sich die einzelnen Kostenparameter priorisieren oder ganz ausblenden, was wiederum Auswirkungen auf die Metriken von Routing- oder Topologieoptimierungsverfahren hat. Im Rahmen der hier durchgeführten Simulationen wurden zunächst zwei anwendungsspezifische Gewichtungen der jeweiligen Parameter definiert W_{S1} und W_{S2} :

W_{S1} : Optimierung des Energiebedarfs

W_{S2} : Minimierung von Latenzen während der Übertragung

GEWICHTETER PARAMETER	W_{S1}	W_{S2}
w_{base}	1.0	1.0
w_{e_level}	1.0	0.7
w_{init}	0.1	0.3
w_{tpow}	0.6	0.3
w_{ttime}	0.3	0.2
w_{lat}	0.1	0.5
w_{swap}	1.0	1.0

Tabelle 5.3.: Zwei unterschiedliche Gewichtungsszenarien für die Kostenfunktion C . Dabei sind die Basiskosten für den Betrieb C_{base} sowie die Kosten für einen Funkstandardwechsel C_{swap} konstant definiert und haben keine Gewichtung.

Teile der folgenden Simulationsergebnisse wurden im Rahmen der Veröffentlichungen [6, 8–10] präsentiert. Die durchgeführten Szenarien beinhalten sowohl gleichmäßig verteilte (uniforme) als auch zufällig initialisierte Knotentopologien unterschiedlicher Größe (Abbildung 5.30). Ausgangssituation ist jeweils ein homogenes Netzwerk mit ausschließlich einem einheitlichen

Kommunikationsstandard. Anschließend werden nun auch hier nach einem zufälligen Muster einzelne Knoten durch Multistandard-Knoten ersetzt, welche über ein zweites, unabhängiges Funkmodul verfügen. Die gegensätzlichen Eigenschaften der beiden Kommunikationsstandards werden durch spezifische Werte innerhalb der Kostenvektoren abgebildet. Jeder Parameter hat somit direkten Einfluss auf die Optimierungsergebnisse.

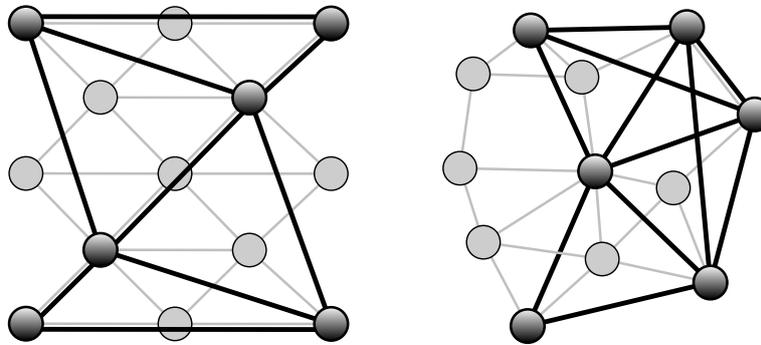


Abbildung 5.30.: Beispielhaft zwei Netzwerktopologien mit gleichförmiger (links) und zufälliger Knotenverteilung (rechts). Beide Topologien beinhalten im aktuellen Zustand einen Anteil von 50% Multistandard-Knoten.

<i>Anzahl Knoten</i>	20-120
<i>Initialzustand</i>	homogene Topologie auf Basis eines Funkstandards
<i>Knotenverteilung</i>	uniform / zufällig
<i>Energiemodell</i>	integriert / Werte statisch
<i>Netzlast</i>	netzlastunabhängig
<i>Knotenbewegung</i>	statische Topologie
<i>Kommunikationsstandards</i>	RS_1 Low-Power RS_2 Mid-Range
<i>Anteil EAN Knoten</i>	schrittweise von 0% bis 50% EAN ; Restmenge RS_1
<i>Routing / Routenmetrik</i>	kostenoptimiert
<i>Kostenfunktion</i>	$C_{overall}$
<i>Gewichtungsszenario</i>	W_{S1} energieoptimiert W_{S2} latenzoptimiert
<i>Eingabegröße</i>	Anteil EAN-konformer Knoten
<i>Kenngrößen</i>	Kosten Worst Case Routenpfad Anzahl Hops Worst Case Routenpfad

Tabelle 5.4.: Übersicht über die Rahmenbedingungen in den folgenden Simulationen.

Im Zuge der jeweiligen Testläufe wird nun der prozentuale Anteil der Multistandard-Knoten beginnend bei 10% sukzessiv bis zu 50% erhöht. Die in den Diagrammen dargestellten Werte repräsentieren jeweils Durchschnittswerte von 10 unabhängigen Testreihen mit gleichen Para-

5. Simulationsumgebung SimANet

metern für die Rahmenbedingungen. Wesentliche Messgrößen sind hierbei die abstrakten Gesamtkosten für die Kommunikation sowie die Anzahl benötigter Hops zwischen der vordefinierten Datenquelle und -senke.

Quelle und Senke werden dabei so gewählt, dass deren räumliche Distanz in der initialen Topologie maximal ist und die Kommunikation auf einem längsten Pfad erfolgen muss (*Worst Case Kommunikation*). Tabelle 5.4 fasst alle Rahmenbedingungen noch einmal zusammen.

Gleichmäßige Knotenverteilungen

Die Messreihen in Abbildung 5.31 und 5.32 zeigen die erzielten Optimierungsergebnisse in gleichförmig verteilten Knotentopologien unterschiedlicher Größe. Nachdem in den zuvor präsentierten Testreihen große Topologien mit 1.000+ Knoten simuliert und hinsichtlich einer Erreichbarkeitssteigerung analysiert wurden, fokussieren die folgenden Szenarien kleinere Topologien aus typischen Anwendungsbereichen. Das Optimierungspotential fällt bei der latenzoptimierten Gewichtung deutlich besser aus, was im Wesentlichen auf den Kostenparameter der benötigten Übertragungsleistung zurückzuführen ist. So beträgt die Kostendifferenz zwischen beiden Funkstandards RS_1 und RS_2 im ersten energieoptimierten Gewichtungsszenario deutlich mehr als in der latenzoptimierten Gewichtung. Die Unterschiede wirken sich somit auf die Routenwahl bzw. die genutzte Anzahl Hops aus. An dieser Stelle sei darauf hingewiesen, dass die Optimierungsergebnisse stark von den definierten Parametern und Wichtungen des Kostenvektors abhängen. Für die gewählten abstrakten Kostenwerte der Funkstandards fließen neben den offiziellen Spezifikationen vor allem die praktischen Erfahrungen mit dem PLANet Prototypen ein.

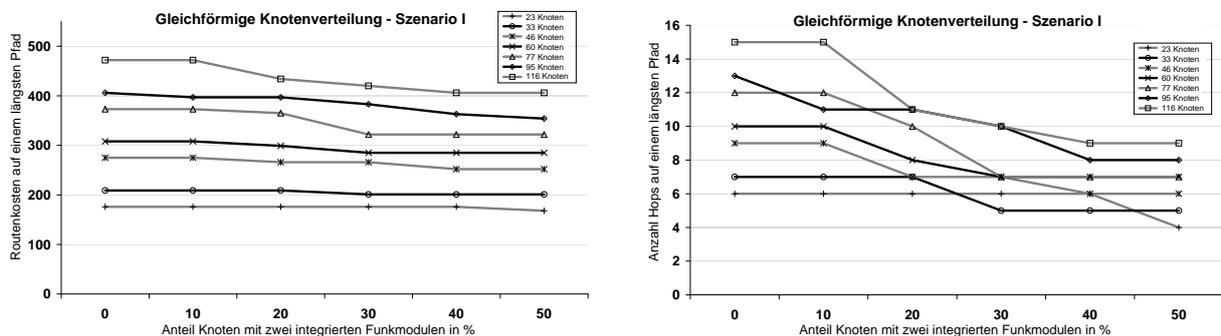


Abbildung 5.31.: Uniforme Knotenverteilungen unterschiedlicher Größe. Dabei kam als Gewichtungsfunktion das Szenario 1 zum Einsatz - Optimierung des Energiebedarfs (Tabelle 5.3). Links die ermittelten Gesamtkosten für die Übertragung eines Datenpaketes entlang eines in der Topologie längsten Pfades. Rechts die dazugehörige Anzahl benötigter Hops von der Quelle zur Senke. Auf der X-Achse ist jeweils der gewählte Prozentsatz verfügbarer Multistandard-Knoten abgebildet, welche in jedem Testlauf randomisiert ausgewählt wurden.

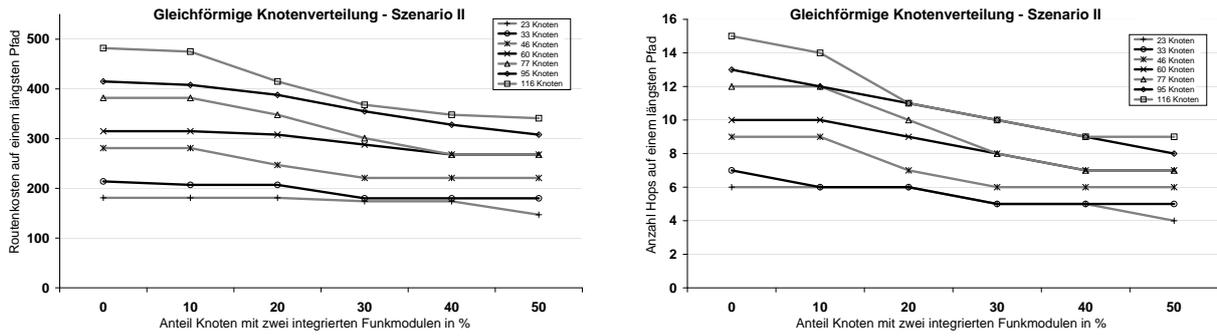


Abbildung 5.32.: Uniforme Knotenverteilungen. Darstellung analog Abbildung 5.31. Als Gewichtungsfunktion kam jedoch Szenario 2 zum Einsatz - Latenzminimierung (Tabelle 5.3).

Zufällige Knotenverteilungen

Die Abbildungen 5.33 und 5.34 zeigen anschließend die Testreihen mit zufällig verteilten Netztopologien. Aus den erwartungsgemäßen Verkürzungen der Routenpfade resultiert eine Verringerung der Anzahl benötigter Hops. Aber auch in Hinblick auf die Pfad- bzw. Übertragungskosten sind hier deutliche Verbesserungen ersichtllich.

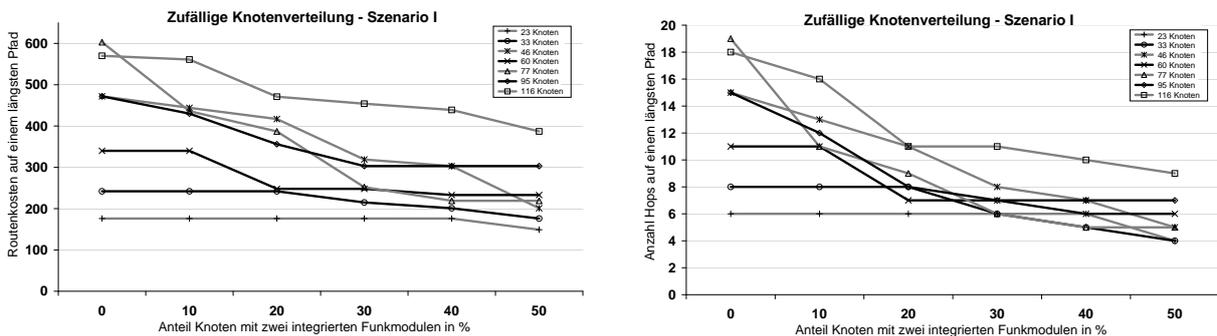


Abbildung 5.33.: Zufällige Knotenverteilungen unterschiedlicher Größe. Darstellung analog zu vorherigen Messreihen. Als Gewichtungsfunktion kam Szenario 1 zum Einsatz - Optimierung des Energiebedarfs (Tabelle 5.3).

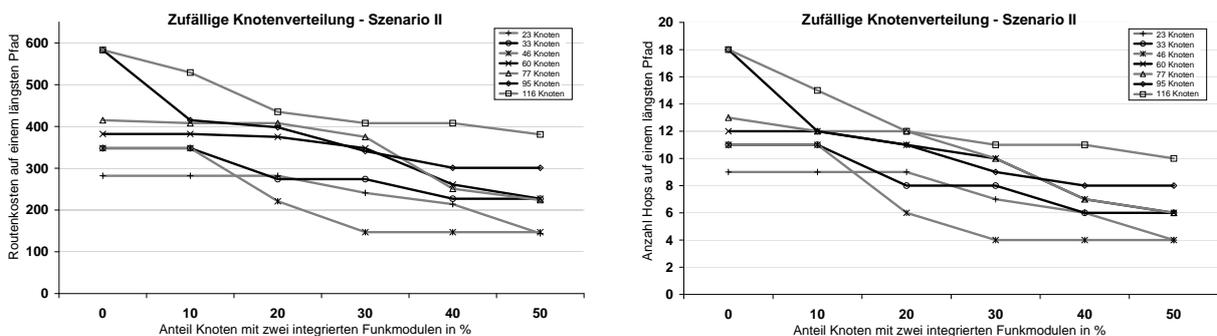


Abbildung 5.34.: Zufällige Knotenverteilungen. Gewichtungsfunktion: Szenario 2 - Latenzminimierung.

5. Simulationsumgebung SimANet

Die Diagramme verdeutlichen die konzeptionellen Vorteile einer funkstandardübergreifenden, heterogenen Netztopologie. Erreicht der Prozentsatz eingestreuter Multistandard-Knoten einen Schwellwert von ca. 20%, sinken die Gesamtkosten für die Datenübertragung trotz erhöhter Basiskosten teils signifikant. Speziell in den zufälligen Knotenverteilungen konnten deutliche Optimierungen erzielt werden. In Folge der Nutzung zusätzlicher Kommunikationspfade können die Daten nun effizienter von der Quelle zur Senke transportiert werden, wodurch sich die Anzahl benötigter Hops im Mittel deutlich reduziert. Bei einer Annäherung an die 50% Marke wird in nahezu allen Fällen ein optimales Ergebnis erzielt. Die weitere Erhöhung des Anteils an Multistandard-Knoten bringt somit nur noch minimale und folglich vernachlässigbare Verbesserungen. Abbildung 5.35 visualisiert diese Ergebnisse.

Zusätzlich konnte mit den Simulationen gezeigt werden, dass das Optimierungspotential mit zunehmender Größe der Topologie ansteigt. Besonders bei erhöhter Knotendichte wirken sich die Vorteile durch das Einstreuen eines zweiten Funkstandard stärker aus. Hier können mehr Hops auf den Routenpfaden eingespart werden. Knoten, die für die Paketweiterleitung nicht mehr benötigt werden, können die verfügbaren Energieressourcen schonen. In Folge dessen verlängert sich die durchschnittliche Knotenlaufzeit und die Erreichbarkeit der gesamten Netztopologie.

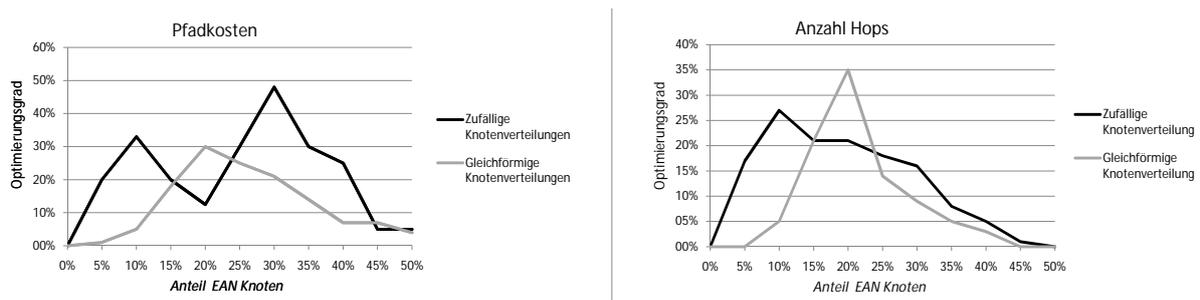


Abbildung 5.35.: Darstellung des durchschnittlichen Optimierungspotentials über alle durchgeführten Messungen in Abhängigkeit vom EAN-Knotenanteil. Während in gleichförmigen Knotenverteilungen bei 20% EAN-Knoteneinstreuung die besten Optimierungsergebnisse erzielt werden, liegen bei zufälligen Verteilungen zwei Maxima bei 10% sowie 30%.

Voraussetzung hierfür ist eine hinreichend hohe Netzlast. Im Zusammenhang mit dieser Kommunikation muss auch das Energieniveau der jeweiligen Knoten beachtet werden. Ein entsprechend adaptierter Routingalgorithmus ist notwendig.

5.6.4. Testreihen EAN - Dynamisches Verhalten

Zur Laufzeit benötigen die Knoten kontinuierlich Energie für die Betriebsbereitschaft und die Kommunikation. In EAN-konformen Knoten sind diese Basis-/Betriebskosten durch simultan betriebene Funkmodule erhöht. Es soll nun untersucht werden, inwiefern bei einer generierten Netzlast eine positive Energiebilanz in der Topologie erzielt werden kann. Dazu muss zunächst das zugrundeliegende Energiemodell definiert werden, nachdem die Energieressourcen jedes Knotens berechnet und aktualisiert werden.

Energiemodell

Jede Aktivität eines Knotens in der Topologie, sowohl die reine Betriebsbereitschaft als auch der Empfang, die Verarbeitung und die Übermittlung von Daten wird auf abstrakte Energieeinheiten abgebildet. Für die folgenden Simulationen werden einheitliche Energieressourcen für alle Knoten im Netzwerk definiert. Dabei stehen initial 10.000 logische Einheiten pro Knoten zur Verfügung. Eine dynamische Anpassung der verbleibenden Energieressourcen soll mit Hilfe der Funktion ΔP_{node} erfolgen, welche nachstehend definiert wird:

$$\begin{aligned} \Delta P_{node} &= f(\Delta T, M, P_{transmit}, RS_x) \\ \Delta P_{node} &= (\#_{packets} \cdot P_{transmit}^2 \cdot g(RS_x)) + (\Delta T \cdot b) \\ g &: (RS_x) \rightarrow N \\ g(RS_x) &= \begin{cases} 1, \text{Übertragung mittels } RS_1 \\ 6, \text{Übertragung mittels } RS_2 \end{cases} \\ b &: \text{Basiskosten für die Betriebsbereitschaft} = 2 \end{aligned}$$

T ... simulierte Zeit
 M ... Nachrichtengröße (in Paketen)
 $P_{transmit}$... benötigte Feldstärke / Sendeleistungsniveau
 RS_x ... genutzter Kommunikationsstandard

In Abhängigkeit von den verwendeten Funkmodulen und der Kommunikationslast variiert die Lebensdauer jedes Knotens.

Routing

Um das Optimierungspotential nutzen zu können, kommt das vorgestellte EBCR Verfahren als Routingstrategie zum Einsatz. Im Fokus der Analysen steht dabei die Fähigkeit von EBCR, Unterschiede in den Energieniveaus einzelner Knoten durch angepasste Routenpfade auszubalancieren. Ein Großteil bekannter Routingverfahren wählt ausschließlich latenz- oder hop-optimierte Metriken für die Berechnung der Routenpfade. In vielen Fällen führt das zu einer unausgewogenen Belastung einzelner Knoten im Netzwerk. In direkter Konsequenz sind diese Knoten aufgrund verbrauchter Energieressourcen vorzeitig nicht mehr verfügbar.

Durch das Ausbalancieren des topologieweiten Energieniveaus auf der Basis von EBCR können solche Schwachstellen in der Kommunikationsinfrastruktur gezielt umgangen werden.

Abbildung 5.36 zeigt ein typisches Szenario mit zufälliger Knotenverteilung. Durch die Mobilität der Knoten ergeben sich häufig starke Schwankungen in der Knotendichte, was zu sogenannten *Hot Spots*, also Bereichen sehr hoher Knotendichte und somit sehr hohem Vernetzungs-

5. Simulationsumgebung SimANet

grad, führt. Zwischen den Hot Spots existieren nur wenige Verbindungen, was in den betreffenden Netzwerkknoten aufgrund erhöhter Datenmengen zu Lastspitzen führt. Die Lebensdauer dieser Knoten verkürzt sich in Folge des hohen Energiebedarfs für die Datenweiterleitung sehr stark. Die müssen die Routingverfahren in solchen Fällen kooperative Strategien verfolgen, da sonst nach sehr kurzer Zeit isolierte, disjunkte Knotenmengen entstehen.

Für die Berechnung des optimalen Routenpfades kommt das bereits eingeführte Gewichtungsszenario W_{S1} (Energieoptimierung) zum Einsatz, welches im vorangegangenen Abschnitt 5.6.3 definiert wurde. Die Werte der jeweiligen Kostenparameter bleiben ebenfalls unverändert. Auch in den folgenden Simulationen werden sowohl uniforme/gleichmäßige als auch zufällig verteilte Knotentopologien unterschiedlicher Größe. Die dargestellten Werte repräsentieren den Durchschnitt aus zehn unabhängigen Messreihen unter identischen Rahmenbedingungen. EBCR werden zwei einfache Routingstrategien gegenübergestellt. Zum einen ein triviales hop-optimiertes Verfahren, welches grundsätzlich kürzeste Pfade mit möglichst wenig Zwischenknoten wählt. Weiterhin kommt ein kostenoptimiertes Routingverfahren zum Einsatz, welches auf den gleichen Kostenvektoren wie EBCR arbeitet, jedoch ohne Cross-Layer Strategien. Somit sind Informationen über die verbleibenden Energieressourcen nicht verfügbar und werden in der Routenmetrik nicht berücksichtigt. Die Rahmenbedingungen werden in Tabelle 5.5 noch einmal zusammengefasst.

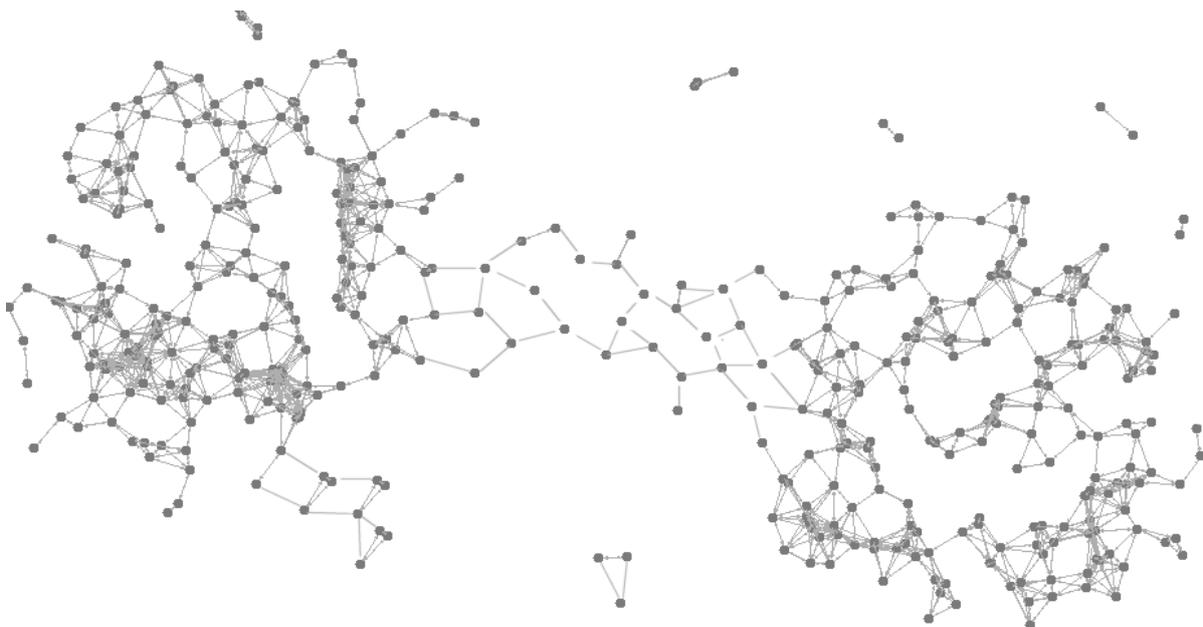


Abbildung 5.36.: Typische Knotenverteilung mit stark variierender Knotendichte (Hot Spot Szenario).

<i>Anzahl Knoten</i>	30-60
<i>Knotenverteilung</i>	uniform / zufällig
<i>Energiemodell</i>	integriert unter Berücksichtigung der Last
<i>Initiale Energieressourcen</i>	einheitlich fest definiert 10.000 Einheiten
<i>Netzlast</i>	randomisiert paketorientiert
<i>Knotenbewegung</i>	statisch / dynamisch (RWP)
<i>Kommunikationsstandards</i>	RS_1 Low-Power RS_2 Mid-Range
<i>Anteil EAN Knoten</i>	50% EAN ($RS_1 + RS_2$) und 50% RS_1
<i>Routing / Routenmetrik</i>	EBCR-Cross-Layer / kostenoptimiert ohne Cross-Layer / Hop-Count
<i>Kostenfunktion</i>	$C_{overall}$
<i>Gewichtungsszenario</i>	W_{S1} energieoptimiert
<i>Eingabegröße</i>	Netzlast
<i>Kenngrößen</i>	maximale Anzahl übertragener Nachrichten verbleibende Anzahl erreichbarer Knoten

Tabelle 5.5.: Übersicht über die Rahmenbedingungen in den folgenden Simulationen.

Ablauf der Simulationsszenarien

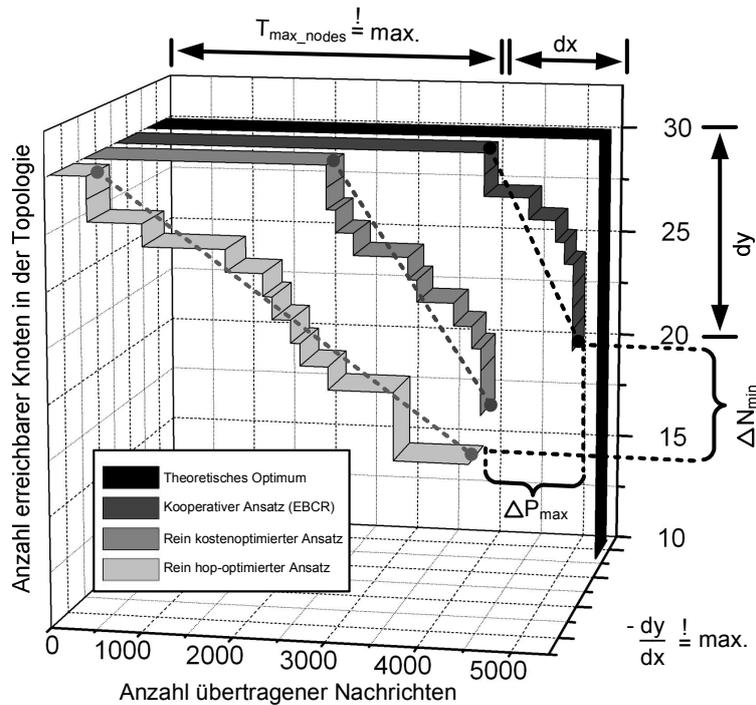


Abbildung 5.37.: Erläuterung des verwendeten Diagrammtyps.

5. Simulationsumgebung SimANet

Der Diagrammtyp in Abbildung 5.37 wird im Rahmen der EBCR Testauswertung eingesetzt. Im dargestellten Beispiel sind drei verschiedene Routingstrategien abgebildet. Ziel der Simulationen ist es, die Erreichbarkeit aller Knoten so lange wie möglich aufrecht zu erhalten ($T_{max.nodes}$). Folglich repräsentiert die schwarze Linie das theoretische Optimum für das Kommunikationsszenario. Aus dieser Forderung resultiert direkt die Maximierung des Anstiegs in negativer Ausrichtung ($-\frac{dx}{dy}$).

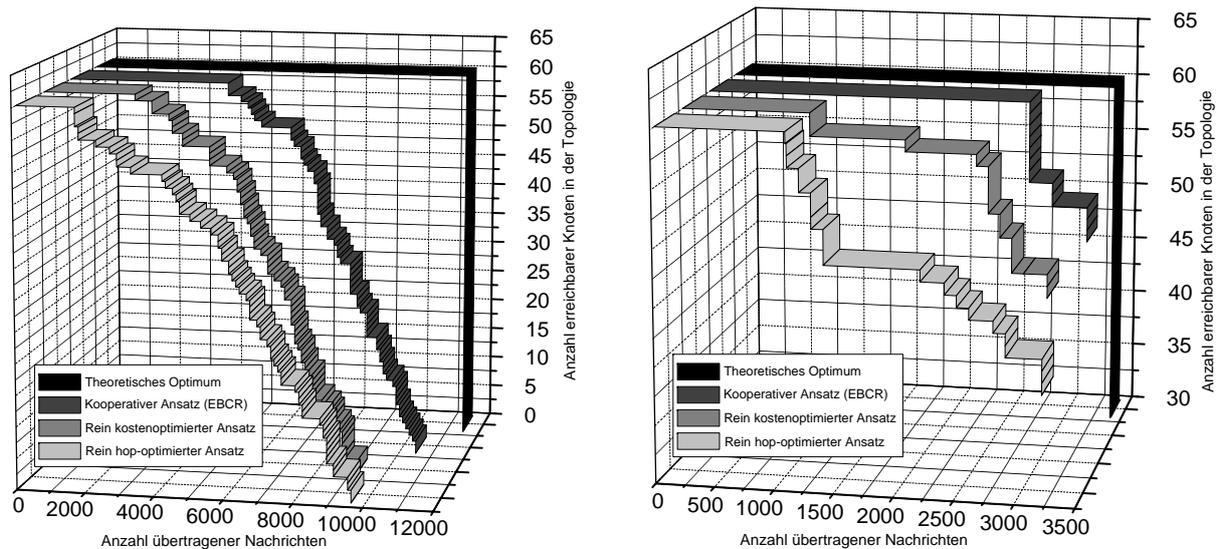
In den Testreihen werden zwei Anwendungsszenarien definiert. *Szenario I - Laufzeittest* analysiert die Verfügbarkeit einer gegebenen Topologie. Diese ist dynamisch und nutzt das Random Waypoint Mobility Model (siehe 5.2.2). Im Zuge der Simulation werden Quelle und Senke eines Kommunikationskanals periodisch zufällig definiert. Die Routingalgorithmen suchen daraufhin einen optimalen Pfad für die Übertragung des Datenpaketes. Nach erfolgter Übertragung wird Quelle und Senke erneut zufällig festgelegt.

Die Prozedur wird im Netzwerk solange wiederholt, bis alle Knoten aufgrund verbrauchter Energieressourcen nicht mehr zur Verfügung stehen. Die entscheidende Messgröße bildet in diesen Testreihen die Anzahl erfolgreich übertragener Nachrichten durch die unterschiedlichen Routingverfahren. Je länger alle Knoten verfügbar sind, desto größer ist die Auswahl alternativer Routenpfade - die Erreichbarkeit steigt, damit einhergehend auch die Anzahl übertragener Datenpakete (ΔP_{max}).

Ein zweites Anwendungsszenario repräsentiert eine *statische Punkt-zu-Punkt Kommunikation*. Quelle und Senke sind dabei fest definiert. Die gesamte Topologie arbeitet ohne Knotenbewegungen. Auch hier wird versucht, möglichst viele Nachrichten zu übermitteln. Je nach Routingstrategie wählen die Verfahren nach unterschiedlichen Metriken ihre Routenpfade. Die Simulation endet, sobald aufgrund verbrauchter Energieressourcen zwei disjunkte Knotenmengen entstehen. Der Zielknoten ist in Folge dessen nicht mehr erreichbar. Die relevanten Messgrößen sind hierbei die Anzahl übertragener Datenpakete sowie die Anzahl noch aktiver Netzwerkknoten (ΔN_{min}) nach Simulationsende.

Gleichmäßige Knotenverteilungen

Die Messreihen in Abbildung 5.38 verdeutlichen das Optimierungspotential von EBCR. Es konnte in beiden Anwendungsszenarien gezeigt werden, dass EBCR die maximale Anzahl übertragener Nachrichten im Gegensatz zu herkömmlichen Routingverfahren deutlich steigern konnte. Der Unterschied beträgt durchschnittlich 1.000 Pakete. Desweiteren konnte der Zeitraum, in dem alle Knoten im Netzwerk verfügbar sind ($T_{max.nodes}$), deutlich gesteigert werden.



(a) Szenario I - Laufzeittest - maximale Verfügbarkeit der Topologie: uniforme Knotenverteilung, 60 Knoten, 50% Anteil Multistandard-Knoten, Nachrichtengröße: 100 Datenpakete.

(b) Statisches Punkt-zu-Punkt Szenario: uniforme Knotenverteilung, 60 Knoten, 50% Anteil Multistandard-Knoten, Nachrichtengröße: 100 Datenpakete.

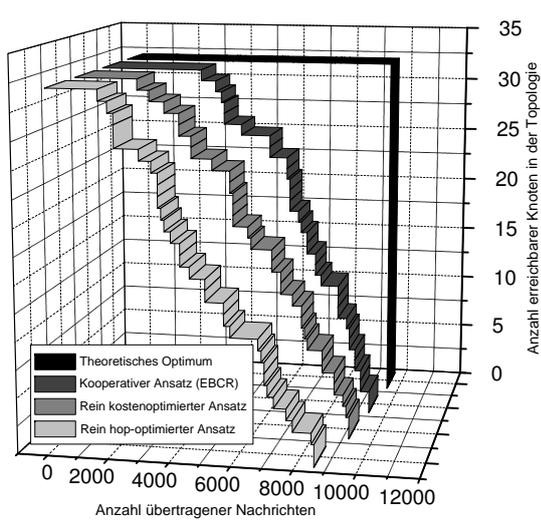
Abbildung 5.38.: Links ein Laufzeittest der gesamten Topologie bis die Energieressourcen aller Knoten verbraucht wurden. Rechts ein statisches Punkt-zu-Punkt Szenario zur Messung der optimalen Lastverteilung in der Topologie. Auf der X-Achse ist dabei die Anzahl erfolgreich übertragener Datenpakete abgetragen. Auf der Y-Achse die Anzahl noch aktiver Netzwerkknoten.

Zufällige Knotenverteilungen

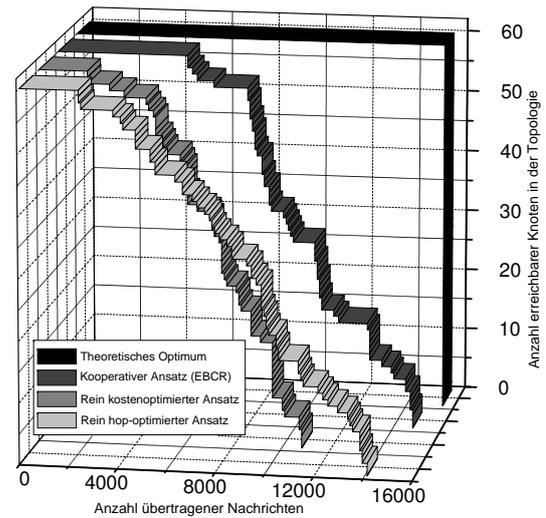
Abbildung 5.39 zeigt zwei Laufzeittests mit unterschiedlicher Topologiegröße. Wie erwartet, treten die Vorteile des EBCR-Verfahrens bei zunehmender Knotenmenge stärker in den Vordergrund. Im zweiten Szenario sind ebenfalls deutliche Verbesserungen ersichtlich. Gleichzeitig konnte die Anzahl der Knoten, welche aufgrund verbrauchter Energieressourcen nicht mehr verfügbar sind, gesenkt werden. Die Erreichbarkeit in der Topologie wird verbessert und bleibt über einen längeren Zeitraum auf konstant hohem Niveau. Das Optimierungspotential liegt hier bei durchschnittlich 15%, was einem Unterschied von 21 offline Knoten (bei einem einfachen hop-optimierten Routingverfahren) gegenüber 12 offline Knoten mittels EBCR entspricht.

Die Diagramme in Abbildung 5.40 zeigen die Abhängigkeiten der Simulationsergebnisse von der definierten Nachrichtengröße. Das EBCR Verfahren erzielt bei allen durchgeführten Test-szenarien hervorragende Ergebnisse und verdeutlicht das primäre Optimierungsziel - einer Maximierung der Erreichbarkeit jedes Knotens in der Topologie [5, 7].

5. Simulationsumgebung SimANet

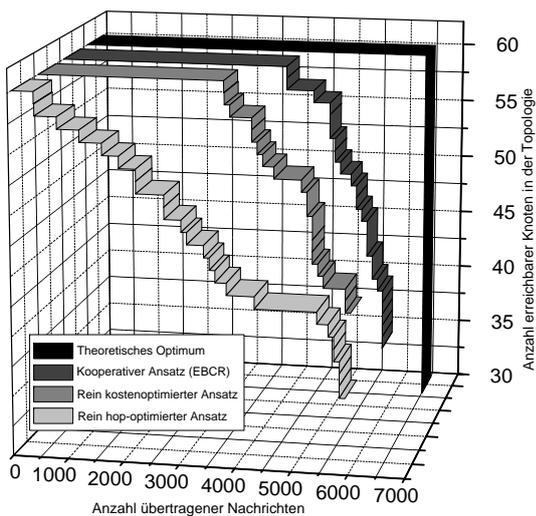


(a) Szenario I - Laufzeittest - maximale Verfügbarkeit der Topologie: zufällige Knotenverteilung, 30 Knoten, 50% Anteil Multistandard-Knoten, Nachrichtengröße: 100 Datenpakete.

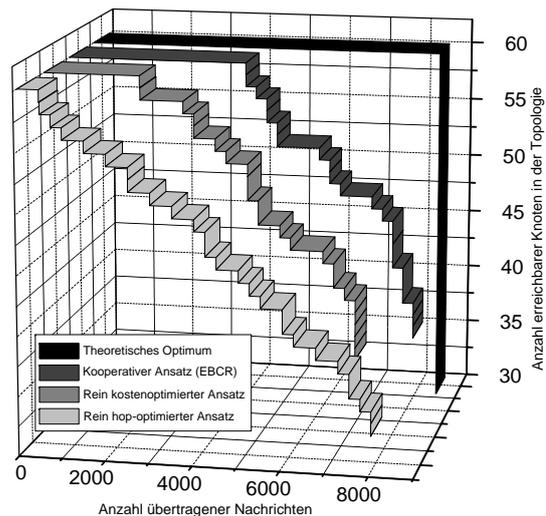


(b) Szenario I - Laufzeittest - maximale Verfügbarkeit der Topologie: zufällige Knotenverteilung, 60 Knoten, 50% Anteil Multistandard-Knoten, Nachrichtengröße: zufällig zwischen 50 und 500 Paketen.

Abbildung 5.39.: Laufzeittests mit zufälliger Knotenverteilung.



(a) Statisches Punkt-zu-Punkt Szenario: zufällige Knotenverteilung, 60 Knoten, 50% Anteil Multistandard-Knoten, Nachrichtengröße: 100 Datenpakete.



(b) Statisches Punkt-zu-Punkt Szenario: zufällige Knotenverteilung, 60 Knoten, 50% Anteil Multistandard-Knoten, Nachrichtengröße: zufällig zwischen 50 und 500 Paketen.

Abbildung 5.40.: Statische Punkt-zu-Punkt Szenarien mit zufälliger Knotenverteilung.

5.7. Zusammenfassung

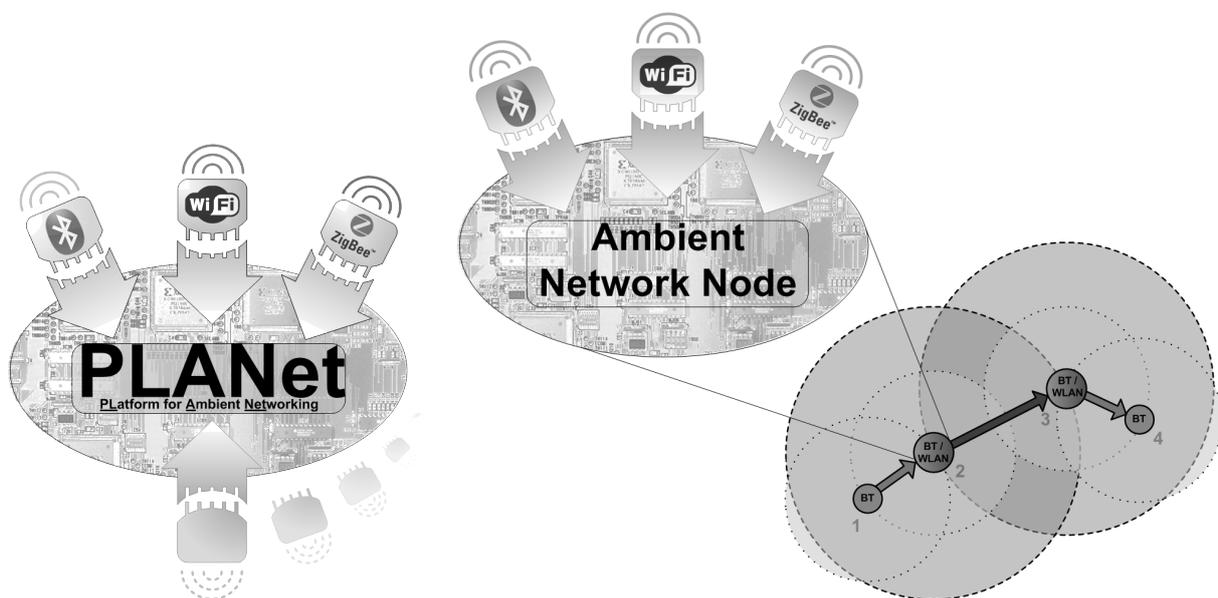
In diesem Kapitel wurde das vorgestellte Konzept für eine funkstandardübergreifende Kommunikation in mobilen Ad Hoc Netzwerken auf Basis der SimANet Plattform evaluiert. Die Simulationsplattform ermöglicht dabei sowohl statische als auch dynamische Testszenarien mit beliebig vielen parallel betriebenen Kommunikationsstandards. Die funkstandardspezifischen Parameter wurden dabei auf abstrakte Kostenwerte abgebildet. Durch die geschaffenen Möglichkeiten für verteilte Simulationen und die genutzten Datenstrukturen entstand ein Simulationswerkzeug mit sehr hoher Leistungsfähigkeit, wodurch nun auch Testreihen mit extrem großen Knotentopologien realisierbar wurden.

Primäres Anwendungsfeld von SimANet ist die Analyse konzeptioneller Studien neuer Routing- und Topologieoptimierungsverfahren. In Folge dessen wurde neben der Leistungsfähigkeit und der modularen Erweiterbarkeit besonderer Wert auf eine intuitive und einfache Konfiguration gelegt. Neue Funktionen und Module, bspw. Bewegungs- oder Energiemodelle können im Zuge der Weiterentwicklung problemlos integriert werden.

Durch die Testszenarien konnte die Skalierbarkeit und Leistungsfähigkeit des Simulationskerns nachgewiesen werden. Weitere Messreihen zeigten in Hinblick auf das EAN Konzept sowohl eine Steigerung der Topologielebensdauer als auch eine Verbesserung der Erreichbarkeit aller Knoten. Trotz eines parallelen Betriebs mehrerer Funkmodule und zusätzlichem Aufwand für die Protokollkonvertierungen konnte eine Senkung des durchschnittlichen Energiebedarfs in der Topologie erzielt. Resultierend aus den Simulationsergebnissen ist die Integration unterschiedlicher Funkstandards in eine heterogene, funkstandardübergreifende Netzwerktopologie effizient und wirtschaftlich umsetzbar. Die Erkenntnisse aus den durchgeführten Simulationen fließen gleichzeitig in die Entwicklung des nun vorgestellten Prototyps ein. Dieser soll die praktische Umsetzbarkeit eines EAN-konformen Kommunikationssystems demonstrieren.

6. Prototyp-Plattform PLANet

Nachdem ein Konzept für die funkstandardübergreifende Kommunikation in mobilen Ad Hoc Netzwerken entwickelt wurde und die Vorteile auf Basis der SimANet Umgebung evaluiert werden konnten, erfolgte die Umsetzung eines prototypischen, EAN-konformen Hardware-Systems. Es sind nun Messungen unter Realbedingungen durchführbar, welche die konzeptionellen Thesen stützen und einen Vergleich mit den Simulationsergebnissen ermöglichen.



(a) Logo der intelligenten, eingebetteten Kommunikationsplattform *PLANet* (*Platform for Ambient Networking*).

(b) *PLANet* im praktischen Einsatz als funkstandardübergreifender Knoten für die Integration inkompatibler Kommunikationsstandards in einer heterogenen Netztopologie.

Abbildung 6.1.: *PLANet*: Links das Logo des entwickelten Prototypen und rechts ein Anwendungsszenario.

Die Entwicklung von *PLANet - Platform for Ambient Networking* - ist dabei grundsätzlich als *Proof of Concept* aufzufassen. Die Plattform demonstriert eine Integration mehrerer Funkmodule unterschiedlicher Standards in eine gekapselte, eingebettete Kommunikationsplattform (Abbildung 6.1). Der Funktionsumfang beschränkt sich im Rahmen dieser Arbeit auf grundlegende Mechanismen und prozedurale Abläufe für die Schaffung einer funkstandardübergreifenden, heterogenen Netzstruktur. Eine Analyse der Messergebnisse erlaubt außerdem die nachträgliche Bewertung der abstrakt definierten Kostenparameter innerhalb der simulierten Testszenarien. Dabei werden die gemessenen, funkstandardspezifischen Parameter der genutzten Hardwaremodule mit den Werten der SimANet Umgebung verglichen und entsprechend in Relation gesetzt.

6.1. Funktionsumfang und Einsatzbereiche

Der PLANet Prototyp (Abbildung 6.2) realisiert eine Gateway-Funktionalität für die funkstandardübergreifende Übertragung von Informationen auf Basis von IP. Im Gegensatz zu herkömmlichen Routing-Hardwarekomponenten erfolgt die Punkt-zu-Punkt-Weiterleitung der Daten nicht ausschließlich über eine Kommunikationstechnologie. Je nach verwendetem Funkmodul ist eine Übertragung auf unterschiedlichen Protokollebenen möglich. Die IP-Pakete werden dabei funkstandardabhängig als Nutzdaten in die jeweiligen Protokolle gekapselt und ggf. aufgeteilt. Entsprechend dem in Kapitel 4 vorgestellten Konzept werden eingehende Datenpakete auf relevante Information, primär Absender- und Empfängeradresse, überprüft. Sind die Daten nicht für den eigenen Host bestimmt, wird die zielgerichtete Weiterleitung mit Hilfe eines geeigneten Routingverfahrens angestoßen. Die LCU (siehe Kapitel 4.4.3) übernimmt dabei das Management der Verbindungskanäle und kontrolliert die integrierten Funkmodule. Für das Host-System mit Betriebssystem und Softwareapplikationen ist der von PLANet genutzte, physische Kommunikationskanal nicht sichtbar.

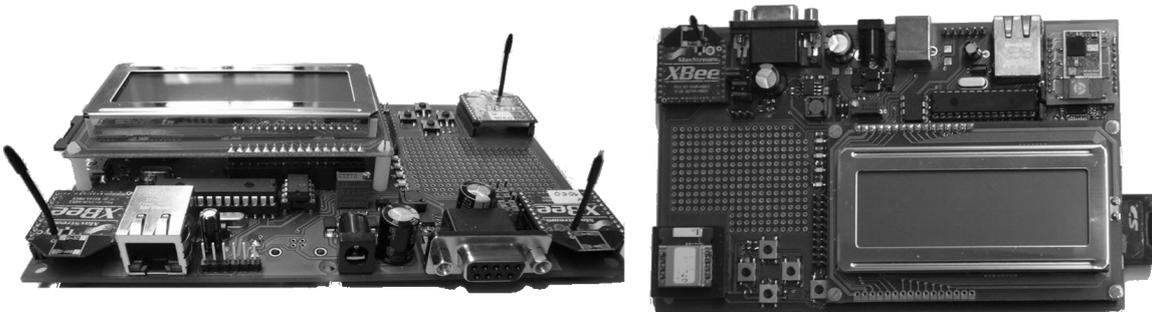


Abbildung 6.2.: Technische Umsetzung von PLANet als voll funktionsfähige, eingebettete Kommunikationsplattform.

Die Einsatzbereiche für den entwickelten Prototypen sind nicht auf Projekte aus dem Bereich der Funkstandardintegration beschränkt. Er dient gleichzeitig als Experimentalplattform für vielfältige Aufgaben im Bereich der Mikrocontrollerprogrammierung oder als Basishardware für unterschiedliche Sensornetzwerkanwendungen.

Für die Kommunikation stehen PLANet drei I/O-Sockel für Funkmodule beliebigen Standards zur Verfügung. Zusätzlich können über einen Ethernet-Port drahtgebundene Komponenten integriert werden. Der Anschluss der Prototypen-Plattform an das Host-System erfolgt über USB, wobei PLANet hier als Netzwerkkarte eingebunden wird. Die implementierten USB-Treiber realisieren diese Funktionalität sowohl auf Windows- als auch auf Linux-Systemen. Somit sind Testszenarien unkompliziert durchführbar, indem das USB-Netzwerkinterface von PLANet als Standardgateway definiert wird. Die anderen verfügbaren Netzwerkschnittstellen sind deaktiviert. In Folge dessen sendet das Host-System jeglichen Netzwerkverkehr applikationsunabhängig über PLANet, was zusätzliche Anpassungen auf Softwareebene unnötig macht.

Alternativ ist auch eine autarke Betriebsart ohne ein darüberliegendes Host-System möglich. Die Interaktion mit PLANet erfolgt hierbei über die seriellen Schnittstellen, welche gleichzeitig

auch als Debug-Konsole dienen. Zusätzliche Taster, LEDs und ein optionales Display dienen als frei definierbare Schnittstellen zur Informationsein- und Ausgabe. PLANet verfügt weiterhin über ein Programmier- und Debug-Interface auf Basis von *JTAG*¹. Über den Bus der integrierten *SPI Schnittstelle*² werden neben der Ethernet-Schnittstelle auch ein SD-Karten-Slot und ein nichtflüchtiger EEPROM³ angesteuert, welche zur Speicherung systemspezifischer Parameter und relevanter Loggingdaten genutzt werden.

Für die Erweiterung von PLANet in Hinblick auf neue Kommunikationsstandards können aktualisierte Firmware-Versionen sowohl über die SD-Card-Massenspeicher als auch optional über das Netzwerk [15, 19] aktualisiert und konfiguriert werden. Im Rahmen der Studienarbeit *Methoden zur modularen Erweiterung funkstandardübergreifender Hardware* [14] wurden mögliche Speicher- und Erweiterungskonzepte für zukünftige Entwicklungen analysiert und evaluiert.

6.1.1. Einschränkungen

Der realisierte Prototyp setzt spezielle Anforderungen an die zu integrierenden Funkmodule, was die Auswahl einschränkt. Neben physikalischen Restriktionen bzgl. der Abmaße und der vorhandenen Schnittstellen betrifft dies vor allem die verwendeten Protokolle. Sind entsprechende Module am Markt nicht verfügbar, kann der jeweilige Kommunikationsstandard nur sehr schwer in PLANet integriert werden. In Hinblick auf die Auswahl der Hardware ergeben sich weitere Einschränkungen bei den Leistungsparametern der Plattform. Dies betrifft primär die Datendurchsatzrate und die Übertragungslatenzen. Hierbei erreichen die verwendeten Funkmodule in vielen Fällen nicht die im Standard spezifizierten Übertragungswerte. Zugunsten einer einfachen Ansteuerung der Funkmodule erfolgt die externe Anbindung meist über eine serielle Schnittstelle, welche oftmals den begrenzenden Faktor für den Datendurchsatz repräsentiert.

Bedingt durch die begrenzten Schnittstellen und Pins des verwendeten Mikrocontrollers ist die maximale Anzahl integrierter Funkmodule begrenzt. Die Verarbeitungsgeschwindigkeit des Mikrocontrollers für die Paketkonvertierung und Weiterleitung bestimmt dabei den maximalen Datendurchsatz.

Desweiteren erhebt die Plattform in der aktuellen, prototypischen Form nicht den Anspruch, herkömmliche Kommunikationsgeräte des alltäglichen Gebrauchs, wie bspw. Mobiltelefone, PDAs oder Laptops vollständig zu integrieren. Es sei an dieser Stelle nochmals darauf hingewiesen, dass diese Integration konzeptionell durchführbar und sinnvoll ist. Die gezeigten Simulationen und der prototypische Aufbau im Rahmen von PLANet sollen daher die Möglichkeiten einer wirtschaftlichen und effizienten Realisierung verdeutlichen. Durch begrenzte Ressourcen konnte nur eine relativ kleine Netztopologie realisiert werden. Diese bietet nur wenig Spielraum für Routing-Szenarien. Neben den durchgeführten Simulationen wurden daher auf Basis von

¹Joint Test Action Group

²Serial Peripheral Interface

³Electrically Erasable Programmable Read Only Memory

6. Prototyp-Plattform PLANet

PLANet keine weiteren Untersuchungen hinsichtlich der Routing-Problematik durchgeführt. Der Funktionsumfang beschränkt sich zunächst auf die zielgerichtete Weiterleitung auf Basis ausgewählter Protokolle der verschiedenen Standards. Unter Verwendung der verfügbaren Hardware konnte ein zuverlässiger Betrieb ermöglicht werden.

6.2. Aufbau / Design

Während des Entwicklungsprozesses mussten zunächst grundlegende Designentscheidungen getroffen werden. Besonders wichtig war dabei die Auswahl geeigneter Hardwarekomponenten. Die Plattform soll ein möglichst breites Anwendungsfeld abdecken und hinreichend Flexibilität sowie großzügig dimensionierte Leistungsreserven gewährleisten. Wesentliche Vorarbeit und erste praktische Erfahrungen wurden im Rahmen von [16] gesammelt. Ergebnisse dieser Arbeit lieferten zunächst erste Softwarekomponenten sowie eine voll funktionstüchtige Experimentalplattform (Abbildung 6.3).

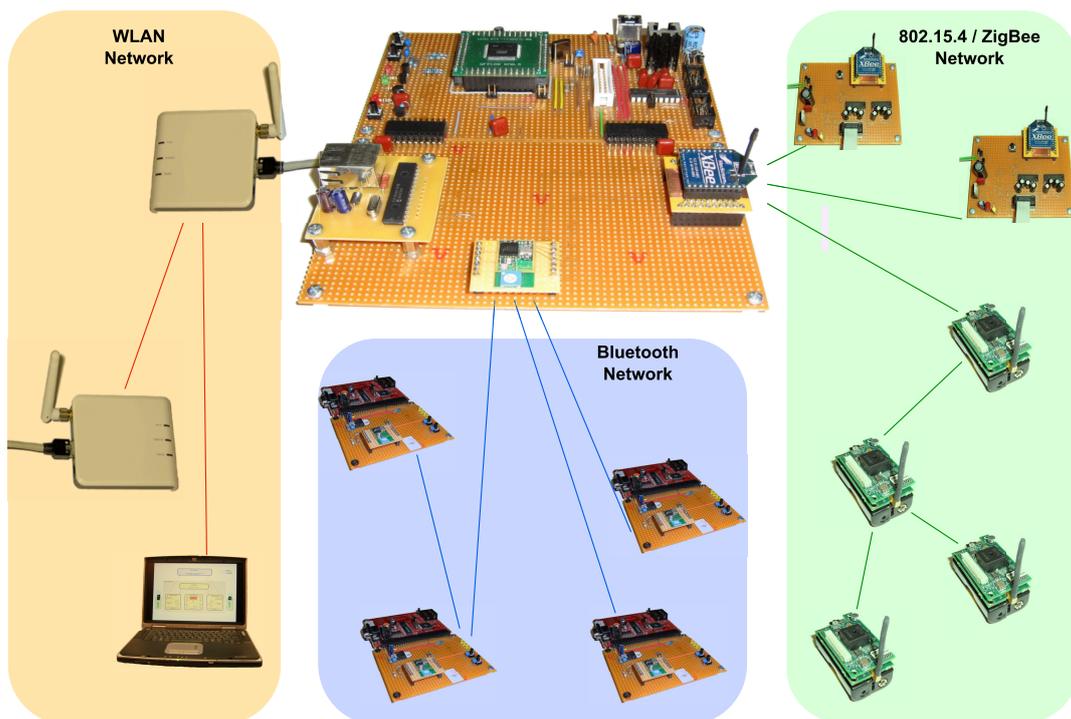


Abbildung 6.3.: Erste prototypische Umsetzung des Konzeptes. Der Aufbau ermöglicht erstmals die Kopplung 3er Funktechnologien (WLAN - IEEE 802.11b/g, ZigBee - IEEE 802.15.4, Bluetooth - IEEE 802.15.1) und somit auch die Kommunikation zwischen diesen Technologien auf unterschiedlichen Ebenen der jeweiligen Protokollstacks.

Auf Basis dieser Ergebnisse konnten nun Entscheidungen für die zentralen Hardware-Bauteile, das verwendete μ -Betriebssystem sowie nötige Treiber- und Softwarekomponenten für die

Mikrocontroller-Firmware getroffen werden.

Die daraufhin weiterentwickelte Prototyp-Plattform verfügt über drei universell nutzbare Sockelschnittstellen für beliebige Funkmodule. Das überarbeitete Design von PLANet kann nun in großen Stückzahlen gefertigt und für unterschiedliche Projekte zum Einsatz kommen (Abbildung 6.4). In Hinblick auf das vorgestellte, funkstandardübergreifende Kommunikationskonzept ist nun die Integration von drei unterschiedlichen Kommunikationsstandards in eine heterogene Netzstruktur möglich.

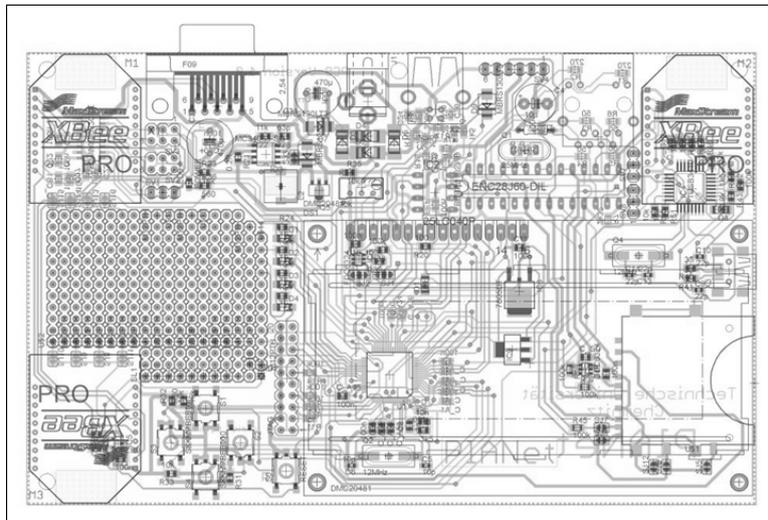


Abbildung 6.4.: Darstellung des PLANet Platinen-Layouts.

6.2.1. Hardware

Im folgenden Abschnitt soll auf die verwendete Hardware näher eingegangen werden. Dies betrifft neben der zentralen Berechnungseinheit vor allem die verwendeten Funkmodule. Abbildung 6.5 visualisiert den schematischen Aufbau der Plattform.

Mikrocontroller

Ursprünglich wurde das IFB Konzept für eine parallele Verarbeitung innerhalb eines FPGAs (feldprogrammierbare Logikbausteine) entwickelt. Daher erscheint der Einsatz eines FPGAs zunächst als naheliegend und sinnvoll. Dies trifft besonders in Hinblick auf den Aufbau der zu verbindenden Interfaces zu, wo die prozeduralen Abläufe der Protokolle direkt als endliche Automaten umgesetzt werden können. Die Transformationen während der Paketkonvertierung wären durch einfache Schaltungen realisierbar.

Dem gegenüber stehen allerdings einige Nachteile. Vor allem die Komplexität der zu integrierenden Funkmodule und deren Protokolle ist problematisch. Diese hebt sich deutlich von den bisher im Rahmen der IFB Forschungen adaptierten Schnittstellen ab. Desweiteren sind FPGAs kostenintensiver als herkömmliche Prozessoren / Mikrocontroller, was sich bei einer Nutzung

6. Prototyp-Plattform PLANet

in größeren Stückzahlen negativ auswirkt. In den einzelnen Kommunikationsstandards existieren außerdem mehrfach Abhängigkeiten zwischen prozeduralen und syntaktischen Spezifikationen. Diese Problematik ist mit Hilfe einer Implementierung auf Softwarebasis deutlich besser beherrschbar.

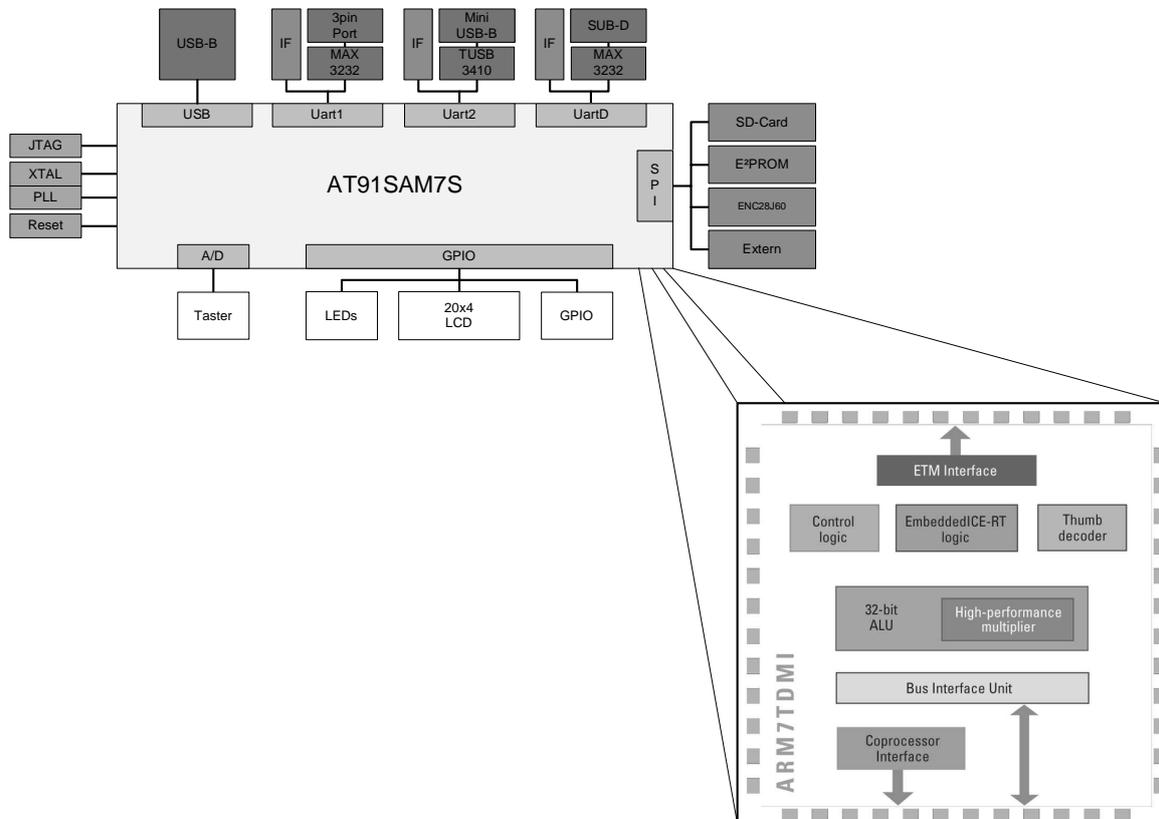


Abbildung 6.5.: PLANet mit ARM7 Mikrocontroller, den einzelnen Komponentengruppen sowie einer Detaildarstellung des μ C Kerns ARM7TDMI. Ein-/Ausgabeeinheiten unten, Speicherschnittstellen rechts, Konfigurationspins links, USB-Interface und Netzwerkschnittstellen oben.

Das ursprüngliche IFB Konzept legte besonderen Wert auf geringe Anforderungen bezüglich des Laufzeitsystems. Durch die Erweiterungen und Modifikationen des Konzeptes im Rahmen dieser Arbeit änderte sich jedoch der Fokus bezüglich dieser Anforderungen. In Hinblick auf eine Mikrocontroller-Umsetzung für die universelle Verwendung muss somit zunächst ein passender Controller gewählt werden. Für die Entwicklungen in einer Experimentalplattform nehmen Low-Power Eigenschaften nur eine untergeordnete Rolle ein. Hier sind andere Leistungsparameter von besonderem Interesse:

- groß dimensionierter RAM
- freidefinierbare Schnittstellen für:
 - Kommunikationseinheiten
 - externe Speichermedien
 - Anbindung an das Host-System

- Display zur Visualisierung von Statusinformationen
- gute Debug-Funktionalität
- schnelles und zuverlässiges Programmierinterface

Zum Einsatz kam ein Mikrocontroller der *ARM7* Baureihe (*AT91SAM7X256*). Der 32 Bit Prozessor der Firma Atmel stellt mit 64kByte Datenspeicher (RAM) und 256kByte Programmspeicher ausreichend Ressourcen für die Umsetzung des Konzeptes zur Verfügung. Er verfügt weiterhin über einen *DMA Controller*⁴ für die integrierten SPI und UART Schnittstellen. Somit können die angeschlossenen Funkmodule ohne Unterbrechungen des Prozessors direkt mit dem Speicher kommunizieren.

Kommunikationsschnittstellen

Für den realisierten Prototypen werden zunächst drei Kommunikationsstandards eingesetzt: *IEEE 802.11b/g (WLAN)*, *IEEE 802.15.4 / ZigBee* und *Bluetooth*. Für die Auswahl der genutzten Module waren vor allem die Verfügbarkeit sowie die Dokumentationsunterlagen entscheidende Kriterien. Die Auswahl der aufgeführten Modelle ist dabei keineswegs vollständig, sondern beschränkt sich auf aktuell verfügbare Hardware, welche für die Anwendung in einem funkstandardübergreifenden Brückenknoten geeignet ist.

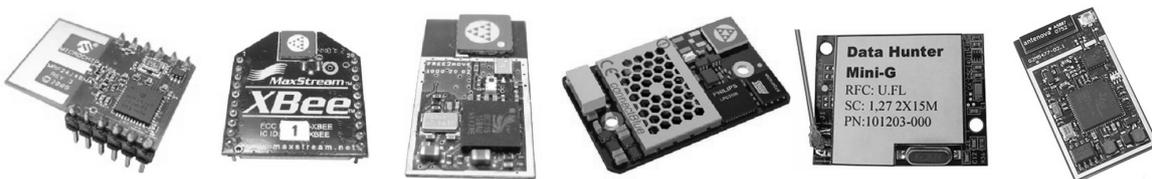


Abbildung 6.6.: Verfügbare Funkmodule unterschiedlicher Kommunikationsstandards für den PLANet Prototypen. Von links: MRF24J40(MA) von Microchip (IEEE 802.15.4); XBee (IEEE 802.15.4 / ZigBee); Free2Move F2M03GLA Bluetooth Modul; OWSPA311g von connectBlue (IEEE 802.11 / WLAN); Datahunter Mini-G (IEEE 802.11 / WLAN); RN-131G (WiFly GSX 802.11.G Module / WLAN).

IEEE 802.15.4 / ZigBee

Der IEEE 802.15.4 kompatible Chip *CC2420* von Texas Instruments benötigt nur eine minimale externe Beschaltung und integriert alle notwendigen Hardwarefunktionalitäten einschließlich CRC Prüfsummenberechnung, Paketpufferung oder Algorithmik für die Verschlüsselung. Die Interaktion mit dem Mikrocontroller erfolgt über eine SPI Schnittstelle. Das verwendete Protokoll ist sehr hardwarenah, wodurch ein direkter Zugriff auf die MAC Ebene notwendig ist. Der prototypische Einsatz im Rahmen des PLANet Projektes ist nur sehr eingeschränkt möglich, da der Umgang mit den nötigen Hochfrequenz-Bauteilen besondere Präzision hinsichtlich der Verarbeitung voraussetzt.

⁴Direct Memory Access

6. Prototyp-Plattform PLANet

Der von Microchip hergestellte *MRF24J40(MA)* IEEE 802.15.4 kompatible RF-Transceiver (Abbildung 6.6) ist dem vorgestellten CC2420 ähnlich und verfügt ebenfalls über ein SPI Interface mit einem Kommunikationsprotokoll auf unterster Ebene. Der Baustein ist als Modulvariante erhältlich, welche auf einer kleinen Platine alle externen Bauteile sowie die Antenne integriert. Probleme in Hinblick auf ein HF-taugliches Design entfallen folglich.

Ein dritter Vertreter ist das *XBee* Modul (Abbildung 6.6), welches im Gegensatz zu den anderen Vertretern über einen eigenen Mikrocontroller verfügt und über eine serielle Schnittstelle angesprochen wird. Der integrierte Controller implementiert einen Großteil der komplexen Protokolle auf den unteren Kommunikationsebenen. Für die Konfiguration der Protokolldetails stehen entsprechende AT-Kommandos zur Verfügung. Mit Hilfe einer API können Aktionen wie das Versenden und Empfangen von Paketen angesteuert werden. Aufbauend auf den unteren, IEEE 802.15.4 konformen Protokollebenen verfügen neuere Module je nach Ausführung über einen ZigBee Protokollstack. Aufgrund der Verfügbarkeit und dem weit verbreiteten Einsatz in unterschiedlichen Forschungsbereichen kamen die XBee Module in PLANet zum Einsatz.

Bluetooth

Die beiden zur Verfügung stehenden Bluetooth Module *F2M03AC2 / F2M03GLA* (Abbildung 6.6) der Firma Free2move verfügen zur Anbindung an den Mikrocontroller über serielle Schnittstellen. Das GLA Modul bietet außerdem volle *Enhanced Data Rate (EDR)* Unterstützung. Je nach Ausführung steht entweder eine standardkonforme HCI Schnittstelle oder eine "Wireless UART" Schnittstelle auf RFCOMM Ebene zur Verfügung. Hierbei beschränkt sich die Nutzung auf jeweils einen exklusiven Verbindungskanal pro Modul. Da aber im konkreten Einsatzszenario meist mehrere, simultane Verbindungen zu unterschiedlichen Knoten notwendig sind, kommt nur die HCI Variante für den Einsatz in PLANet in Frage. Um in diesem Fall jedoch eine standardkonforme Kommunikation zu ermöglichen, sind umfangreiche Protokollimplementierungen nötig. Hier müssen zwingend die Funktionalitäten der in Kapitel 4.4.3 vorgestellten LCU zum Einsatz kommen.

WLAN / IEEE 802.11

Das WLAN Funkmodul *OWSPA311g* von connectBlue (Abbildung 6.6) verfügt über eine AT-Kommando basierte UART Schnittstelle. Für den Betrieb steht leider nur eine Kommunikation auf TCP oder UDP Basis zur Verfügung. Dabei unterstützt das Modul Kommunikationskanäle für die Standards IEEE 802.11b/g sowie optional die Verschlüsselungsalgorithmen WEP64, WEP128 oder auch WPA-PSK bzw. WPA2-PSK.

Das *Datahunter Mini-G* Modul (Abbildung 6.6) ermöglicht im Gegensatz zur connectblue Lösung auch eine Kommunikation über IP, was speziell für die Kopplung an den xIFB zwingend notwendig ist. Neben IP, TCP und UDP stehen auch High-Level Protokolle wie SMTP, POP3, HTTP oder FTP zur Verfügung. Es können zehn simultane Kommunikationskanäle genutzt und verwaltet werden. Alle aktuellen Verschlüsselungsverfahren sind ebenfalls möglich. Die Verbindung zum Host erfolgt analog zum connectBlue Modul über eine serielle UART Schnittstelle, welche die maximale Datenübertragungsrate nach oben begrenzt. Beide Module erreichen hierbei zwischen 2,7 und 3,0 MBit/s (effektiv unidirektional ca. 1,5 MBit/s), was weit unter den

Referenzwerten der 802.11b bzw. 802.11g Spezifikation liegt. Im Rahmen einer prototypischen Umsetzung eignet sich dieses Modul hervorragend.

Auch das *RN-131G (WiFly GSX 802.11.G)* erfüllt durch geeignete Schnittstellen, wie SPI und UART / RS232 oder auch SDIO, sowie einem integrierten TCP/IP Stack alle Anforderungen für einen Einsatz in PLANet. Zusätzliche Dienste auf höherer Ebene wie bspw. ARP, DHCP, DNS oder auch ICMP sind ebenfalls implementiert. Die Interaktion mit dem Modul erfolgt über eine spezielle API und die Kommunikation beschränkt sich, ähnlich dem connectBlue Modul, auf TCP und UDP. Auch hier stehen aktuelle Verschlüsselungsalgorithmen bis hin zu WPA2 zur Verfügung und der effektive Datendurchsatz beträgt bis zu 4 MBit/s. Aufgrund der Verfügbarkeit wurde dieses Modul für den Einsatz in PLANet gewählt.

6.2.2. Funkmodulansteuerung / Verbindungsmanagement

Im laufenden Betrieb von PLANet kommen die ausgewählten Funkmodule zum Einsatz, welche ihrerseits über hardware-spezifische Befehlsätze und APIs verfügen. Hinsichtlich ihrer Ansteuerung sind die verbauten Module in zwei Kategorien zu unterteilen.

Funkmodule, die über einen proprietären Befehlssatz verfügen, kapseln funkstandardspezifische Kommunikationsabläufe nach außen in Form von abstrakten Systemrufen. In einem Konfigurationsmodus werden dazu alle relevanten Parameter definiert. Die auf den Modulen verbauten Controller-Hardware übernimmt daraufhin alle prozeduralen Abläufe des Verbindungsmanagements sowie die interne Verwaltung aller aktiven *Connection Handles*. Nach Abschluss der Konfiguration steht dem Nutzer im Übertragungsmodus ein gültiger und aktiver Kommunikationskanal zur Verfügung, der keinen weiteren administrativen Aufwand erfordert. Diese Kategorie wird im Rahmen des EAN Konzeptes und der praktischen Umsetzung in Form von PLANet favorisiert.

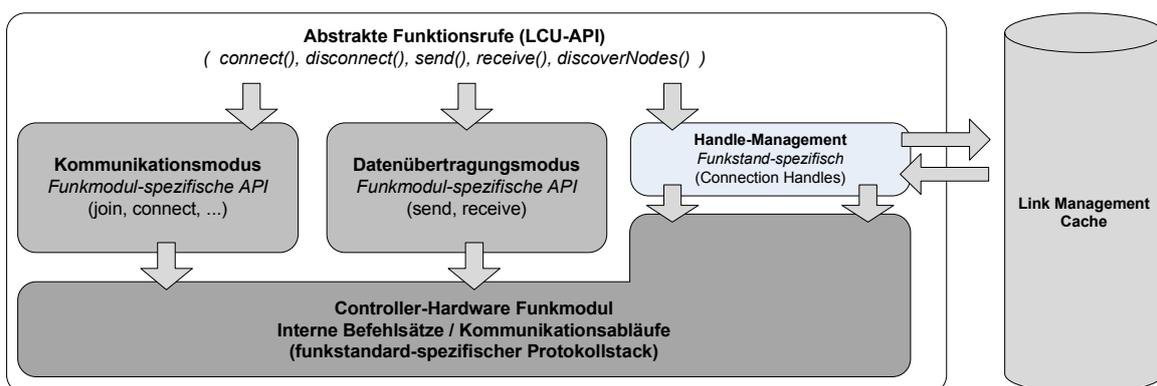


Abbildung 6.7.: Angepasster, hybrider Lösungsansatz für die Integration der Funkmodule.

Einige Module verfügen aufgrund komplexer Protokollstacks jedoch nur über Schnittstellen auf Basis bestimmter Protokollebenen. Dementsprechend müssen hier protokollspezifische Abläufe beachtet werden. Ein konkretes Beispiel stellen die verwendeten Module des Bluetooth Stan-

6. Prototyp-Plattform PLANet

dards dar, bei denen ausschließlich eine Kommunikation auf HCI Ebene in Frage kommt. Alternative Firmwareversionen auf Basis einer emulierten, seriellen Verbindung existieren zwar, diese beschränken sich jedoch auf einen exklusiven, fest konfigurierten Kommunikationskanal, was den effizienten Einsatz in einer dynamischen Netzwerkstruktur unmöglich macht.

Aus diesem Grund muss das Integrationskonzept für die Ansteuerung der Funkmodule erweitert und an die Rahmenbedingungen der verwendeten Hardware angepasst werden.

Abbildung 6.7 verdeutlicht das generische Integrationskonzept weiterer Funkmodule. Die Anbindung an den *Link Management Cache* der LCU ermöglicht eine zentrale Verwaltung generierter Connection Handles, falls diese nicht funktmodul-intern verwaltet werden. Bei der prototypischen Umsetzung erhöht dies den Freiheitsgrad bezüglich der Hardwareauswahl erheblich. Das System ist somit nicht ausschließlich an Module mit einer gekapselten, abstrakten Befehls-API gebunden.

Daraus resultierend werden die verbauten ZigBee und WLAN Module nach dem favorisierten, funktmodulspezifischen Konzept über abstrakte Funktionsrufe angesteuert, welche anschließend auf hardware-spezifische Befehlsrufe für Konfiguration und Datenübertragung abgebildet werden.

Im Gegensatz dazu kann das Bluetooth-Modul alternativ auf Basis von HCI Kommandos angesteuert werden. Dabei wird jeder Verbindungskanal durch ein Connection Handle repräsentiert. Über eine Verwaltungsinstanz müssen diese dem *Link Management Cache* übergeben und den entsprechenden Hardware-Adressen zugeordnet werden. Im Falle einer erneuten Kommunikationsanfrage erfolgt zunächst die Suche nach einem passenden Eintrag im Link Management Cache. Existiert ein aktives Handle, steht ein gültiger Kommunikationskanal ohne Verzögerungen zur Verfügung.

Obwohl das HCI-Protokoll sehr komplex und umfangreich ist, nutzen die implementierten Automaten für das Verbindungsmanagement einen relativ überschaubaren Befehlssatz und wenig komplexe Abläufe für die Initialisierung einer Verbindung, was den Umgang dementsprechend vereinfacht. Allerdings verhindern die hardwareseitigen Restriktionen der genutzten Bluetooth Module an dieser Stelle Verbindungen auf höheren Protokollebenen. Diese Funktionalität muss manuell auf Basis von HCI implementiert werden. Aufgrund der hohen Komplexität wurde die Kommunikation zunächst auf einen Datenaustausch über HCI beschränkt.

6.2.3. Laufzeitsystem / Software

Die Umsetzung der Funktionalität von PLANet umfasst zum einen die Ansteuerung der einzelnen Hardwarekomponenten sowie die Realisierung der prozeduralen Abläufe des vorgestellten Konzeptes in Form von Softwaremodulen auf dem Mikrocontroller. Auf Basis einer speziellen Adapterklasse werden dazu alle Systemrufe des xIFB Kerns auf das Laufzeitsystem abgebildet. Die einzelnen softwareseitigen Teilaspekte sollen im Folgenden näher erläutert werden.

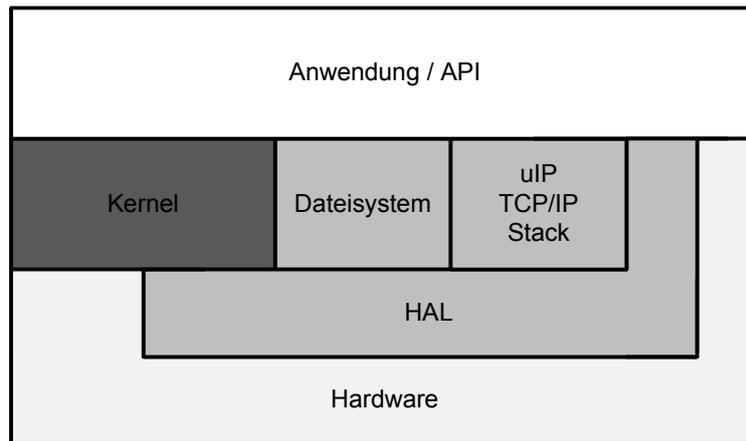


Abbildung 6.8.: Darstellung der einzelnen Verarbeitungsebenen von PLANet. Auf die Hardware greift sowohl der *HAL*⁵ als auch der Kernel zu. Dateisystem und Netzwerk-Protokollstack nutzen die Funktionen der HAL. Auf oberster Abstraktionsebene befinden sich die Softwareapplikationen innerhalb des PLANet Systems.

Betriebssystem

Für die Umsetzung der Software-Funktionalität auf dem Mikrocontroller steht neben einer nativen Programmierung auch die Nutzung eines entsprechenden Betriebssystems zur Auswahl. Dies hat den entscheidenden Vorteil, dass bereits die meisten Rahmenbedingungen der Laufzeitumgebung durch das Betriebssystem zur Verfügung stehen. Dazu gehören vor allem eine Multiprozess- / Multithread-Unterstützung einschließlich essentieller Konstrukte zur Prozesssynchronisation wie Semaphoren oder Mutex, eine dynamische Speicherverwaltung sowie Queues und Events. In Folge dessen konzentriert sich die Programmierung auf die Umsetzung der funktionalen Aspekte des Projektes. Erkauft wird sich dieser Komfort durch zusätzliche Ressourcen für die Funktionalität des Betriebssystems, was primär den genutzten RAM Speicher betrifft. Weiterhin hat der Nutzer fortan keine volle Kontrolle über den Controller und dessen Funktionalitäten, da das Betriebssystem den Zugriff steuert und die Ressourcen verwaltet. Betrachtet man jedoch die Komplexität des Projektes, überwiegen die Vorteile und der Einsatz erscheint zwingend erforderlich.

Zahlreiche verfügbare Mikrocontroller-Betriebssysteme sind jedoch an bestimmte Hardware gebunden, was den Einsatz in der PLANet Plattform unmöglich macht. Dazu gehört bspw. das vielseitig eingesetzte *TinyOS* [185], welches ausgereifte Konzepte und APIs integriert, sehr gut dokumentiert und speziell auf WSN-Applikationen zugeschnitten ist. Trotz der Unterstützung vieler Hardwareplattformen und einer stetigen Weiterentwicklung durch ein internationales Konsortium existiert bisher keine Portierung für den ARM7. Desweiteren ist die verfügbare Codebasis sehr komplex und die Einarbeitung in die verwendeten Konzepte erfordert einen hohen Zeitaufwand. *Contiki* [58] vereint mehrere unabhängige Konzepte in einem leichtgewichtigen Betriebssystem für ressourcenbeschränkte, eingebettete Systeme. Es bietet

⁵Hardware Abstraction Layer

6. Prototyp-Plattform PLANet

ein gewohntes Programmiermodell mit kooperativem oder präemptivem Multithreading und einem ereignisbasierten Ausführungsmodell. Ein TCP/IP-Stack ist bereits integriert und es stehen vielfältige Kommunikationsprimitive zur Verfügung. Die Anzahl der unterstützten Hardware ist leider nur gering und eine Portierung auf dem ARM7 ist bisher nicht verfügbar. *Mantis OS* [116] ist ein weiteres Mikrocontroller-Betriebssystem. Aufgrund der schlechten Dokumentation, dem recht hohen Speicherverbrauch (primär durch mehrere Kontexte und den dazugehörigen Stacks) und einem ungewohnten Buildsystem wurde dieser Vertreter bei der Auswahl für PLANet vernachlässigt.

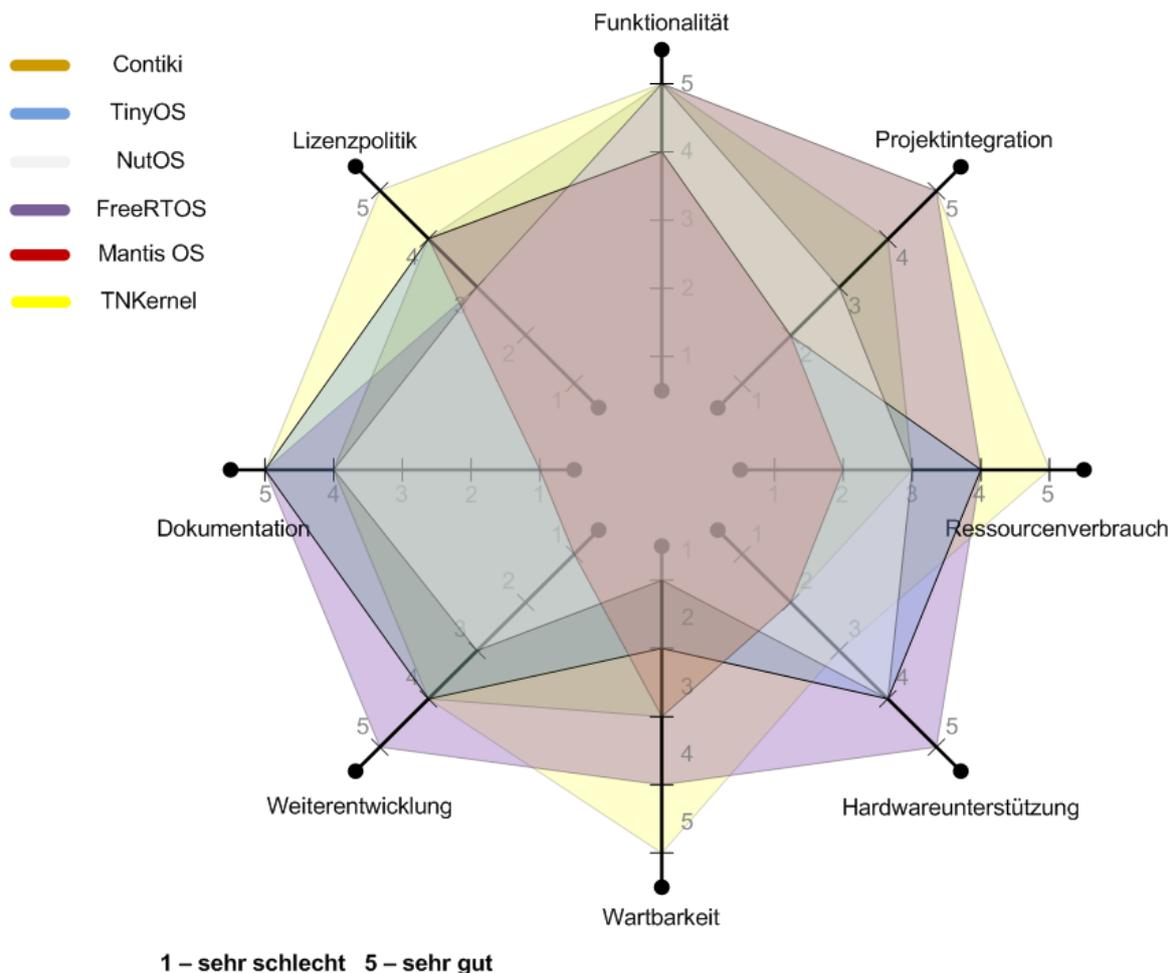


Abbildung 6.9.: Bewertung aktuell verfügbarer Betriebssysteme für ressourcenbeschränkte, eingebettete Systeme. Als Bewertungskriterien dienen neben der reinen Funktionalität und dem Ressourcenverbrauch auch Parameter wie Wartbarkeit, Dokumentation, Aktivität (Weiterentwicklung), Hardwareunterstützung oder Projektintegration. Auch die an das Betriebssystem gebundenen Lizenzen mit den damit verbundenen Einschränkungen sind von Relevanz.

Nut/OS [59], ein weiteres echtzeitfähiges Betriebssysteme für eingebettete Systeme mit dem Fokus auf Netzwerkkonnektivität, ist für den ARM7 Mikrocontroller verfügbar und bietet sehr viele einzelne Komponenten, welche modular in Form von Bibliotheken eingebunden werden. Dazu gehört auch ein TCP/IP Stack, eine dynamische Speicherverwaltung, die priorisierte Behandlung von Ereignissen oder ein kooperatives Multithreading. Ein Konfigurator hilft

bei der Zusammenstellung und bei evtl. auftretenden Konflikten. Doch die Masse von Einzelkomponenten führt dabei zu einer kaum überschaubaren Softwarestruktur, was eine Wartung und nachträgliche Modifikation deutlich erschwert. *FreeRTOS* [30] ist, wie aus der Bezeichnung ersichtlich, ein Open Source Projekt für ein Echtzeitbetriebssystem im Bereich eingebetteter Systeme. Im Gegensatz zu anderen Softwareplattformen ist FreeRTOS für nahezu jede Mikrocontroller-Architektur verfügbar und wird ständig weiterentwickelt.

Die Wahl fiel jedoch auf *TNKernel* [179] (siehe Abbildung 6.9), ein leistungsfähiges und flexibel einsetzbares Mikrocontrollerbetriebssystem, welches alle wesentlichen Funktionalitäten unterstützt. Es wurde speziell auf und für die verwendeten Mikrocontroller entwickelt und steht unter einer komplett freien Open Source Lizenz. Zu den wichtigsten Eigenschaften von TNKernel gehören die Echtzeitfähigkeit, ein präemptives, prioritätenbasiertes Multithreading sowie viele essentielle Konstrukte wie bspw. Semaphoren, Queues oder Events. TNKernel belegt minimale Ressourcen und ist sehr kompakt konzipiert, was eine Wartung und Erweiterung deutlich vereinfacht. Eine komplette Portierung für den ARM7 Core existiert bereits. Somit sind nur wenige Anpassungen an die Hardware nötig. TNKernel bietet aufgrund des verwendeten, FreeBSD-ähnlichen Lizenzmodells alle Freiheiten im Zuge von Weiterentwicklungen und Modifikationen am Systemkern.

Innerhalb von PLANet übernimmt das System die gesamte Ressourcenverwaltung einschließlich des Scheduling und der Ein- und Ausgabeverwaltung mittels entsprechender Treiber. Die zentralen Anpassungen innerhalb und aufbauend auf TNKernel für den PLANet Hardware umfassen dabei:

- Mapping der Peripherie des Mikrocontrollers auf die Pins
- Einbindung der Treiber / Konfiguration für:
 - UART, LCD, SPI, Display, Taster, LEDs, Ethernet-Controller, SD-Karten-Slot, EEPROM
- Konfiguration des CPU-Taktgebers
- Entwicklung API / HAL

Speicherkonzepte und Dateisystem

PLANet stehen zur Datenspeicherung unterschiedliche Speicherkonzepte zur Verfügung. Primär kommt zur Laufzeit der flüchtige Speicher des Mikrocontrollers zum Einsatz (*RAM*⁶). Hier können alle Variablenwerte und Einstellungen temporär abgelegt werden. Als zentraler, nicht flüchtiger Speicher für das eigentlich Programm dient der *Flash-ROM*⁷, welcher bei der Programmausführung sequenziell ausgelesen wird. Auch der *EEPROM* - ein elektrisch löschbarer, programmierbarer Nur-Lese-Speicher - ist nicht flüchtig und folglich bleiben alle Daten bei

⁶Random Access Memory - Speicher mit wahlfreiem Zugriff

⁷Flash Read Only Memory - rücksetzbarer Nur-Lese-Speicher

6. Prototyp-Plattform PLANet

einem Zusammenbruch der Spannungsversorgung erhalten. Er kann deutlich häufiger als der Flash wiederbeschrieben werden, ist jedoch deutlich langsamer. Er dient vorrangig der Speicherung von initialen Einstellungen und Parametern.

Zur Speicherung relevanter Messdaten oder auch Logging- oder Debug-Informationen reichen die genannten Speicherkonzepte jedoch nicht aus. Hierfür steht ein Massenspeicher in Form einer SD-Karte zur Verfügung, welcher genügend Kapazität zur Verfügung stellt. Für die korrekte und standardkonforme Verwaltung der Daten muss ein Dateisystem realisiert werden, welches losgelöst von der hardwareseitigen Ansteuerung des SD-Card Slots arbeitet. Die eingesetzte Implementierung unterstützt dabei FAT12, FAT16 sowie FAT32, wobei lange Dateinamen nicht genutzt werden können, und ist durch die gekoppelten Lizenzen uneingeschränkt frei verwendbar. Das Dateisystem benötigt je nach Funktionsumfang 3-10KByte ANSI-C Code, statisch 554 Byte RAM und je geöffneter Datei nochmals dynamisch 550 Byte.

Netzwerk-Protokollstack

Als Basis für die Kommunikation aller drahtlosen Schnittstellen dient die TCP/IP Protokollfamilie. Hierfür existieren mehrere Open Source Projekte, welche in den PLANet Prototyp integriert werden können. Der verwendete, lizenzfreie TCP/IP-Stack ist voll funktionsfähig und unterstützt neben TCP und UDP auch zusätzliche Protokolle wie bspw. ARP oder ICMP. Die C-Implementierung benötigt nur 2-4 KByte Code und ist mit 2 KByte RAM-Verbrauch sehr ressourcensparend.

6.2.4. Erweiterbarkeit

Sowohl die Hardwarekomponenten als auch die genutzten Softwaremodule erlauben eine unkomplizierte Erweiterung mit neuen Funktionalitäten. In Hinblick auf die Hardware betrifft dies im Wesentlichen die Möglichkeit, Funkmodule der verfügbaren Slot-Schnittstellen auszutauschen bzw. um neue Kommunikationstechnologien zu erweitern. In diesem Zusammenhang müssen innerhalb der Firmware Änderungen und Erweiterungen erfolgen, um die prozeduralen Abläufe der jeweiligen Protokolle korrekt abarbeiten zu können. Dies betrifft primär nicht das verwendete Betriebssystem sondern lediglich den funktionalen Kern der Multistandard-Gateway-Applikation. Für den universellen Einsatz von PLANet können die Kommunikationsslots für die Funkmodule auch als herkömmliche IO-Schnittstellen eingesetzt werden. Andere Projekte nutzen die Plattform im Bereich drahtloser Sensornetzwerk-Applikationen, in denen die Schnittstellen die benötigte Sensorik aufnehmen [15]. Hierfür kommt eine angepasste Laufzeitumgebung und entsprechende Treiber für die Sensormodule zum Einsatz. Da alle Softwarekomponenten lizenzrechtlich keine Einschränkungen verursachen, ist auch die Weiterentwicklung und Modifikation des Betriebssystemkerns, des Dateisystems oder des Netzwerkstacks unproblematisch realisierbar. Der genutzte ARM7 Mikrocontroller bietet ausreichend Leistungsreserven für einen Großteil der möglichen Anwendungsszenarien.

6.3. Anwendungsszenarien

Nachdem die Prototyp-Plattform näher vorgestellt wurde, soll nun auf dieser Basis die grundlegende Funktionalität eines EAN-konformen Systems unter Realbedingungen untersucht werden. Im Fokus der Betrachtungen steht dabei eine Multihop Kommunikation mit einem Protokollkonvertierungsprozess während der Übertragung. Die dabei zusätzlich entstehenden Verzögerungszeiten und die Stabilität des EAN Kerns repräsentieren hier wesentliche Faktoren. Im weiteren Verlauf soll außerdem der Leistungsbedarf des PLANet Prototypen und der einzelnen Komponenten untersucht werden.

6.3.1. Testszenario I: Multi-Standard, Multi-Hop Kommunikation

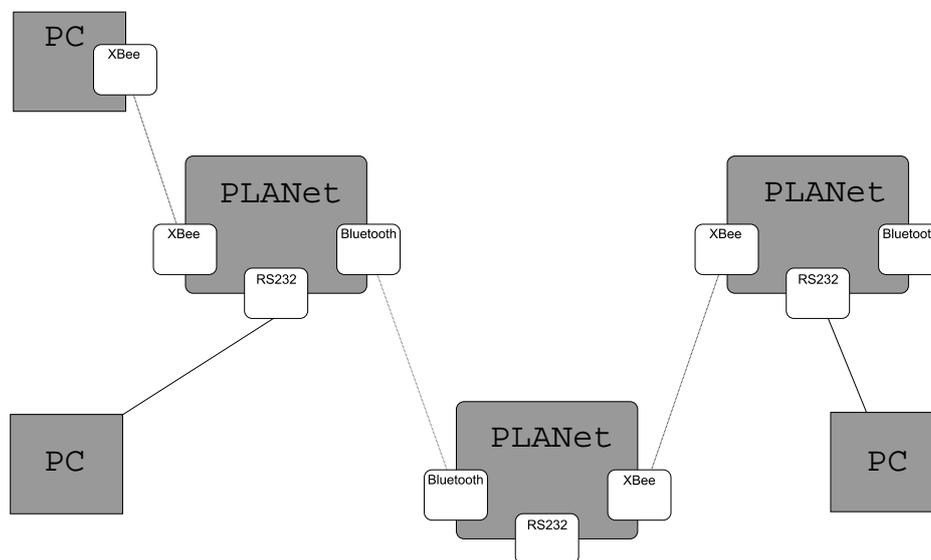


Abbildung 6.10.: Funktionstest des PLANet Prototypen in Form eines Testszenarios mit heterogener Netzwerkstruktur. Ein PC ist ausschließlich über ein XBee Modul mit dem Netzwerk verbunden. Die restliche Netzstruktur verfügt über unterschiedliche drahtlose Kommunikationsschnittstellen und zusätzliche Debugschnittstellen mittels RS232.

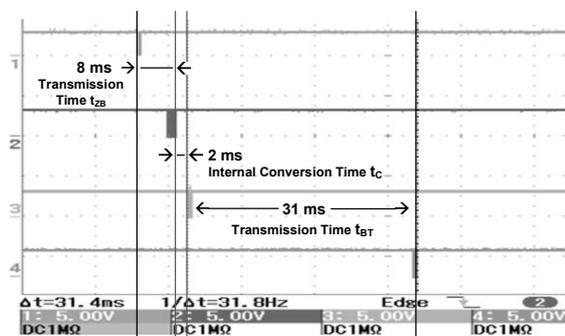
Abbildung 6.10 zeigt den schematischen Aufbau eines ersten funktionalen Tests. Der PLANet Prototyp wird hier innerhalb eines statischen Netzwerkszenarios mit mehreren integrierten Funkstandards auf seine grundlegenden Funktionen hinsichtlich der Protokollkonvertierung getestet. Die zu übertragenden Daten können dabei an verschiedenen Stellen in der Konfiguration abgegriffen und überwacht werden. Die einzelnen Knoten verfügen über eindeutige Adressen innerhalb des Netzwerkes, welche unabhängig von den Funkmodulen auf einer höheren Ebene definiert wurden. Die Paketübertragung wird jeweils von den angeschlossenen PCs initiiert. Dabei kann aus unterschiedlichen Protokollen gewählt und der Zielknoten definiert werden. Hardwareseitig kamen für die Funkübertragung die bereits vorgestellten XBee Module sowie die Free2move Bluetooth Module zum Einsatz. Aufgrund der aufwendigen Kommunikationspro-

6. Prototyp-Plattform PLANet

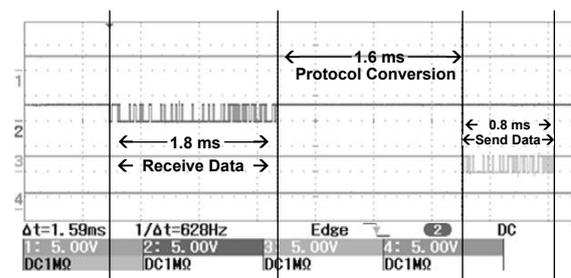
tolle des Bluetooth-Standards wurde für die entsprechenden Abschnitte auf vorkonfigurierte Verbindungskanäle auf Basis von HCI zurückgegriffen. Um höhere Schichten wie L2CAP oder dedizierte Dienstprofile nutzen zu können, wäre ein erheblicher Mehraufwand auf der softwareseitigen Implementierung auf dem Mikrocontroller notwendig gewesen. Im hier realisierten Testaufbau sind während der Paketkonvertierung bereits unterschiedliche Elemente der Datenpakete automatisch durch den xIFB zu generieren oder zu modifizieren. Dazu gehört bspw. die Länge der Nutzdaten, Checksummen, Delimiter oder auch Escape-Sequenzen.

Im Rahmen der durchgeführten Tests konnten die Anforderungen erfüllt werden. Im Normalbetrieb erhöhen sich die Latenzen durch eine Protokollkonvertierung mit rund 2 ms nur marginal. Auf die Einbindung von WLAN musste im Testaufbau verzichtet werden, da kompatible Module nicht verfügbar waren und sich eine Konfiguration herkömmlicher WLAN-Router über die Ethernet-Schnittstelle als nicht praktikabel erwies. Um Aussagen hinsichtlich der Übertragungslatenzen treffen zu können, wurde daher ein separater Testaufbau auf Basis von WLAN realisiert und analysiert. Die gemessenen Werte fließen dabei in die Diagrammdarstellungen mit ein.

Abbildung 6.11 zeigt eine durchschnittliche Übertragungsdauer von ca. 40 ms bei einer Multihop-Kommunikation des linken und rechten PLANet Knotens (siehe Abbildung 6.10). Dabei wird ein Großteil der Zeit für die Bluetooth-Kommunikation benötigt, was mit hoher Wahrscheinlichkeit aus der Komplexität der Protokollstacks resultiert. Die PLANet-interne Protokollkonvertierung nimmt mit durchschnittlich 2 ms nur einen Bruchteil der Übertragungszeit ein.



(a) Messungen während einer Multihop-Kommunikation mit drei PLANet-Knoten. Die einzelnen Messpunkte repräsentieren die Phasen: (1) - Datenübertragung Quelle → PLANet Knoten 1 (ZigBee); (2) - PLANet-interne Protokollkonvertierung ZigBee → Bluetooth; (3) - Datenübertragung PLANet Knoten 1 → PLANet Knoten 2 (Bluetooth); (4) - Empfang Datensenke (Bluetooth).



(b) Detaildarstellung des Protokollkonvertierungsprozesses innerhalb eines PLANet-Knotens. Anhand durchgehender Messergebnisse von unter 2 ms sind diese zusätzlichen Verzögerungen auch bei längeren Routenpfaden minimal.

Abbildung 6.11.: Messungen der Paketlaufzeit mit Hilfe eines Oszilloskops. Ungenauigkeitsfaktoren durch das Betriebssystem, speziell deren Schedulingmechanismen, konnten somit umgangen werden. Nutzdatenlänge 10 Byte.

Bluetooth bzw. die Anbindung des Bluetooth-Moduls bremst die Übertragung in diesem Kom-

munikationsszenario erheblich aus. Dies zeigt auch die nachfolgende Abbildung 6.12.

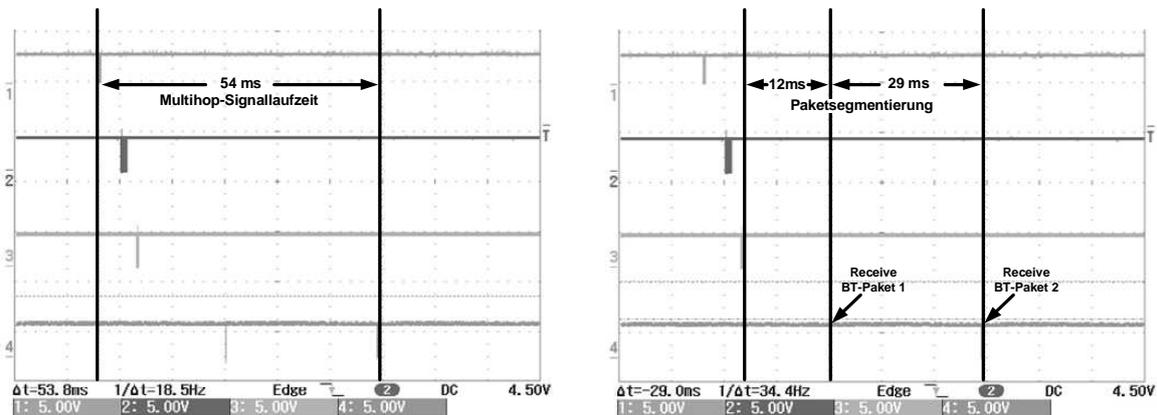


Abbildung 6.12.: Multihop-Übertragung unter ungünstigen Umständen. Während der Kommunikation über Bluetooth werden die Nutzdaten auf zwei Bluetooth-Pakete aufgesplittet. Somit erhöht sich die Gesamtübertragungsdauer deutlich um 29 ms auf insgesamt 54 ms.

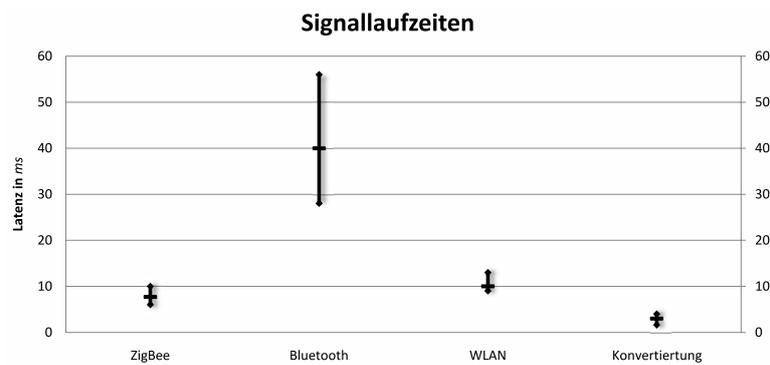


Abbildung 6.13.: Signallaufzeiten während der Datenübertragung eines Paketes auf Basis unterschiedlicher Kommunikationsstandards sowie die Latenzzeiten in Folge von Protokollkonvertierungen innerhalb des EAN Systems. Das Diagramm visualisiert dabei die Messstreuungen von 25 Messwerten inkl. Mittelwert, Maximum und Minimum. Nutzdatengröße: 25 Byte

Durch die Aufteilung der Nutzdaten in zwei Pakete verlängert sich die Übertragungsdauer um zusätzliche 29 ms. Eine mögliche Fehlerursache kann dabei die Anbindung des Bluetooth-Moduls an den EAN Kern darstellen. Die serielle Schnittstelle erlaubt dabei eine maximale Datenrate von 115 KBit/s, wohingegen die theoretische Datenrate für die Bluetooth-Übertragung bis zu 1 Mbit/s ermöglicht. Dieser Geschwindigkeitsunterschied lässt darauf schließen, dass das Modul nach einem definierten Timeout alle bis dahin anliegenden Nutzdaten in einem Paket kapselt und in einem freien Zeitschlitz übermittelt. Da der Datenstrom verhältnismäßig langsam in die Bluetooth-Module einfließt und Hardware keinerlei Wissen über die Menge noch anstehender Daten hat, erscheint dieses Verhalten aus Sicht der Funkmodule nachvollziehbar. Desweiteren kann auch das Verbindungsmanagement, bspw. durch Handshaking-Prozesse, den Kommunikationsverlauf stören und eine Segmentierung der Daten provozieren. Abweichungen der einzelnen Paketlaufzeiten sind auf den Medienzugriff zurückzuführen und nicht zu beeinflussen.

6. Prototyp-Plattform PLANet

Die in Diagramm 6.13 dargestellten Messstreuungen repräsentieren mehrere Multihop-Übertragungen innerhalb des vorgestellten PLANet Testszenarios. Im Vergleich zum gesamten Übertragungsprozess fallen die Verzögerungen durch EAN-interne Konvertierungsprozesse zwischen den jeweiligen Kommunikationsstandards gering aus. Die Übertragungslatenzen von WLAN basieren auf einem zweiten Testaufbau. Zeitmessungen für den Konvertierungsprozess erfolgten aus diesem Grund nur von ZigBee nach Bluetooth und vice versa.

6.3.2. Testszenario II: Stabilität

Nachdem die Funktionalität einer funkstandardübergreifenden Multihop-Kommunikation evaluiert und analysiert wurde, soll nun die Stabilität der EAN Konvertierungskomponente betrachtet werden. Dazu wird während einer kontinuierlichen Steigerung der Datendurchsatz die Latenz für die Protokollkonvertierung gemessen. Aufgrund der seriellen Anbindung der Funkmodule an PLANet ist der maximale Datendurchsatz auf 115KBit/s begrenzt. Betrachtet man jedoch das Testszenario, erfolgt die Konvertierung zwischen dem ZigBee Standard (präzise 802.15.4) und Bluetooth. ZigBee als Low-Power, Low-Distance Funkstandard besitzt hierbei die geringste Datenrate. Diese liegt bei 250KBit/s brutto und bidirektional. Für die Kommunikation in eine Richtung und unter Berücksichtigung des Protokolloverheads verbleiben $\leq 120\text{KBit/s}$ Netto-Datenrate. Somit erweisen sich die möglichen 115KBit/s (ca. 14 KByte/s) als vollkommen ausreichend, um eine Maximalauslastung für die Konvertierung von Bluetooth nach ZigBee und vice versa zu emulieren. Die Messreihen wurden mit Hilfe des Oszilloskops durchgeführt und ermöglichen die Aufzeichnungen aller Konvertierungsprozesse innerhalb des EAN Kerns (Abbildung 6.14).

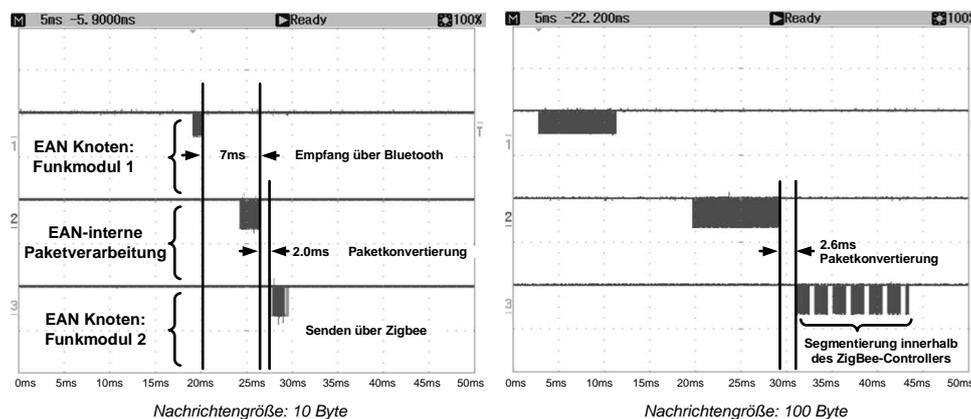


Abbildung 6.14.: Messungen der Paketkonvertierungszeit. Konvertierung Bluetooth \rightarrow Zigbee. Die einzelnen Phasen der Verarbeitung sind gut erkennbar (links). Rechts ein Konvertierungsprozess mit automatischer Paketsegmentierung durch den Controller des Funkmoduls.

Interessant ist hierbei besonders die Paketsegmentierung bei zu großer Nutzdatenlänge. Die Funktionalität wird hier in den integrierten Controllern der Funkmodule gekapselt und entlastet den EAN Kern von PLANet. Das Verbindungsmanagement stellt gültige Kommunikationspfade sicher. Die erzielten Durchschnittswerte aus 10 Durchläufen in Diagramm 6.15 zeigen kontinu-

ierlich geringe Schwankungen in der Konvertierungszeit über den gesamten Messbereich.

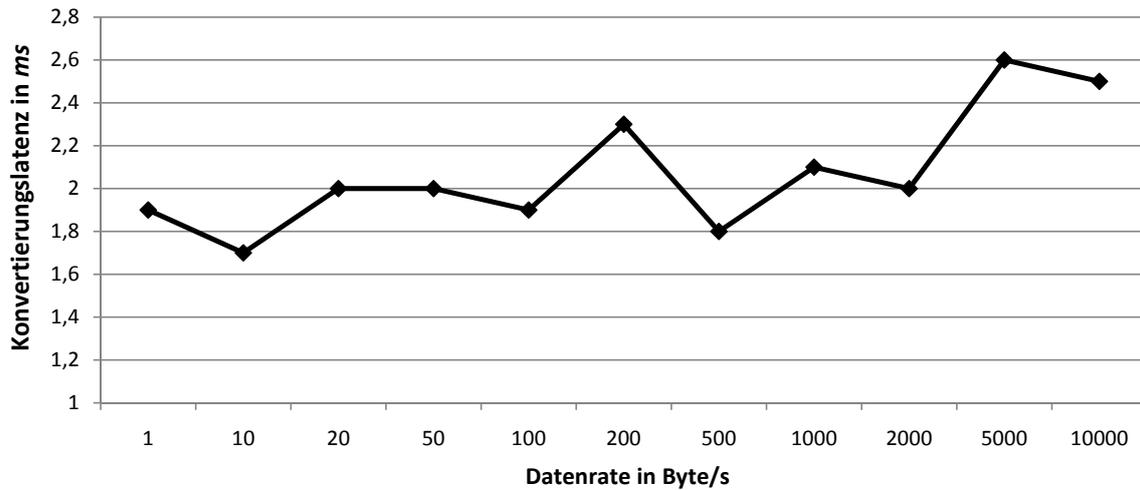


Abbildung 6.15.: Abhängigkeit der Konvertierungszeiten von der Datenübertragungsrate. Das Diagramm zeigt dabei eine kontinuierliche Kurve mit gemessenen Durchschnittswerten. Konvertierung ZigBee → Bluetooth

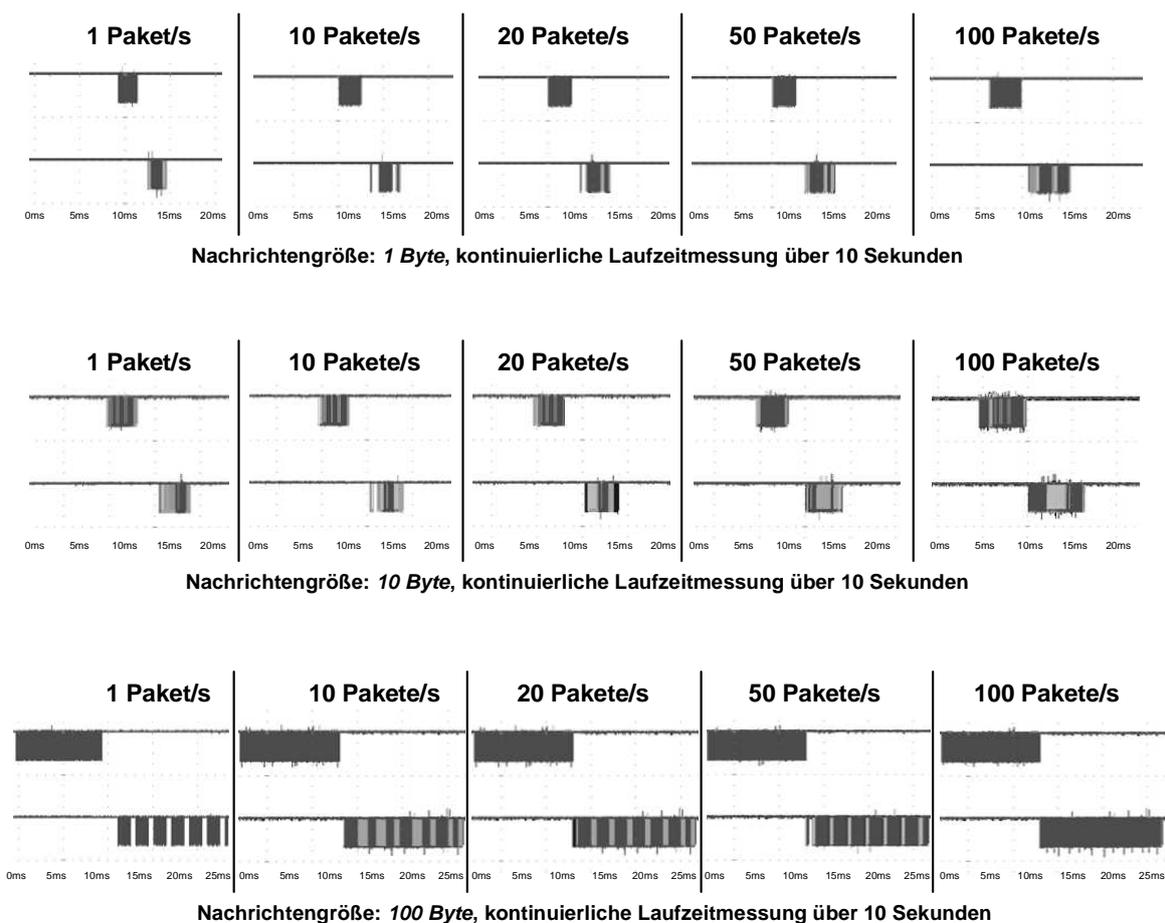


Abbildung 6.16.: Übersicht über die aufgenommenen Messkurven auf dem Oszilloskop. Die Persistenzmessungen ermöglichen dabei die Visualisierung mehrerer Einzelmessungen in einer überlagerten Darstellung.

6. Prototyp-Plattform PLANet

Abbildung 6.16 zeigt dazu passend die durchgeführten Persistenzmessungen auf dem Oszilloskop mit unterschiedlichen Paketgrößen und Übertragungsintervallen. Die gezeichneten Flankendiagramme repräsentieren dabei mehrere aufeinanderfolgende Messungen, welche in einer Darstellung überlagernd visualisiert werden. Die maximal auftretenden Schwankungen sind somit erkennbar. Die Messreihe mit 100 Byte Nachrichten erweckt hier den Anschein, dass die interne Konvertierungszeit gegen null tendiert. Dies ist jedoch nicht der Fall. Die Überlagerungen mehrerer Messkurven verursachen durch leicht erhöhte Schwankungen diese Darstellung.

Diagramm 6.17 zeigt die detaillierte Auswertung einzelner Messreihen. Die Abweichungen von Minimal- und Maximalwerten sind mit unter 1ms zu vernachlässigen. Auch unter Last ist nur ein minimaler Einbruch der Konvertierungszeiten erkennbar. Die Leistungsfähigkeit des EAN Kerns ermöglicht somit eine stabile Verarbeitung des Datenstroms.

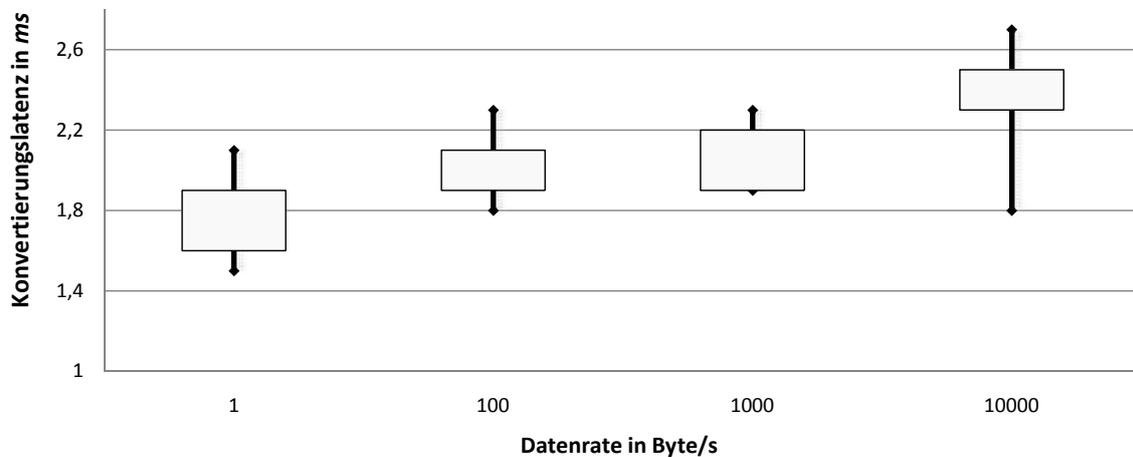


Abbildung 6.17.: Abhängigkeit der Konvertierungszeiten von der Datenübertragungsrate. Detaildarstellung ausgewählter Datenraten mit Durchschnitt- sowie Minimal- und Maximalwerten.

6.3.3. Testscenario III: Energiebedarf

Um die im Kapitel 5.6 durchgeführten Simulationsergebnisse zu evaluieren, wurden mit Hilfe des PLANet Prototypen unterschiedliche Messreihen in Hinsicht auf den Energiebedarf der verfügbaren Funkmodule und des ARM7 Mikrocontrollers durchgeführt. Dazu kam wieder das Oszilloskop zum Einsatz, mit dessen Hilfe die Werte ermittelt wurden. Eine direkte Strommessung ist dabei nicht möglich. Daher wurde ein sehr kleiner Widerstand ($R < 1\Omega$) verwendet und über den Spannungsabfall der Strombedarf über $I = \frac{U}{R}$ bei $U = 3,3V$ errechnet. Da im Oszilloskop der zeitliche Verlauf aufgezeichnet wird, entstehen auf diese Weise präzise Aussagen hinsichtlich der Verbrauchswerte. Für den μC wurden zusätzlich Werte aus der Spezifikation verwendet. Die Messergebnisse konnten im Anschluss in die abstrakten Kostenparameter der SimANet Fallstudien einfließen. Der Leistungsbedarf des gesamten PLANet-Systems sowie der zentralen Komponenten wurde in Diagramm 6.18 dargestellt und anschließend detailliert aufgeschlüsselt.

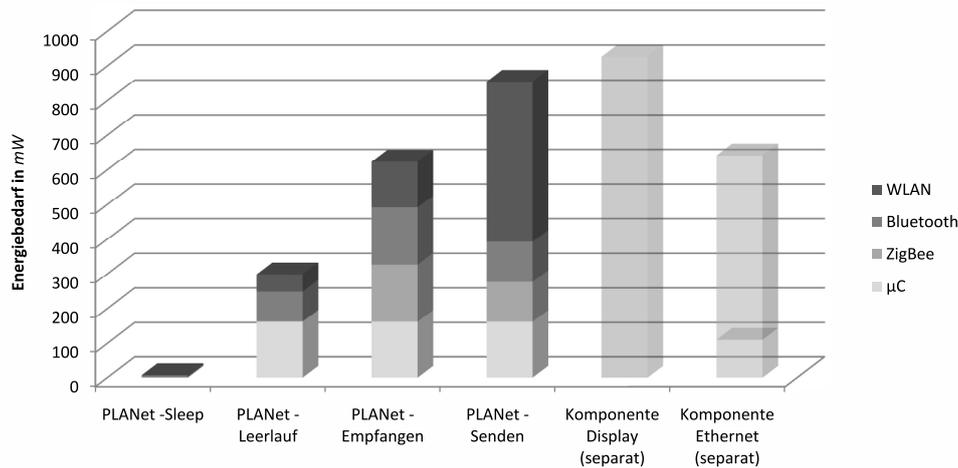


Abbildung 6.18.: Leistungsbedarf der PLANet Einzelkomponenten in den unterschiedlichen Betriebsmodi. Das Display sowie der drahtgebundene Ethernet-Port werden als separate Komponenten halbtransparent aufgeführt, um die Darstellung nicht zu verfälschen. Sie sind für den Betrieb der Plattform nicht erforderlich und dienen lediglich zur Informationsaufbereitung PLANet-interner Systemzustände bzw. als optionale Schnittstelle.

Angaben in mW	PLANet Core (ARM7 μController)	PLANet Core + Ethernet (ohne Display)	PLANet Core + Ethernet (mit Display)	Funkmodul XBee (IEEE 802.15.4 / ZigBee)	Funkmodul Bluetooth (Free2Move / IEEE 802.15.1)	Funkmodul WLAN (WiFi / IEEE 802.11.G)	PLANet komplett 3 x Funkmodul + Ethernet (ohne Display)	PLANet komplett 3 x Funkmodul + Ethernet (mit Display)
Leerlauf	≤162	≤266	≤1192	≤0.03	≤86	≤50	≤402	≤1328
Sleep	≤0.16	≤110	≤1036	≤0.03	≤5.28	≤0.01	≤116	≤1042
Empfangen (RX)	—	≤804	≤1730	≤165	≤165	≤132	≤1266	≤2192
Senden (TX)	—	≤804	≤1730	≤115	≤116	≤462	≤1497	≤2423

Tabelle 6.1.: Messungen des Energiebedarfs von PLANet mit seinen wesentlichen Komponenten. Die Messreihen schließen sowohl unterschiedliche Energiesparmodi als auch die wichtigsten Betriebszustände der Hardware mit ein.

Die Messwerte verdeutlichen die Verteilung des Energiebedarfs innerhalb des Gesamtsystems. Besonders stark steigt der Verbrauch durch das zusätzliche Display an. Auch der Energiebedarf der drahtgebundenen Ethernet-Schnittstelle ist höher als erwartet. Um rund 550 mW unterscheiden sich hier die Messwerte zwischen Idle-Betrieb und aktivem Kommunikationsmodus. Beide Komponenten werden für den Betrieb als EAN-konformes System nicht benötigt und stellen dem Nutzer optionale Schnittstellen während der Testphasen zur Verfügung.

6. Prototyp-Plattform PLANet

	SimANet (abstrakte Kosteneinheiten)	PLANet	Quotient $\frac{\text{PLANet}}{\text{SimANet}}$
$RS_1 \leftrightarrow \text{Bluetooth}$ <i>Basisbetrieb</i>	20	86 mW	4.3
$RS_2 \leftrightarrow \text{WLAN}$ <i>Basisbetrieb</i>	20	50 mW	2.5
$RS_1 \leftrightarrow \text{Bluetooth}$ <i>Senden mit Maximalleistung</i>	8	116 mW	14.5
$RS_2 \leftrightarrow \text{WLAN}$ <i>Senden mit Maximalleistung</i>	50	462 mW	9.24
$RS_1 \leftrightarrow \text{Bluetooth}$ <i>max. Übertragungslatenz</i>	20	40 ms	2
$RS_2 \leftrightarrow \text{WLAN}$ <i>max. Übertragungslatenz</i>	10	10 ms	1

Tabelle 6.2.: Gegenüberstellung der abstrakt definierten Kostenwerte in der SimANet Umgebung und dem gemessenen Werten auf Basis von PLANet.

	SimANet $\frac{RS_2}{RS_1}$	PLANet $\frac{\text{WLAN}}{\text{Bluetooth}}$
<i>Basiskosten</i>	1.0	0.58
<i>Senden bei Maximalleistung</i>	6.25	3.98
<i>Übertragungsdauer 1 Paket 10 Byte</i>	0.05	0.25
<i>Protokollkonvertierung bidirektional</i>	1	1

Tabelle 6.3.: Gegenüberstellung der Relationen zwischen den beiden Funkstandards RS_1 und RS_2 aus der Simulation sowie den gemessenen Werten für Bluetooth und WLAN auf Basis von PLANet.

Die Relationen zwischen Simulation und PLANet Messungen wurden in Tabelle 6.2 und 6.3 dargestellt. Der praktische Testaufbau entspricht dem ersten Messszenario aus Abschnitt 6.3.1 mit der Kommunikation zweier PLANet Knoten über einen Brückenknoten (siehe Abbildung 6.10).

Dabei ist zu beachten, dass die Übertragungseigenschaften beider Standards unter Realbedingungen von den definierten Charakteristika in SimANet abweichen. So beträgt die Kommunikationsreichweite $T_{RS,2}$ in der Simulation die doppelte Reichweite des Funkstandards RS_1 ($T_{RS,2} = 2 \cdot T_{RS,1}$). In der praktischen Umsetzung ist die Reichweite von WLAN meist deutlich

höher als die doppelte Bluetooth-Reichweite.

Interessant sind hierbei die analysierten Basisbetriebskosten beider Funkstandards ohne Last. Während diese in der Simulation gleich angenommen wurden, zeigen die Messungen einen geringeren Energieverbrauch für das WLAN Modul. Hier wird die Abhängigkeit der Messergebnisse von der jeweiligen Hardware deutlich. Je nach gewähltem Hersteller und Modell können hier die Werte stark variieren. Der Energiebedarf für die Datenübertragung mittels RS_1 wurde in der Simulationsumgebung wiederum etwas zu hoch definiert. Die Relation $RS_1 \leftrightarrow Bluetooth$ weist einen Faktor von 14.5 auf, wohingegen bei $RS_2 \leftrightarrow WLAN$ nur ein Faktor von 9.24 entsteht.

Bei der Wahl zwischen Übertragungsstandard RS_1 oder RS_2 errechnet sich in den Simulationsszenarien ein abweichender Energiebedarf von Faktor 6.25. Im Gegensatz dazu beträgt der Niveauunterschied bei den PLANet Messungen lediglich 3.98. Somit wurden die Simulationen unter etwas härteren Rahmenbedingungen durchgeführt als in der realen Messumgebung beobachtet. Hier benötigt WLAN für die Übertragung weniger als die 6.25fache Sendeleistung von Bluetooth. Die gemessenen Übertragungslatenzen bei der Bluetooth-Kommunikation sind deutlich höher als in der Simulation erwartet. Auch wenn die Ergebnisse stark von den eingesetzten Hardwaremodulen abhängen, ist diese Latenz als kritisch zu werten. Der Zeitfaktor für den Funkstandardwechsel ergibt auch unter Realbedingungen gleichbleibend stabile Werte in beide Konvertierungsrichtungen.

Der gewählte ARM7 Mikrocontroller hat aufgrund seiner Leistungsparameter einen relativ hohen Energiebedarf. Dies war jedoch bei der Konzeption der Prototyp-Plattform bekannt und wurde zu Gunsten großzügiger Leistungsreserven in Kauf genommen. Da PLANet nicht primär für den Batteriebetrieb entwickelt wurde, spielt der Verbrauch durch den μC nur eine untergeordnete Rolle. Für die Realisierung eines vollintegrierten EAN Systems kann an dieser Stelle ein entsprechend klein dimensionierter Low-Power μC zum Einsatz kommen.

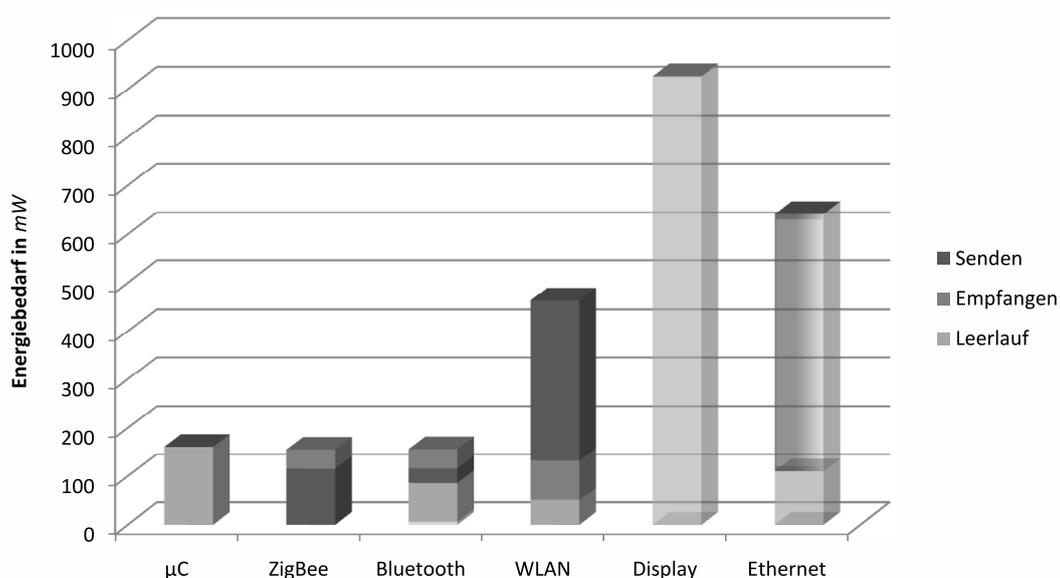


Abbildung 6.19.: Aufteilung des Leistungsbedarfs der einzelnen PLANet Komponenten in Abhängigkeit von den unterschiedlichen Betriebsmodi.

Der benötigte Energieanteil für die einzelnen Funkmodule bewegt sich innerhalb der erwarteten Parameter. Alle drei Funkadapter verbrauchen im Idle-Zustand aufsummiert weniger als 150 mW. Interessant ist jedoch der Unterschied zwischen den Modulen verschiedener Funkstandards. Besonders deutlich zeigt sich dieser zwischen den Nahbereichs-Modulen und dem WLAN-Interface. Während die Messwerte für den Datenempfang nah beieinander liegen, benötigt das Senden mittels WLAN die 4fache Energie. 6.19 verdeutlicht nochmals die Relationen jeder Komponente in den einzelnen Betriebsmodi. Der Sleep-Modus wurde im Diagramm aufgrund minimaler Verbrauchswerte vernachlässigt.

Trotz leichter Abweichungen in den Relationen konnten die durchgeführten Messungen alle Simulationsergebnisse stützen, da die Rahmenbedingungen in SimANet strenger als unter realen Bedingungen definiert wurden.

6.4. Zusammenfassung und offene Punkte

In diesem Kapitel wurde eine prototypische Umsetzung des vorgestellten Konzeptes für eine funkstandardübergreifende Kommunikation in mobilen Ad Hoc Netzwerken präsentiert. Die Umsetzung verlief dabei in mehreren Schritten. Zunächst wurden im Rahmen einer experimentellen Umsetzung erste Funktionstests durchgeführt. Basierend auf den Ergebnissen konnten anschließend Entscheidungen für der Auswahl zentraler Hardwarekomponenten getroffen werden. Im Zuge der Weiterentwicklungen entstand eine integrierte, modulare EAN Kommunikationsplattform. Auf Basis eines Mikrocontrollers sowie eines geeigneten Betriebssystem wurden mehrere Funkmodule eingebunden und die funktionalen Aspekte des vorgestellten Konzeptes implementiert.

Es folgten praktische Messungen hinsichtlich des Energiebedarfs der Plattform sowie die Evaluierung der konzeptionellen Grundlagen unter Realbedingungen. Die Ergebnisse zeigen einen stabilen EAN Kern für die Protokollkonvertierung sowie sehr geringe Latenzen im Zuge der Paketverarbeitung. Zusätzlich konnten die Messergebnisse mit den Simulationsergebnissen aus Kapitel 5 verglichen werden. Dabei wurden die abstrakt definierten SimANet Kostenparameter analysiert und entsprechend in Relation gesetzt.

Es konnte gezeigt werden, dass eine effiziente Umsetzung des Konzeptes für unterschiedliche Anwendungsszenarien möglich ist. Wichtige Kriterien sind dabei sowohl die nötigen Hardwareressourcen als auch die Leistungsfähigkeit der zentralen Komponenten, welche einen störungsfreien Protokollkonvertierungsprozess sicherstellen müssen. Eine weiterführende Analyse komplexer Kommunikationsabläufe und die vollständige Integration handelsüblicher, mobiler Endgeräte stellt einen wichtigen Punkte für weiterführende Arbeiten innerhalb des Projektes dar. Konkrete Themen und Ansatzpunkte können wie folgt definiert werden:

- Einbindung weiterer drahtloser Kommunikationstechnologien, bspw. GPRS/UTMS
- Testszenarien in hochskalierten Topologien mit erweiterten Cross-Layer Routingansätzen

6.4. Zusammenfassung und offene Punkte

- Erweiterung der unterstützten Protokolle innerhalb der LCU für die Kommunikation auf höheren Protokollebenen
- Firmwareaktualisierung via SD-Card / Netzwerk
- Verwendung von *Smart Antenna Systemen* sowie *Wake-Up-Receiver Technologien* [76]

7. Zusammenfassung und Ausblick

In der vorliegenden Dissertation wurde ein Konzept für die Integration unterschiedlicher drahtloser Kommunikationstechnologien in eine heterogene Netzwerktopologie vorgestellt. Der Fokus lag dabei auf Anwendungsszenarien im Bereich mobiler Ad Hoc Netzwerke und deren dynamischen Strukturen. Der untersuchte Lösungsansatz basiert auf der Kopplung handelsüblicher, standardisierter Funkmodule und ermöglicht so eine funkstandardübergreifende Kommunikation. Eine zentrale Steuerinstanz auf einer hardwarenahen Ebene erlaubt es dabei, je nach Anforderungen an den jeweiligen Kommunikationskanal, einen optimalen Funkstandard für die Übertragung zu wählen. Somit können die jeweiligen Vorteile der unterschiedlichen Übertragungstechnologien genutzt und deren Nachteile gleichzeitig umgangen werden.

Die Arbeit befasste sich dabei zunächst mit grundlegenden Modellen auf dem Bereich drahtgebundener sowie drahtloser Netzwerke. Hierbei ist ein Verständnis für das ISO/OSI Referenzmodell sowie dessen Abbildung auf den TCP/IP Protokollstack mit den jeweiligen Funktionskapselungen in den Protokollebenen von besonderer Bedeutung.

Desweiteren standen aktuelle, drahtlose Kommunikationsstandards und deren Anwendungsbereiche im Fokus der Betrachtungen. Neben den unterschiedlichen Charakteristika der einzelnen, standardisierten Funkstandards zeigte sich ein gemeinsamer Nachteil bei nahezu allen Technologien. Durch die starke Anpassung an die jeweiligen Anwendungsgebiete sind die Übertragungsstandards sehr unflexibel, wodurch dynamische Anpassungen der Übertragungseigenschaften nur in stark begrenztem Umfang möglich sind. Dies erklärt auch die aktuellen Forderungen der Industrie nach immer neuen Funkstandards für einzelne, spezifische Einsatzgebiete. Dem entgegen steht somit die Idee des in dieser Arbeit vorgestellten Integrationskonzeptes für unterschiedliche Kommunikationstechnologien in eine gekoppelte Übertragungseinheit.

Neben der Thematik aktueller Funkstandards wurde auch auf verschiedene Aspekte der Selbstorganisation eingegangen, die Lösungsansätze für die dezentrale Organisation großer, mobiler Netzwerktopologien aufzeigen soll. Primäres Ziel der Selbstorganisation ist die Senkung des administrativen Aufwands für mobile Ad Hoc Netztopologien und die gleichzeitige Erhöhung der Robustheit gegenüber äußeren Störeinflüssen. Eine Möglichkeit besteht dabei in die Schaffung eines hybriden Netzwerkmanagements aus klassischen, hierarchischen Organisationsformen auf höherer Ebene und selbstorganisierender Strukturen auf den unteren Ebenen der Topologie.

Ein weiterer Themenschwerpunkt des Grundlagenkapitels vermittelt einen Einstieg in die Problematik der Schnittstellensynthese eingebetteter Systeme und deren Teilkomponenten. Ziel soll dabei sein, logisch inkompatible Protokolle bzw. Schnittstellen durch einen angepassten Adapterblock zu koppeln und eine Kommunikation zu ermöglichen. Die Generierung der Konvertereinheit wird dabei durch einen automatisierten Workflow übernommen. Das vorgestellte IFB Konzept bildet eine zentrale Grundlage für den im Rahmen dieser Dissertation vorgestellten

7. Zusammenfassung und Ausblick

Lösungsansatz.

Nachdem allgemeine Prinzipien vermittelt wurden, repräsentiert das Kapitel Stand der Technik einen Überblick über aktuelle Forschungsarbeiten auf dem Gebiet mobiler Ad Hoc und Sensornetzwerke. Dies umfasst Ansätze für eine Optimierung der Topologie sowie angepasste Routingstrategien.

Im Anschluss werden drei für diese Arbeit relevante Forschungsprojekte vorgestellt. Software Defined Radio thematisiert hierbei die technische Umsetzung einer Echtzeitkonvertierung unterschiedlicher Übertragungsprotokolle auf Basis von Softwaremodulen. Ein zweites Forschungsprojekt trägt den Titel Cognitive Radio und befasst sich mit der Interferenzproblematik auf den unteren Netzwerkebenen. Es erweitert den SDR Ansatz um eine Frequenzband-Überwachung und ggf. um einen Katalog aus Richtlinien bzw. Restriktionen hinsichtlich der Ressourcenallokation. Das Ziel ist eine adaptive, intelligente Auf- und Zuteilung des verfügbaren Frequenzspektrums. Abstrakt kann diese Zuteilung auch auf unterschiedliche Funkstandards abgebildet werden.

Ambient Networking, ein dritter Forschungsbereich, verfolgt ähnliche Ziele wie das vorgestellte Kommunikationskonzept. Mobile Endgeräte sollen dabei alle zur Verfügung stehenden Kommunikationsschnittstellen auf Basis spezieller Dienste für die Datenübertragung nutzen.

Resultierend aus den offengelegten Schwachstellen von Ambient Networking und Cognitive Radio Ansätzen entstand ein neues Konzept für die funkstandardübergreifende Kommunikation auf einer hardwarenahen Ebene, welches Softwareapplikationen eine einfache Nutzung unterschiedlicher drahtloser Funkstandards ermöglicht. Durch eine Funktionskapselung auf Hardwareebene wird das Hostsystem im Gegensatz zu anderen vorgestellten Lösungsansätzen nicht zusätzlich belastet.

Die einzelnen Resultate dieser Arbeit sollen in den nachfolgenden Abschnitten unter den genannten Aspekten detailliert reflektiert werden.

7.1. Ergebnisse dieser Arbeit

Die Ergebnisse gliedern sich in drei wesentlichen Bereichen auf - Konzept, Simulation und praktische Umsetzung. Die Grundlage bildet das entworfene Konzept "Embedded Ambient Networking" mit den jeweiligen Teilkomponenten und deren funktionaler Interaktion.

7.1.1. Konzept

Das EAN Konzept basiert auf dem IFB Ansatz zur Kopplung logisch inkompatibler Schnittstellen. Dementsprechend wurden in dieser Arbeit zunächst die Hintergründe und Konzeptgrundlagen der Interface Synthese und des Interface Blocks rekapituliert. Im Zuge einer softwareseitigen Umsetzung innerhalb eines Mikrocontrollers wurde sich gegen eine Kapselung in Protocol

Handler und Sequence Handler als dedizierte Komponenten entschieden

Die konzeptionellen Änderungen wurden unter der xIFB Erweiterung zusammengefasst. Auf Basis der eingeführten Interface Handler wird nun die Verarbeitung dynamischer Protokolle erleichtert und der Betrieb in drahtlosen Netzwerkstrukturen realisiert. Das EAN Konzept beinhaltet außerdem eine weitere, zentrale Komponente für das Verbindungsmanagement. Die Link Controller Unit überwacht dabei alle aktiven Verbindungskanäle und repräsentiert die Entscheidungsinstanz für die zielgerichtete Weiterleitung eingehender Datenpakete.

Neben der Protokollkonvertierung und dem Verbindungsmanagement richtet sich eine dritte Kernfrage an die Wegfindung in der geschaffenen, heterogenen Netzwerktopologie. Das im Rahmen dieser Arbeit entwickelte EBCR Verfahren repräsentiert einen hybriden Cross-Layer Ansatz. Es kombiniert dabei grundlegende Routing-Anforderungen aus dem Bereich mobiler Ad Hoc Netzwerke mit kooperativen Strategien aus dem Anwendungsgebiet drahtloser Sensornetzwerke. Das Verfahren ermöglicht das Ausbalancieren topologieweiter Unterschiede in den Energieniveau einzelner Netzwerkknoten. Zur Senkung der Netzlast integriert EBCR zusätzliche Caching-Mechanismen, wodurch gleichzeitig die Netzwerklatenzen gesenkt werden können.

7.1.2. SimANet

Um die funktionalen Aspekte des EAN Ansatzes grundlegend evaluieren zu können, war die Simulation unterschiedlicher Topologien und Anwendungsszenarien zwingend erforderlich. Die am Markt verfügbaren Simulationsumgebungen wiesen dabei mehrere zentrale Probleme auf. Daraus resultierend folgte die Konzeption und Umsetzung einer plattformunabhängigen, funkstandardübergreifenden Simulationsplattform - SimANet. Diese ermöglicht es dem Nutzer, Funkstandards auf Basis abstrakter Kostenwerte zu modellieren und an die jeweiligen Netzwerkknoten zu binden. SimANet erlaubt dabei die simultane Coexistenz mehrerer autonomer Kommunikationsschnittstellen in einem Knoten. Mit Hilfe des Simulationsframeworks wurden die funktionalen Aspekte des vorgestellten EAN Konzeptes untersucht und die Leistungsfähigkeit des Lösungsansatzes verdeutlicht. Die Simulationsszenarien integrieren auch das entwickelte Routingverfahren und ermöglichen somit eine umfassende Leistungsanalyse. Durch die Konzeption zusätzlicher Parallelisierungsmechanismen können nun auch hochskalierte Testreihen auf leistungsfähigen Rechenclustern, wie z.B. dem CHiC der TU Chemnitz, realisiert werden.

Sowohl die These einer Verbesserung der Erreichbarkeit als auch die Steigerung der Topologielebensdauer konnten dadurch gestützt werden. Diese Ergebnisse bildeten gleichzeitig die Grundlage für die Entwicklung einer Prototypplattform, welche die Umsetzbarkeit des EAN Ansatzes in einem experimentellen Rahmen und unter realen Bedingungen belegen soll.

7.1.3. PLANet

PLANet repräsentiert eine Plattform für Ambient Networking und stellt prototypisch eine integrierte Kommunikationsplattform gemäß dem vorgestellten EAN Konzeptes dar. Aufbauend auf dieser Arbeit können nun auch weiterführende, komplexe Anwendungsszenarien praktisch umgesetzt werden.

Auf der Hardwareplattform wurden im Zuge unterschiedlicher Messreihen zentrale Funktionen eines EAN Knotens umgesetzt. Die durchgeführten Testszenarien umfassten hierbei anfallende Latenzzeiten während einer Multihop-Kommunikation mit ein- und mehrfachen Konvertierungsprozessen auf den Kommunikationspfaden. Weiterführende Analysen des Energiebedarfs stellen außerdem die praktischen Messwerte und die abstrakten Kostenwerte innerhalb der SimANet Umgebung in Relation.

7.2. Ausblick

Die vorliegende Dissertation bietet eine Vielzahl möglicher Ansatzpunkte für weiterführende Arbeiten. Ein primäres Ziel von Weiterentwicklungen stellt die ganzheitliche Umsetzung der LCU Einheit und somit des vollständigen Verbindungsmanagements dar. Im Rahmen der praktischen Arbeit wurde diese zunächst als statische Komponente realisiert. Mit dem Fokus auf ein intelligentes, funkstandardübergreifendes Verbindungs- und Topologiemanagement existiert somit eine weitreichende Grundlage für unterschiedliche wissenschaftliche Arbeiten.

Auf der Basis des PLANet Systems sind Untersuchungen mit größeren und komplexeren Anwendungsszenarien anzustreben. Der umgesetzte Experimentalaufbau umfasste eine Topologie mit drei EAN Netzwerkknoten und drei zusätzlichen, statischen Endgeräten. Qualitative Aussagen hinsichtlich des Multihop-Kommunikationsverhaltens sind daher nur einschränkt möglich. Um die Kompatibilität zu handelsüblichen Endgeräten zu verbessern und folglich die Integration existierender Netzinfrastrukturen vollständig zu gewährleisten, sind an dieser Stelle noch umfassende Arbeiten durchzuführen.

Das EAN Konzept sieht die Integration applikationsseitiger Anforderungen an den Verbindungskanal vor. Der vorgestellte Lösungsansatz basiert auf IP-spezifischen Headerelementen, mit deren Hilfe die relevanten Metainformationen vom Hostsystem in den EAN Verarbeitungskern gelangen. Für die praktische Umsetzung in PLANET wurde auf diese erweiterte Funktionalität bisher verzichtet. Weitere Arbeiten sollten diese Abläufe integrieren und die und den Funktionsumfang unter Realbedingungen analysieren.

In Zusammenhang mit PLANet steht auch die Detektion von Störeinflüssen durch ein erhöhtes Interferenzniveau weiterhin im Fokus der Betrachtungen. Die Möglichkeiten einer aktiven Reaktion des EAN Systems nach den beschriebenen Mustern bieten Inhalt für wissenschaftliche Untersuchungen. Nachdem die These einer zwangsläufigen Verschlechterung des Interferenzverhaltens in der Topologie durch mehrere simultan betriebene Funkstandards diskutiert und

entkräftet wurde, existieren zum jetzigen Zeitpunkt noch keine konkreten Aussagen, ob und ggf. wie hoch das Optimierungspotenzial in der Netzwerktopologie ausfallen kann.

Der vorgestellte PLANet Prototyp verfügt aufgrund begrenzter Ressourcen nur über eine limitierte Anzahl gleichzeitig aktiver Funkstandards. Um eine modulare Erweiterbarkeit für zukünftige, drahtlose Kommunikationstechnologien gewährleisten zu können, ist auch hier der Einsatz dynamischer Rekonfigurierung für unterschiedliche Anwendungsszenarien vorstellbar. Weiterführende Arbeiten auf dem Gebiet der Schnittstellensynthese beschreiben die Möglichkeiten dynamisch rekonfigurierbarer Schnittstellen [91, 92]. Auf dieser Basis könnte der Grad an Flexibilität nochmals deutlich gesteigert werden. Im derzeitigen Entwicklungsstand sind entsprechende Routinen noch nicht vorgesehen.

Sowohl PLANet als auch die Simulationsumgebung SimANet sind in ihrer Funktion nicht ausschließlich auf die Problemstellungen dieser Arbeit beschränkt. Beide Systeme bieten sich für den Einsatz in anderen Projekten auf dem Gebiet drahtloser Ad Hoc und Sensornetzwerke an und sollten für diese genutzt und weiterentwickelt werden.

A. Anhang 1 - SimANet Strukturbeschreibung

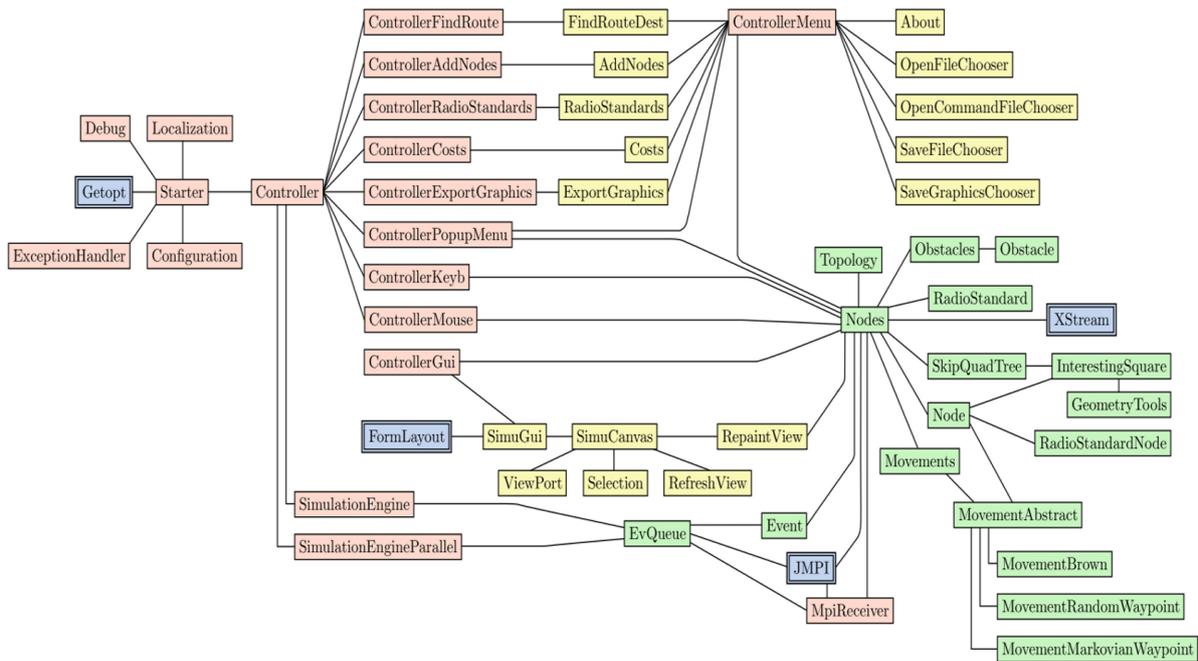


Abbildung A.1.: Darstellung der SimANet Strukturbeschreibung. Die einzelnen Komponenten sind entsprechend ihrer Zuordnung im *Model View Controller* Design Pattern eingefärbt. Doppelt eingeraumte Rechtecke repräsentieren externe Packages und nehmen daher eine Sonderrolle ein. Das MVC-Modell ist deutlich erkennbar. Die Kommunikation zwischen *RepaintView* und der *Nodes* Klasse erfolgt zur Leistungssteigerung auf direktem Wege, wodurch an dieser Stelle das MVC-Design umgangen wird. *RepaintView* aktualisiert dabei in regelmäßigen Abständen die Zeichenfläche und wird als dedizierter Thread ausgeführt.

B. Anhang 2 - SimACon Skriptsprache

Die in dieser Arbeit vorgestellte Steuersprache *SimACon* ermöglicht das Ausführen von internen und externen SimANet-Funktionsaufrufen zu definierten Zeitpunkten. Die so erstellten Skripte erlauben somit einen vollständig automatisierten Simulationsablauf und den Export der gewonnenen Informationen. Die folgenden Abschnitte geben einen Überblick über die Syntax von SimACon sowie über einzelne Befehle und deren Bedeutung.

Syntax

SimACon ist eine zeilenorientierte Skriptsprache. Jede Zeile repräsentiert einen Befehl. Kommentare können durch ein vorgestelltes # eingefügt werden. Jede Befehlszeile ist mit einem Zeitstempel versehen, welcher den Ausführungszeitpunkt festlegt. Das Format wird dabei in Stunde, Minute, Sekunde und Zehntelsekunde getrennt (hh:mm:ss.z). Benötigte Parameter werden in Klammern an den Befehl angefügt und durch Kommata getrennt. Die Trennung zwischen Zeitstempel und Befehl erfolgt durch einen Tabulator.

Befehlssatz

Quit()

- **Funktion:** Beendet SimANet
- **Parameter:** keine

LoadControlFile(Dateiname)

- **Funktion:** Lädt eine weitere SimuCon-Datei (damit lässt sich auch eine periodische Ausführung einer Datei bewirken)
- **Parameter** *Dateiname*: Name der zu ladenden SimuCon-Datei

LoadModelFromFile(Dateiname)

- **Funktion:** Lädt eine gespeicherte Datenstruktur
- **Parameter** *Dateiname*: Name der zu ladenden XML-Datei

WriteModelToFile(Dateiname)

- **Funktion:** Schreibt die aktuelle Datenstruktur in eine Datei
- **Parameter** *Dateiname*: Name der XML-Datei

WriteGraphicsToFile(Dateiname, x1, y1, x2, y2, Auflösung)

- **Funktion:** Exportiert den Bereich (x1, y1) bis (x2, y2) in eine Grafikdatei
- **Parameter** *x1, y1*: Koordinaten der oberen linken Ecke
- **Parameter** *x2, y2*: Koordinaten der unteren rechten Ecke
- **Parameter** *Auflösung*: Gibt an, mit wievielen Pixeln der Radius eines Knotens dargestellt wird. Je größer die Auflösung, desto höher der Detailgrad

CreateNode(x, y, Periode, Energie, zeigeRadius, zeigePfad, zeigeVerbindungen, Bewegung, Funkstandards...)

- **Funktion:** Erstellt einen Knoten an Position (x, y)
- **Parameter** *x, y*: Position, an der der Knoten eingefügt werden soll
- **Parameter** *Periode*: Aktivierungsintervall in ms
- **Parameter** *Energie*: Batteriegröße bzw. Lebensdauer
- **Parameter** *zeigeRadius*: **true** oder **false** – Umkreis der Funkstandards anzeigen
- **Parameter** *zeigePfad*: **true** oder **false** – Anzeigen des aktuellen Bewegungspfades
- **Parameter** *zeigeVerbindungen*: **true** oder **false** – Anzeigen der aktiven Verbindungen zu Nachbarknoten
- **Parameter** *Bewegung*: „Brown“, „Random Waypoint“ oder „Markovian Waypoint“; „null“ bedeutet: keine Bewegung
- **Parameter** *Funkstandards*: Durch Kommata getrennte Liste der implementierten Funkstandards. Liste kann auch leer sein

Beispiel

```
# Erstellen eines Knotens nach 10 Sekunden
# Bewegung: Random Waypoint
# Funkstandards: Zigbee und WLAN
00:00:10.0    CreateNode(1000, 1500, 820, 150000, true, true, true, ↗
                Random Waypoint, Zigbee, WLAN)

# Erstellen eines Knotens nach 1 Minute
# keine Bewegung
# keine Funkstandards
00:01:00.0    CreateNode(-1400, 20, 1322, 50000, true, true, true, null)

# Exportieren der Datenstruktur nach 30 Minuten Simulation
00:30:00.0    WriteModelToFile(model.xml)

# Exportieren eines Snapshots nach 30 Minuten Simulation
# Bereich: (-1500, -1500) bis (1500, 1500)
# Auflösung: Knoten haben Radius von 20 Pixeln
00:30:00.0    WriteGrahicsToFile(shot.png, -1500, -1500, 1500, 1500, 20)

# Unmittelbar danach: Simulator beenden
00:30:00.1    Quit()
```

Abbildung B.1.: Beispielhaftes SimACon Skript mit Funktionsaufrufen aus unterschiedlichen Bereichen (Knoteninteraktion, Datenexport und globale Steuerkommandos).

Eigene Veröffentlichungen

- [1] BORSCHBACH, Markus ; VODEL, Matthias: An Evolutionary Solution for Cross Layer Ad Hoc Connectivity Selection. In: *Proceedings of the ACM & IEEE Fifth Annual Communications Networks and Services Research Conference (CNSR)*. Fredericton, Canada : IEEE Computer Society, May 2007. – ISBN 0–7695–2835–X, S. 381–386
- [2] BORSCHBACH, Markus ; VODEL, Matthias ; LIPPE, Wolfram-M.: Evolutionary Optimized Ad Hoc Connectivity. In: *Proceedings of the IASTED International Conference on Communication and Computer Networks (CCN)*. Lima, Peru : ACTA Press, Oktober 2006. – ISBN 0–88986–630–9, S. no pp. given
- [3] VODEL, Matthias: *Topologieoptimierung in Mobilen Ad Hoc und Sensornetzwerken: Systematische Evaluierung und Vergleich dezentraler Algorithmen zur optimierten Vernetzung selbstorganisierender Systeme*. Vdm Verlag, 2008. – ISBN 978–3639085105
- [4] VODEL, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: A Concept for Radio Standard Spanning Communication in Mobile Ad Hoc Networks / TU Chemnitz. 2007 (CSR-07-04). – Forschungsbericht. – In Chemnitzer Informatik Berichte
- [5] VODEL, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: Energy-Balanced Cooperative Routing Approach for Radio Standard Spanning Mobile Ad Hoc Networks. In: *Proceedings of the 6th International Information and Telecommunication Technologies Symposium (I2TS)*. Brasilia, Brazil : IEEE Region 9, December 2007. – ISBN 978–85–89264–08–X, S. 42–47
- [6] VODEL, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: Performance Analysis of Radio Standard Spanning Communication in Mobile Ad Hoc Networks. In: *Proceedings of the 7th International Symposium on Communications and Information Technologies (ISCIT)*. Sydney, Australia : IEEE Computer Society, October 2007. – ISBN 978–1–4244–09, S. 848–853
- [7] VODEL, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: EBCR - A Routing Approach for Radio Standard Spanning Mobile Ad Hoc Networks. In: *Proceedings of the 4th National Conference on Computing and Information Technology (NCCIT)*. Mahasarakham, Thailand : KMUTNB, May 2008. – ISBN 978–974–19–3296–2, S. 57–58
- [8] VODEL, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: *Funkstandardintegration in Mobilen Ad Hoc Netzwerken*. Wiesbaden, Germany : GI/ITG - Workshop über Selbstorganisierende, Adaptive, Kontextsensitive verteilte Systeme (SAKS), March 2008
- [9] VODEL, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: A Capable, Lightweight Communication Concept by Combining Ambient Network Approaches with Cognitive Radio Aspects. In: *Proceedings of the 17th International Conference on Telecommunications (ICT)*. Doha, Qatar : IEEE Computer Society, April 2010. – ISBN 978–1–4244–5247–7, S. 869–873

Eigene Veröffentlichungen

- [10] VODEL, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: Embedded Ambient Networking - A New, Lightweight Communication Concept. In: *Proceedings of the 9th International Conference on Communications (ICC)*. Cape Town, South Africa : IEEE Computer Society, May 2010. – ISBN 978-1-4244-6826-3, S. no pp. given
- [11] VODEL, Matthias ; SAUPPE, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: A Large Scalable, Distributed Simulation Framework for Ambient Networks. In: *Recent Advances in Information Technology and Security - Journal of Communications (JCM - EI Compendex)* 3 (2008), January, Nr. 7, S. 11-19. – ISSN 1796-2021
- [12] VODEL, Matthias ; SAUPPE, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: A Large Scalable, Distributed Simulation Framework for Radio Standard Spanning Mobile Ad Hoc Networks. In: *Proceedings of the 2008 International Symposium on Computer Science and Computational Technology (ISCST2008)*. Shanghai, China : IEEE Computer Society, December 2008. – ISBN 978-0-7695-3498-5, S. no pp. given
- [13] VODEL, Matthias ; SAUPPE, Matthias ; CASPAR, Mirko ; HARDT, Wolfram: The SimANet Framework. In: *Proceedings of the 1st International Conference on M4D: Mobile Communication Technology For Development (M4D)*. Karlstad, Schweden : Karlstad University, December 2008. – ISBN 978-91-7063-222-8, S. 88-97

Studentische Arbeiten

- [14] LANG, Wolfgang: *Methoden zur modularen Erweiterung funkstandardübergreifender Hardware*. Straße der Nationen 62, 09111 Chemnitz, Deutschland, Technische Universität Chemnitz, Fakultät für Informatik, Professur Technische Informatik : Studienarbeit, September 2009
- [15] LIPPMANN, Mirko: *Schedulingverfahren für verteilte, ressourcenbeschränkte, eingebettete Systeme*. Straße der Nationen 62, 09111 Chemnitz, Deutschland, Technische Universität Chemnitz, Fakultät für Informatik, Professur Technische Informatik, Diplomarbeit, Dezember 2009
- [16] OPP, Daniel: *Evaluierung und Realisierung einer drahtlosen funkstandardübergreifenden Punkt-zu-Punkt-Kommunikation*. Straße der Nationen 62, 09111 Chemnitz, Deutschland, Technische Universität Chemnitz, Fakultät für Informatik, Professur Technische Informatik : Studienarbeit, März 2009
- [17] SAUPPE, Matthias: *SimANet: Implementierung einer Simulationsumgebung für drahtlose Ad-Hoc-Netzwerke*. Straße der Nationen 62, 09111 Chemnitz, Deutschland, Technische Universität Chemnitz, Fakultät für Informatik, Professur Technische Informatik : Studienarbeit, November 2008
- [18] SAUPPE, Matthias: *Java-Anbindung von MPI-Bibliotheken für hochperformante Parallelanwendungen*. Straße der Nationen 62, 09111 Chemnitz, Deutschland, Technische Universität Chemnitz, Fakultät für Informatik, Professur Technische Informatik, Diplomarbeit, September 2009
- [19] SCHMIDT, Gordon: *Konfiguration heterogener drahtloser Sensornetzwerke*. Straße der Nationen 62, 09111 Chemnitz, Deutschland, Technische Universität Chemnitz, Fakultät für Informatik, Professur Technische Informatik, Diplomarbeit, April 2008

Literaturverzeichnis

- [20] AHLGREN, Bengt ; EGGERT, Lars ; OHLMAN, Börje ; SCHIEDER, Andreas: Ambient Networks: Bridging Heterogeneous Network Domains. In: *Proceedings of the 16th Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, IEEE Computer Society, September 2005, S. no pp. given
- [21] AKYILDIZ, Ian F. ; LEE, Won-Yeol ; VURAN, Mehmet C. ; MOHANTY, Shantidev: NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey. In: *Computer Networks Journal (Elsevier)* 50 (2006), September, Nr. 13, S. 2127–2159. – ISSN 1389–1286
- [22] ARSLAN, Huseyin ; CHEN, Zhi N. ; BENEDETTO, Maria-Gabriella D.: *Ultra Wide-band Wireless Communication*. John Wiley & Sons - Wiley-Interscience, 2006. – ISBN 0471715212
- [23] AU-YONG, Jun-Hoong: Comparison of On-Demand Mobile Ad Hoc Network Routing Protocols under On/Off Source Traffic Effect. In: *Proceedings of the IASTED International Conference Networks and Communication Systems*, IASTED, March 2006, S. 527–531
- [24] AUSTRALIA'S ICT RESEARCH CENTRE OF EXCELLENCE: *Castalia - A Simulator for WSNs*. <http://castalia.npc.nicta.com.au/>, 2008. – [Online; abgerufen 01-December-2008]
- [25] BABICH, F. ; COMISSO, M. ; ORLANDO, M. ; MANI'A, L.: Simultaneous Communications in Ad-Hoc Networks Using Smart Antennas in Multipath Environment. In: *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, IEEE Computer Society, November 2006. – ISBN 1–4244–0356–1, S. 1–6
- [26] BACCELLI, Francois ; BASZCZYSZYN, Bartomiej ; MÜHLEHALER, Paul: An Aloha Protocol for Multihop Mobile Wireless Networks. In: *IEEE Transactions on Information Theory* 52 (2006), February, Nr. 2, S. 421–436
- [27] BAJAJ, Lokesh ; TAKAI, Mineo ; AHUJA, Rajat ; TANG, Ken ; BAGRODIA, Rajive ; GERLA, Mario: *GloMoSim: A Scalable Network Simulation Environment / -*. 1999. – Forschungsbericht
- [28] BAKER, Mark ; CARPENTER, Bryan ; FOX, Geoffrey ; KO, Sung H. ; LIM, Sang: MPI-JAVA: An Object-Oriented JAVA Interface to MPI. In: *Proceedings of 11 Workshops Held in Conjunction with the 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing (IPPS/SPDP)*, Springer-Verlag, 1999. – ISBN 3–540–65831–9, S. 748–762
- [29] BALDO, Nicola ; MAGUOLO, Federico ; MIOZZO, Marco ; ROSSI, Michele ; ZORZI, Michele: ns2-MIRACLE: a Modular Framework for Multi-Technology and Cross-Layer Support in Network Simulator 2. In: *Proceedings of the 2nd International Conference*

- on Performance Evaluation Methodologies and Tools (ValueTools)*, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007. – ISBN 978–963–9799–00–4, S. 1–8
- [30] BARRY, Richard: *The FreeRTOS.org Project*. <http://www.freertos.org/>, 2009. – [Online; abgerufen 10-Mai-2009]
- [31] BELLOFIORE, Salvatore ; BALANIS, Consfantine A. ; FOUFZ, Jeffrey ; SPANIAS, Andreas S.: Smart-Antenna Systems for Mobile Communication Networks Part I: Overview and Antenna Design. In: *IEEE Antenna's and Propagation Magazine* 44 (2002), June, Nr. 3, S. 145–154. – ISSN 1045–9243
- [32] BERG, Mark de ; KREVELD, Marc van ; OVERMARS, Mark ; SCHWARZKOPF, Otfried C.: *Computational Geometry*. SV, 2000. – ISBN 3–540–65620–0
- [33] BLAZEVIC, L. ; BUTTYAN, L. ; CAPKUN, S. ; GIORDANO, S. ; HUBAUX, J. ; BOUDEC, J. L.: Self-Organization in Mobile Ad-Hoc Networks: The Approach of Terminodes. In: *IEEE Communications Magazine* (2001), June, S. 166–174
- [34] BLESS, Roland: Using Realistic Internet Topology Data for Large Scale Network Simulations in OMNeT++. In: *Proceedings of the 2nd International OMNeT++ Workshop*, TU Berlin, January 2002, S. no pp. given
- [35] BLUETOOTH SPECIAL INTEREST GROUP: *Specification of the Bluetooth System, Covered Core Package version: 1.2*. <http://www.bluetooth.org>, November 2003
- [36] BORBASH, S.A. ; JENNINGS, E.: Distributed Topology Control Algorithm for Multihop Wireless Networks. In: *Proceedings of the World Congress on Computational Intelligence*, IEEE Computer Society, 2002, S. no pp. given
- [37] BORSCHBACH, Markus: A New Approach for Modelling Local Selective Connectivity and Maintaining Isolated Regions in Ad Hoc Networks. In: *International Journal of Wireless and Optical Communication (IJWOC)* (2006), S. 3(1):23–48
- [38] BORSCHBACH, Markus ; STOLZE, A. ; LIPPE, Wolfram-M.: Optimized Connectivity for Ad Hoc Networks. In: *Proceedings of the 2nd IASTED International Conference on Wireless and Optical Communications (WOC)*, 2002, S. 681–688
- [39] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG: *hyperNET (Universelle Nutzung von Kommunikationsnetzen für künftige Mobilfunkgenerationen)*. <http://www.pt-it.pt-dlr.de/en/1059.php>, 2006. – [Online; abgerufen 15-December-2007]
- [40] BUNDESMINISTERIUM FÜR BILDUNG UND FORSCHUNG: *IBMS²: Integriertes Bandbreiteneffizientes Mobiles Software-Radio System*. <http://www.pt-it.pt-dlr.de/de/1239.php>, 2006. – [Online; abgerufen 15-December-2007]
- [41] BURKHART, Martin ; RICKENBACH, Pascal von ; WATTENHOFER, Roger ; ZOLLINGER, Aaron: Does Topology Control Reduce Interference? In: *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, ACM, 2004. – ISBN 1–58113–849–0, S. 9–19

- [42] CHAKERES, Ian D. ; BELDING-ROYER, Elizabeth M.: AODV Routing Protocol Implementation Design. In: *International Workshop on Wireless Ad Hoc Networking (WWAN)*, IEEE Computer Society, March 2004. – ISBN 0–7695–2087–1, S. 698–703
- [43] CHOWDHURY, K. R. ; AKYILDIZ, I. F.: Cognitive Wireless Mesh Networks for Dynamic Spectrum Access. In: *Journal of Selected Areas in Communications (JSAC)* (2008)
- [44] CHRYSSOMALLIS, Michael: Smart Antennas. In: *IEEE Antennas and Propagation Magazine* 42 (2000), June, Nr. 3, S. 129–136. – ISSN 1045–9243
- [45] CLAUDE ELWOOD SHANNON: Communication in the Presence of Noise. In: *IRE, Nachdruck* 86 (1998), Feb, Nr. 2
- [46] CLAUSEN, T. ; JACQUET, P. ; LAOUITI, A. ; MUHLETHALER, P. ; QAYYUM, A. ; VIENNOT, L.: Optimized Link State Routing Protocol. In: *Proceedings of the International Multitopic Conference (INMIC)*, IEEE Computer Society, 2001, S. no pp. given
- [47] COMER, Douglas: *Internetworking with TCP/IP. Principles, Protocols, and Architectures*. Prentice Hall, 2000. – ISBN 0–13–018380–6
- [48] COOKLEV, T.: *Wireless Communication Standards - A Study of IEEE 802.11, 802.15 and 802.16*. IEEE Computer Society, 2004. – ISBN 0–7381–4066–X
- [49] COOPER, Martin ; GOLDBERG, Marc: Intelligent Antennas: Spatial Division Multiple Access. In: *Proceedings of the IEC Annual Review of Communications*, IEC, 1996, S. 999–1002
- [50] COWIE, J. ; OGIELSKI, A. ; NICOL, D.: *The SSFNet network simulator*. <http://www.ssfnet.org/homePage.html>, 2002. – [Online; abgerufen 15-December-2007]
- [51] DALL, Jesper ; CHRISTENSEN, Michael: Random Geometric Graphs. In: *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics* 66(2) (2002), Nr. 1, S. 016212.1–016212.9. – ISSN 1539–3755
- [52] DARPA: *DARPA - Defense Advanced Research Projects Agency*. <http://www.darpa.mil/>, 2009. – [Online; abgerufen 01-Juni-2009]
- [53] DEFENSE ADVANCED RESEARCH PROJECTS AGENCY (DARPA): *The XG Vision. Request for Comments. Version 2.0. BBN*. http://www.darpa.mil/ato/programs/xg/rfc_vision.pdf, 2003. – [Online; abgerufen 15-December-2007]
- [54] DEPARTMENT OF DEFENSE: *Joint Tactical Radio System - Programmable, Modular Communications System*. <http://www.globalsecurity.org/military/systems/ground/jtrs.htm>, 1997. – [Online; abgerufen 15-December-2007]
- [55] DFG: *Schwerpunktprogramm 1148*. <http://www12.informatik.uni-erlangen.de/spprr/abstract.php>, 2008. – [Online; abgerufen 01-June-2008]
- [56] DRESSLER, F.: Self-Organization in Ad Hoc Networks: Overview and Classification / University of Erlangen, Dept. of Computer Science 7. 2006. – Forschungsbericht
- [57] DUARTE-MELO, E. ; LIU, M.: Analysis of Energy Consumption and Lifetime of Heterogeneous Wireless Sensor Networks. In: *Proceedings of the Global Telecommunications Conference (GLOBECOM)*, IEEE Computer Society, 2002, S. no pp. given

- [58] DUNKEL, Adam: *Contiki - The Operating System for Embedded Smart Objects - the Internet of Things*. <http://www.sics.se/contiki/>, 2009. – [Online; abgerufen 10-Mai-2009]
- [59] EGNITE GMBH: *Ethernut Firmware*. <http://www.ethernut.de/en/firmware/index.html>, 2009. – [Online; abgerufen 10-Mai-2009]
- [60] END COMMUNICATION, SDR Based E.: S. Bhattacharya / DataSoft Corporation. 2005. – Forschungsbericht
- [61] EPPSTEIN, David ; GOODRICH, Michael T. ; SUN, Jonathan Z.: The Skip Quadtree: A Simple Dynamic Data Structure for Multidimensional Data. In: *Proceedings of the 21st Annual Symposium on Computational Geometry (SCG)*, ACM, 2005. – ISBN 1–58113–991–8, S. 296–305
- [62] ERTEL, Richard B. ; CARDIERI, Paulo ; SOWERBY, Kevin W. ; RAPPAPORT, Theodore S. ; REED, Jeffrey H.: Overview of Spatial Channel Models for Antenna Array Communication Systems. In: *IEEE Personal Communications* (1998), February, S. 10–22. – ISSN 1070–9916
- [63] FAHMY, Nader S. ; TODD, Terence D. ; KEZYS, Vytas: Ad Hoc Networks with Smart Antennas Using IEEE 802.11-Based Protocols. In: *Proceedings of the International Conference on Communications (ICC)* Bd. 5, IEEE Computer Society, 2002. – ISBN 0–7803–7400–2, S. 3144–3148
- [64] FEDERAL COMMUNICATION COMMISSION: *Cognitive Radio Technologies Proceeding (CRTP)*. <http://www.fcc.gov/oet/cognitiveradio/>, 2007. – [Online; abgerufen 15-December-2007]
- [65] FEKETE, Sandor P. ; KRÖLLER, Alexander ; FISCHER, Stefan ; PFISTERER, Dennis: Shawn: The Fast, Highly Customizable Sensor Network Simulator. In: *Proceedings of the 4th International Conference on Networked Sensing Systems (INSS 2007)*, 2007, S. no pp. given
- [66] FOERSTER, H. V. ; M.C.YOVITTS ; S.CAMERON: On Self-Organizing Systems and Their Environments. In: *Self-Organizing Systems* (1960), S. 31–50
- [67] GAEDKE, Martin: *Vorlesung Rechnernetze - Version 2009*. <http://vsr.informatik.tu-chemnitz.de/edu/2009/rn/>, 2009
- [68] GAMMA, Erich ; HELM, Richard ; JOHNSON, Ralph E.: *Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software*. Addison Wesley, 2004. – ISBN 3–8273–2199–9
- [69] GANESAN, Deepak ; GOVINDAN, Ramesh ; SHENKER, Scott ; ESTRIN, Deborah: Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks. In: *SIGMOBILE Mobile Computing and Communications* 5 (2001), Nr. 4, S. 11–25. – ISSN 1559–1662
- [70] GERHARZ, M. ; WAAL, C. de ; MARTINI, P. ; JAMES, P.: Strategies for Finding Stable Paths in Mobile Wireless Ad Hoc Networks. In: *Proceedings of the 28th Annual International Conference on Local Computer Networks (LCN)*, IEEE Computer Society, October 2003, S. no pp. given

- [71] GHAVAMI, Mohammad ; MICHAEL, Lachlan ; KOHNO, Ryuji: *Ultra Wideband Signals and Systems in Communication Engineering*. John Wiley & Sons, 2004. – ISBN 978–0–470–86751–8
- [72] GIORDANO, S. ; STOJMENOVIC, I. ; BLAZEVIĆ, L.: *Position Based Routing Algorithms for Ad Hoc Networks: A Taxonomy*. <http://www.site.uottawa.ca/~ivan/routing-survey.pdf>, October 2001. – [Online; abgerufen 02-February-2008]
- [73] GÜNES, Mesut: *Routing und Adressierung in mobilen multi-hop Ad-hoc-Netzen*, Rheinisch-Westfälischen Technischen Hochschule Aachen, Diss., 2004
- [74] GNU: *GNU Radio - The GNU Software Radio*. <http://www.gnu.org/software/gnuradio>, 2004. – [Online; abgerufen 15-December-2007]
- [75] GULTCHEV, Stoytcho ; MOESSNER, Klaus ; THILAKAWARDANA, Duminda ; DODGSON, Terence ; TAFAZOLLI, Rahim ; VADGAMA, Sunil ; TRUELOVE, Stephen: *Evaluation of Software Defined Radio Technology / University of Surrey, Fujitsu Laboratories of Europe Ltd.* 2006. – Forschungsbericht
- [76] GUO, Chunlong ; ZHONG, Lizhi C. ; RABAEY, Jan. M.: Low Power Distributed MAC for Ad Hoc Sensor Radio Networks. In: *Proceedings of the Global Communications Conference (GLOBECOM)*, IEEE Computer Society, November 2001. – ISBN 0–7803–7206–9, S. 2944–2948
- [77] GUSTAFSSON, E. ; JONSSON, A.: Always Best Connected. In: *IEEE Wireless Communications* 10 (2003), Nr. 1, S. 49–55
- [78] HAAS, Zygmunt J. ; PEARLMAN, Marc R.: The Zone Routing Protocol (ZRP) for Ad Hoc Networks. In: *INTERNET-DRAFT, IETF MANET Working Group*, 1997, S. no pp. given
- [79] HAMMERSCHALL, Ulrike: *Verteilte Systeme und Anwendungen*. Pearson Studium, 2005. – ISBN 978–3827370969
- [80] HARDT, Wolfram: *Vorlesung Hardware/Software Codesign II - Version 2009*. <http://www.tu-chemnitz.de/informatik/CE/Lehre/HSCII/SS09/>, 2009
- [81] HARDT, Wolfram ; IHMOR, Stefan: *Schnittstellensynthese - Volume 2 of Wissenschaftliche Schriftenreihe: Eingebettete, selbstorganisierende Systeme*. TUDpress, 2006. – ISBN 3–398863–63–3
- [82] HARDT, Wolfram ; LEHMANN, T. ; VISARIUS, Markus: Towards a Design Methodology Capturing Interface Synthesis. In: *Monjau, Dieter, editor, 4. GI/ITG/GMM Workshop: Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen 1* (2001), February, S. 93–97. ISBN 3–00–007439–2
- [83] HARRINGTON, Adam S. ; HONG, Chin-Gi ; PIAZZA, Anthony L.: *Software Defined Radio - The Revolution of Wireless Communication*. International Engineering Consortium, 2004
- [84] HEDRICK, C.: *Request for Comments (RFC) 1058 - Routing Information Protocol*. <http://tools.ietf.org/rfc/rfc1058.txt>, June 1988. – [Online; abgerufen 15-Mai-2009]

- [85] HEINZELMAN, W.R. ; CHANDRAKASAN, A. ; BALAKRISHNAN, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences* Bd. 2, IEEE Computer Society, January 2000. – ISBN 0–7695–0493–0, S. 10
- [86] HENDERSON, Thomas R. ; ROY, Sumit ; FLOYD, Sally ; RILEY, George F.: ns-3 Project Goals. In: *Proceeding from the 2006 Workshop on NS-2: the IP Network Simulator (WNS2)*, ACM, 2006. – ISBN 1–59593–508–8, S. 13
- [87] HO, Ming-Ju ; STÜBER, Gordon L. ; AUSTIN, Mark D.: Performance of Switched-Beam Smart Antennas for Cellular Radio Systems. In: *IEEE Transactions on Vehicular Technology* 47 (1998), February, Nr. 1, S. 10–19. – ISSN 0018–9545
- [88] HSU, Chih-Shun ; TSENG, Yu-Chee ; SHEU, Jang-Ping: An Efficient Reliable Broadcasting Protocol for Wireless Mobile Ad Hoc Networks. In: *Ad Hoc Networks* 5 (2007), Nr. 3, S. 299–312. – ISSN 1570–8705
- [89] IANNONE, L. ; KHALILI, R. ; SALAMATIAN, K. ; FDIDA, S.: Cross-Layer Routing in Wireless Mesh Networks. In: *1st International Symposium on Wireless Communication Systems*, IEEE Computer Society, September 2004. – ISBN 0–7803–8472–5, S. 319–323
- [90] IHMOR, Stefan: *Modeling and Automated Synthesis of Reconfigurable Interfaces*. Paderborn, Universität Paderborn, Heinz Nixdorf Institut, Entwurf Paralleler Systeme, Diss., January 2006
- [91] IHMOR, Stefan ; FLADE, Marcel ; HARDT, Wolfram: *Rekonfigurierbare Schnittstellen - volume 1 of Wissenschaftliche Schriftenreihe: Eingebettete, selbstorganisierende Systeme*. TUDpress, 2005. – ISBN 3–398863–37–4
- [92] IHMOR, Stefan ; HARDT, Wolfram: Runtime Reconfigurable Interfaces - The RTR-IFB Approach. In: *Proceedings of the 11th Reconfigurable Architectures Workshop (RAW)*, IEEE Computer Society, April 2004. – ISBN 0–7695–2132–0, S. 136–136
- [93] II, Joseph M.: *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*, Royal Institute of Technology (KTH) Stockholm, Sweden, Diss., 2000
- [94] INTANAGONWIWAT, Chalermek ; GOVINDAN, Ramesh ; ESTRIN, Deborah: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, ACM, 2000. – ISBN 1–58113–197–6, S. 56–67
- [95] J. MITOLA II AND Z. ZVONAR: *Software Radio Technologies*. IEEE Computer Society, 2001
- [96] JONDRAL, Friedrich K.: Software-Defined Radio-Basics and Evolution to Cognitive Radio. In: *EURASIP Journal on Wireless Communications and Networking* 3 (2005), April, S. 275–283
- [97] JOSEPH MITOLA II: *Software Radio Architecture: Object-Oriented Approaches to Wireless System*. John Wiley & Sons, 2000
- [98] KAMPMANN, Markus ; VORWERK, Marc ; GROTHUES, Ingo ; LEON, María del Mar Ivars d. ; KLEIS, Michael ; SCHUETZ, Simon ; SCHMID, Stefan ; AGUERO, Ramon ;

- CHOQUE, Johnny: An Experimental Study of Novel Ambient Network Concepts. In: *Proceedings of the 13th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, IEEE Communications Society, September 2005, S. no pp. given
- [99] KARP, Brad ; KUNG, H. T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In: *Proceedings of the Mobile Computing and Networking (Mobicom)*, IEEE Computer Society, August 2000, S. 243–254
- [100] KENNEDY, Joseph ; SULLIVAN, Mark C.: Direction Finding and "Smart Antennas" Using Software Radio Architectures. In: *IEEE Communications Magazine* (1995), May, S. 62–68. – ISSN 0163–6804
- [101] KIM, Young-An ; NAKAGAWA, Masao: R-ALOHA Protocol for Spread Spectrum Inter-Vehicle Communication Network using Head Spacing Information. In: *Proceedings of the Intelligent Vehicles Symposium*, IEEE Computer Society, September 1996. – ISBN 0–7803–3652–6, S. no pp. given
- [102] KLEINROCK, L. ; TOBAGI, F.A.: Packet switching in radio channels, part I - carrier sense multiple access modes and their throughput-delay characteristics. In: *IEEE Transactions on Communications COM 23* (1975), S. 1400–1416
- [103] KOCHHAL, Manish ; SCHWIEBERT, Loren ; GUPTA, Sandeep: Role-based Hierarchical Self Organization for Wireless Ad Hoc Sensor Networks. In: *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, ACM, 2003. – ISBN 1–58113–764–8, S. 98–107
- [104] KOUDOURIDIS, G. P. ; KARLSSON, P. ; LUNDSJÖ, J. ; BRIA, A. ; BERG, M. ; JORGU-SESKI, L. ; MEAGO, F. ; AGÜERO, R. ; SACHS, J. ; KARIMI, R.: Multi-Radio Access in Ambient Networks. In: *Everest Workshop: Trends in Radio Resource Management (2nd Edition)* (2005), November
- [105] KOUDOURIDIS, Georgios P. ; KARIMI, H. R. ; DIMOU, Konstantinos: Switched Multi-Radio Transmission Diversity in Future Access Networks. In: *Proceedings of the 62th Semiannual Vehicular Technology Conference (VTC)*, IEEE Computer Society, September 2005, S. no pp. given
- [106] KULIK, Joanna ; HEINZELMAN, Wendi R. ; BALAKRISHNAN, Hari: Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks. In: *Wireless Networks, Vol. 8* (2002), S. 169–185
- [107] LEE, W. Y. ; AKYILDIZ, I. F.: Optimal Spectrum Sensing Framework for Cognitive Radio Networks. In: *Transaction on Wireless Communications* (2008)
- [108] LI, N. ; HOU, J. ; SHA, L.: Design and Analysis of an MST-based Topology Control Algorithm. In: *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies Bd. 3*, IEEE Computer Society, 2003, S. 1702–1712
- [109] LI, Ning ; JENNIFER, C. H.: Topology Control in Heterogeneous Wireless Networks: Problems and Solutions. In: *Proceedings of the 21th International Conference of the Computer and Communications Societies (INFOCOM)*, IEEE Computer Society, March 2004. – ISBN 0–7803–8355–9, S. 243

- [110] LIU, J. ; LI, B.: Distributed Topology Control in Wireless Sensor Networks with Asymmetric Links. In: *Proceedings of the Global Telecommunications Conference (GLOBECOM), Wireless Communications Symposium*, IEEE Computer Society, 2003, S. 1257–1262
- [111] LOPEZ, A.R.: Performance Predictions for Cellular Switched-Beam Intelligent Antenna Systems. In: *IEEE Communications Magazine* 34 (1996), October, Nr. 10, S. 152–154. – ISSN 0163–6804
- [112] LUNDSJÖ, Johan ; AGÜERO, Ramón ; ALEXANDRI, Eftychia ; BERGGREN, Fredrik ; CEDERVALL, Catarina ; DIMOU, Konstantinos ; GEBERT, Jens ; JENNEN, Ralf ; JORGUESKI, Ljupco ; KARIMI, Reza ; MEAGO, Francesco ; TANG, Haitao ; VERONESI, Riccardo: A Multi-Radio Access Architecture for Ambient Networking. In: *Proceedings of the 14th IST Mobile & Wireless Communications Summit 2005*, European Information Society Technologies, June 2005, S. no pp. given
- [113] MAGNUSSON, Per ; BERGGREN, Fredrik ; KARLA, Ingo ; LITJENS, Remco ; MEAGO, Francesco ; TANG, Haitao ; VERONESI, Riccardo: Multi-Radio Resource Management for Communication Networks beyond 3G. In: *Proceedings of the 62nd Semiannual Vehicular Technology Conference (VTC)*, IEEE Computer Society, September 2005. – ISBN 0–7803–9152–7, S. 1653–1657
- [114] MAHLMANN, Peter ; SCHINDELHAUER, Christian: *Peer-to-Peer Netzwerke: Algorithmen Und Methoden*. Springer, 2007. – ISBN 978–3540339915
- [115] MALKIN, G.: *Request for Comments (RFC) 1983 - Internet Users' Glossary*. <http://www.ietf.org/rfc/rfc1983.txt>, August 1996. – [Online; abgerufen 15-Mai-2009]
- [116] MANTIS GROUP: *Multimodal System for NeTworks of In-Situ Wireless Sensors*. <http://mantis.cs.colorado.edu/>, 2009. – [Online; abgerufen 10-Mai-2009]
- [117] MEISEL, Andre ; VISARIUS, Markus ; HARDT, Wolfram ; IHMOR, Stefan: Self-Reconfiguration of Communication Interfaces. In: *Proceedings of the 15th International Workshop on Rapid System Prototyping (RSP)*, IEEE Computer Society, June 2004. – ISBN 0–7695–2159–2, S. 144–150
- [118] MORIN, Steven ; KOREN, Israel ; KRISHNA, C. M.: jMPI: Implementing the Message Passing Standard in Java. In: *Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS)*, IEEE Computer Society, 2002. – ISBN 0–7695–1573–8, S. 191
- [119] MPI FORUM: *The Message Passing Interface (MPI) Standard*. <http://www-unix.mcs.anl.gov/mpi>, 2007. – [Online; abgerufen 15-December-2007]
- [120] MPI FORUM: *MPICH Home Page*. <http://www-unix.mcs.anl.gov/mpi/mpich1>, 2007. – [Online; abgerufen 15-December-2007]
- [121] MÖSSENBOCK, Hanspeter: *Objektorientierte Programmierung*. Springer-Verlag, 1993. – ISBN 3–540–55690–7

- [122] MUELLER, Stephen ; TSANG, Rose P. ; GHOSAL, Dipak: *Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges*. Springer Verlag, 2004. – ISBN 978–3–540–21945–3
- [123] NAOUMOV, Valeri ; GROSS, Thomas: Simulation of Large Ad Hoc Networks. In: *Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*, ACM, 2003. – ISBN 1–58113–766–4, S. 50–57
- [124] NASIPURI, A. ; DAS, S.: Demand Multipath Routing for Mobile Ad Hoc Networks. In: *Proceedings of the 8th Annual International Conference on Computer Communications and Networks (ICCCN)*, IEEE Computer Society, October 1999, S. 64–70
- [125] NASIPURI, A. ; LI, K. ; SAPPIDI, U. R.: Power Consumption and Throughput in Mobile Ad Hoc Networks using Directional Antennas. In: *Proceedings of the International Conference on Computer Communication and Networks (ICCCN)*, IEEE Computer Society, October 2002. – ISBN 0–7803–7553–X, S. 620–626
- [126] NEUGEBAUER, Mario: *Energieeffiziente Anpassung des Arbeitszyklus in drahtlosen Sensornetzen*, Technische Universität Dresden, Diss., 2006
- [127] NI, S.-Y. ; TSENG, Y.-C. ; CHEN, Y.-S. ; SHEU, J.-P.: The Broadcast Storm Problem in a Mobile Ad Hoc Network. In: *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, ACM, 1999. – ISBN 1–58113–142–9, S. 151–162
- [128] NIEBERT, Norbert ; FLINCK, H. ; HANCOCK, R. ; KARL, Holger ; PREHOFER, Christian: Ambient Networks - Research for Communication Networks Beyond 3G. In: *IST Mobile and Wireless Communications Summit 2004* (2004)
- [129] NIEBERT, Norbert ; SCHIEDER, Andreas ; ABRAMOWICZ, Hendrik ; MALMGREN, Göran ; SACHS, Joachim ; HORNA, Uwe ; PREHOFER, Christian ; KARL, Holger: Ambient Networks: An Architecture for Communication Networks Beyond 3G. In: *IEEE Wireless Communications* 11 (2004), Nr. 2, S. 14–22. – ISSN 1536–1284
- [130] NYQUIST, Harry: Certain Topics in Telegraph Transmission Theory. In: *Trans. Amer. Inst. Elect. Eng., Nachdruck* 90 (2002), February, Nr. 2, S. 617–644
- [131] OPPERMAN, Ian ; HAMALAINEN, Matti ; IINATTI, Jari: *UWB - Theory and Applications*. John Wiley & Sons, 2004. – ISBN 978–0–470–86917–8
- [132] PAQUEREAU, Laurent ; HELVIK, Bjarne E.: A Module-Based Wireless Node for NS-2. In: *Proceeding from the 2006 Workshop on NS-2: the IP Network Simulator (WNS2)*, ACM, 2006. – ISBN 1–59593–508–8, S. 4
- [133] PARK, Vincent D. ; CORSON, M. S.: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In: *Proceedings of the Conference on Computer Communications (INFOCOM)* Bd. 3, IEEE Computer Society, April 1997, S. 1405–1413
- [134] PERKINS, Charles E. ; BHAGWAT, Pravin: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In: *Proceedings of the SIGCOMM 94*, ACM, 1994. – ISBN 0–89791 –682–4, S. 234–244

- [135] PLUMMER, David C.: *Request for Comments (RFC) 826 - An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*. <http://tools.ietf.org/rfc/rfc826.txt>, November 1982. – [Online; abgerufen 15-Mai-2009]
- [136] POSTEL, J.: *Request for Comments (RFC) 768 - User Datagram Protocol*. <http://www.ietf.org/rfc/rfc768.txt>, August 1980. – [Online; abgerufen 15-Mai-2009]
- [137] POSTEL, J.: *Request for Comments (RFC) 791 - Internet Protocol*. <http://www.ietf.org/rfc/rfc791.txt>, September 1981. – [Online; abgerufen 15-Mai-2009]
- [138] POSTEL, J.: *Request for Comments (RFC) 792 - Internet Control Message Protocol*. <http://www.ietf.org/rfc/rfc792.txt>, September 1981. – [Online; abgerufen 15-Mai-2009]
- [139] POSTEL, J.: *Request for Comments (RFC) 793 - Transmission Control Protocol*. <http://www.ietf.org/rfc/rfc793.txt>, September 1981. – [Online; abgerufen 15-Mai-2009]
- [140] POUTTU, Ari ; ROMPPAINEN, Harri ; TAPIO, Visa ; BRÄYSY, Timo ; LEPPÄNEN, Pentti ; TUUKKANEN, Topi: Finnish Software Radio Program and Demonstrator. In: *IEEE Military Communications Conference 2004 (MILCOM 2004)* 3 (2004), October, S. 1371–1376. ISBN 0–7803–8847–X
- [141] PREE, Wolfgang: *Design Patterns for Object-Oriented Software Development*. Addison Wesley, ACM, 1994. – ISBN 978–0201422948
- [142] QINETIQ: An Evaluation of Software Defined Radio - An Overview / QinetiQ. 2006 (PUB0603673). – Forschungsbericht
- [143] QINETIQ: Cognitive Radio Technology A Study for Ofcom - Summary Report / QinetiQ. 2007 (QINETIQ/06/00420 Issue 1.1). – Forschungsbericht
- [144] RAJENDRAN, V. ; OBRACZKA, K. ; GARCIA-LUNA-ACEVES, J. J.: Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. In: *Proceedings of the First International Conference on Embedded Networked Sensor Systems*, ACM, November 2003, S. 181–192
- [145] RAMAKRISHNAN, K. ; FLOYD, S. ; BLACK, D.: *Request for Comments (RFC) 3168 - The Addition of Explicit Congestion Notification (ECN) to IP*. <http://tools.ietf.org/rfc/rfc3168.txt>, September 2001. – [Online; abgerufen 15-Mai-2009]
- [146] RAMANATHAN, R. ; ROSALES-HAIN, R.: Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment. In: *Proceedings of the 19th International Conference of the Computer and Communications Societies (INFOCOM)*, IEEE Computer Society, 2002, S. 404–413
- [147] RAZAVILAR, Javad ; RASHID-FARROKHI, Farrokh ; LIU, K. J. R.: Software Radio Architecture with Smart Antennas: A Tutorial on Algorithms and Complexity. In: *IEEE Journal on Selected Areas in Communications* 17 (1999), April, Nr. 4, S. 662–676. – ISSN 0733–8716

- [148] RECH, Jörg: *Wireless LANs. 802.11-WLAN-Technologie und praktische Umsetzung im Detail. 2. Auflage.* Heise Verlag, 2006. – ISBN 3–936931–29–1
- [149] REKHTER, Y. ; WATSON, T.J. ; LI, T.: *Request for Comments (RFC) 1771 - A Border Gateway Protocol 4 (BGP-4).* <http://tools.ietf.org/rfc/rfc1771.txt>, March 1995. – [Online; abgerufen 15-Mai-2009]
- [150] RHIEMEIER, Arnd-Ragnar: Modular Software-Defined Radio. In: *EURASIP J. Wirel. Commun. Netw.* 5 (2005), Nr. 3, S. 333–342. – ISSN 1687–1472
- [151] RILEY, Dr. G. ; PARK, Alfred: *PDNS - Parallel/Distributed NS.* <http://www.cc.gatech.edu/computing/compass/pdns/index.html>, 2004. – [Online; abgerufen 15-December-2007]
- [152] RILEY, G.F. ; FUJIMOTO, R.M. ; AMMAR, M.H.: A Generic Framework for Parallelization of Network Simulations. In: *Proceedings of the 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (1999)*, S. 128–135
- [153] ROYER, E.M. ; TOH, C.-K.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. In: *IEEE Personal Communications* 6 (1999), April, Nr. 2, S. 46–55
- [154] S. BHATTACHARYA: SDR Based End-to-End Communication / DataSoft Corporation. 2005. – Forschungsbericht
- [155] SAHA, A.K. ; JOHNSON, D.B.: Routing Improvement using Directional Antennas in Mobile Ad Hoc Networks. In: *Proceedings of the Global Telecommunications Conference (GLOBECOM) Bd. 5*, IEEE Computer Society, November 2004. – ISBN 0–7803–8794–5, S. 2902–2908
- [156] SAHA, Siddharth: *TeNS - The Enhanced Network Simulator.* <http://www.cse.iitk.ac.in/users/braman/tens/>, 2008. – [Online; abgerufen 01-December-2008]
- [157] SANTI, P.: Topology Control in Wireless Ad Hoc and Sensor Networks. In: *ACM Computing Surveys, Vol 37, Number 2* (2005)
- [158] SCALABLE NETWORK TECHNOLOGIES (SNT): *QualNet.* <http://www.qualnet.com>, 2008. – [Online; abgerufen 01-December-2008]
- [159] SCHERFF, Jürgen: *Grundkurs Computernetze - Eine kompakte Einführung in die Rechnerkommunikation - Anschaulich, verständlich, praxisnah.* Friedr. Vieweg & Sohn, 2006. – ISBN 978–3–528–05902–6
- [160] SCHÜRMAN, Bernd: *Grundlagen der Rechnerkommunikation.* Friedr. Vieweg & Sohn Verlag, 2004. – ISBN 3–528–15562–0
- [161] SDR FORUM: *SDR Forum.* <http://www.sdrforum.org/>, 2007. – [Online; abgerufen 15-December-2007]
- [162] SEKERCIOGLU, Y. A. ; VARGA, Andras ; EGAN, Gregory K.: Parallel Simulation made easy with OMNeT++. In: *Proceedings of the 15th European Simulation Symposium (ESS2003)*, The Society for Computer Simulation International, October 2003, S. no pp. given

- [163] SHEIKH, Khurram ; GESBERT, David ; GORE, Dhananjay ; PAULRAJ, Arogyaswami: Smart Antennas for Broadband Wireless Access Networks. In: *IEEE Communications Magazine* (1999), November, S. 100–105. – ISSN 0163–6804
- [164] SIWIAK, Kazimierz ; MCKEOWN, Debra: *Ultra-Wideband Radio Technology*. John Wiley & Sons, 2004. – ISBN 978–0–470–85931–5
- [165] SLIJEPCEVIC, S. ; POTKONJAK, M.: Power Efficient Organization of Wireless Sensor Networks. In: *Proceedings of the International Conference on Communications (ICC)*, IEEE Computer Society, June 2001, S. 472–476
- [166] SOCIETY, IEEE C.: *Part 16: Standard for Local and Metropolitan Area Networks*. <http://standards.ieee.org/getieee802/download/802.16-2004.pdf>, 2004. – [Online; abgerufen 15-Mai-2008]
- [167] SOCIETY, IEEE C.: *802.11 LAN/MAN Wireless LANS*. <http://standards.ieee.org/getieee802/download/802.11-2007.pdf>, 2007. – [Online; abgerufen 15-Mai-2008]
- [168] SOCIETY, IEEE C.: *Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)*. <http://standards.ieee.org/getieee802/download/802.15.1-2005.pdf>, 2007
- [169] SOCIETY, IEEE C.: *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>, 2007
- [170] SOHRABI, K. ; GAO, V. Ailawadhi J. ; ; POTTIE, G.: Protocols for Self Organization of a Wireless Sensor Network. In: *IEEE Personal Communication Magazine* 7 (2000), October, S. 16–27
- [171] SOHRABI, Katayoun ; GAO, Jay ; AILAWADHI, Vishal ; POTTIE, Gregory J.: Protocols for Self-Organisation of a Wireless Sensor Network. In: *IEEE Personal Communications* (2000), S. 16–27
- [172] SRIVASTAVA, G. ; BOUSTEAD, P. ; CHICHARO, J.F.: A Comparison of Topology Control Algorithms for Ad-Hoc Networks. In: *Proceedings of the Australian Telecommunications, Networks and Applications Conference (ATNAC)*, Springer Verlag, 2003, S. no pp. given
- [173] STAPLE, G. ; WERBACH, K.: The End of Spectrum Scarcity. In: *IEEE Spectrum* 41 (2004), Nr. 3, S. 48–52
- [174] STEVENS, Richard: *TCP/IP Illustrated. Volume 1. The Protocols*. Addison-Wesley, 1994. – ISBN 0–2016–3346–9
- [175] STEVENS, Richard: *TCP/IP Illustrated. Volume 2. The Implementation*. Addison-Wesley, 1994. – ISBN 0–2016–3354–X
- [176] STIFTINGER, Martin: *The Bucket Quadtree*. <http://www.iue.tuwien.ac.at/phd/halama/node112.html>, 1994. – [Online; abgerufen 15-December-2007]

- [177] SUNDRESH, Sameer ; KIM, Wooyoung ; AGHA, Gul: SENS: A Sensor, Environment and Network Simulator. In: *Proceedings of the 37th annual symposium on Simulation (ANSS)*, IEEE Computer Society, 2004. – ISBN 0–7695–2110–X, S. 221
- [178] TANENBAUM, Andrew S.: *Computernetzwerke - 4. Auflage*. Pearson Studium, 2003. – ISBN 3–8273–7046–9
- [179] TIOMKIN, Yuri: *TNKernel real-time system*. <http://www.tnkernel.com>, 2009. – [Online; abgerufen 10-Mai-2009]
- [180] TOH, C.: *Topology Control in Wireless Sensor Networks*. J. Wiley & Sons, 2005
- [181] TSENG, Y.-C. ; LI, Y.-F. ; CHANG, Y.-C.: On Route Lifetime in Multihop Mobile Ad Hoc Networks. In: *IEEE Transactions on Mobile Computing 2* (2003), October, Nr. 4
- [182] TSENG, Yu-Chee ; NI, Sze-Yao ; SHIH, En-Yu: Adaptive Approaches to Relieving Broadcast Storms in a Wireless Multihop Mobile Ad Hoc Network. In: *Transactions on Computers 52* (2003), Nr. 5, S. 545–557. – ISSN 0018–9340
- [183] TSOULOS, G. V.: Smart Antennas For Mobile Communications Systems. In: *IEEE Electronics & Communications Engineering Journal 11* (1999), April, Nr. 2, S. 84–94. – ISSN 0954–0695
- [184] TUUKKANEN, Topi ; POUTTU, Ari ; LEPPÄNEN, Pentti: Finnish Software Radio Programme / Finnish Defense Forces Telecommunication Laboratory and Centre for Wireless Communications. 2003. – Forschungsbericht. – ISBN 951–42–7187–4
- [185] UNIVERSITY OF CALIFORNIA BERKELEY: *TinyOS - The Open-Source Operating System Designed for Wireless, Embedded Sensor Networks*. <http://www.tinyos.net>, 2009. – [Online; abgerufen 10-Mai-2009]
- [186] VARGA, A.: *The OMNeT++ Discrete Event Simulation System*. <http://www.omnetpp.org>, 1999. – [Online; abgerufen 15-December-2007]
- [187] VENUTURUMILLI, Abhinay ; MINAI, Ali: Obtaining Robust Wireless Sensor Networks through Self-Organization of Heterogenous Connectivity. In: *InterJournal Complex Systems* (2006), S. 1823 ff.
- [188] WANG, Lei ; SHU, Yantai ; DONG, Miao ; YANG, Oliver W.: Adaptive Multipath Source Routing in Ad Hoc Networks. In: *Proceedings of the International Conference on Communications* Bd. 3, IEEE Computer Society, June 2001. – ISBN 0–7803–7097–1, S. 867–871
- [189] WANG, S.Y. ; LIN, Y.B.: NCTUns Network Simulation and Emulation for Wireless Resource Management. In: *Wiley Wireless Communications and Mobile Computing 5* (2005), December, Nr. 8, S. 899–916
- [190] WELTER, Heinrich: *Heterogene Netze - Einführung in Standardarchitekturen, Protokolle, Verwaltung und Sicherheit*. Addison-Wesley, 1993. – ISBN 3–89319–499–1
- [191] WIESLER, Anne: *Parametergesteuertes Software Radio für Mobilfunksysteme, Band 4*. Forschungsberichte aus dem Institut für Nachrichtentechnik der Universität Karlsruhe (T.H.), 2001. ISSN 1433–3821

- [192] WOLF, T. D. ; HOLVOET, T.: Emergence Versus Self-Organisation: Different Concepts but Promising When Combined. In: *S. Brueckner et al. (Eds.): ESOA 2004 (2005)*, S. 1–15
- [193] YANG, Guangsong ; XIAO, Mingbo ; CHEN, Huihuang ; YAO, Yan: A Novel Cross-Layer Routing Scheme of Ad Hoc Networks with Multi-Rate Mechanism. In: *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing* Bd. 2, IEEE Computer Society, September 2005, S. 701–704
- [194] YE, F. ; CHEN, A. ; LU, S.W. ; ZHANG, L.: A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks. In: *Proceedings of the 10th International Conference on Computer Communications and Networks*, 2001, S. 304–309
- [195] ZHANG, Yan ; LUO, Jijun ; HU, Honglin: *Wireless Mesh Networking: Architectures, Protocols and Standards*. Auerbach Pubn, 2006. – ISBN 978–0849373992
- [196] ZIGBEE ALLIANCE: *Zigbee Specification*. http://www.zigbee.org/en/spec_download/zigbee_downloads.asp, 2007

Symbolverzeichnis

AIM	<u>A</u> bstract <u>I</u> nterface <u>M</u> anagement / <u>I</u> nstruction <u>S</u> ets
ARP	<u>A</u> ddress <u>R</u> esolution <u>P</u> rotocol
ARPANET	<u>A</u> dvanced <u>R</u> esearch <u>P</u> rojects <u>A</u> gency <u>N</u> etwork
BGP	<u>B</u> order <u>G</u> ateway <u>P</u> rotocol
CAN	<u>C</u> ontroller <u>A</u> rea <u>N</u> etwork
CDMA	<u>C</u> ode <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
CHiC	<u>C</u> hemnitz <u>H</u> igh Performance <u>L</u> inux <u>C</u> luster
CRC	<u>C</u> yclic <u>R</u> edundancy <u>C</u> heck
CSMA/CA	<u>C</u> arrier <u>S</u> ense <u>M</u> ultiple <u>A</u> ccess / <u>C</u> ollision <u>A</u> voidance
CSMA/CD	<u>C</u> arrier <u>S</u> ense <u>M</u> ultiple <u>A</u> ccess / <u>C</u> ollision <u>D</u> etection
CU	<u>C</u> ontrol <u>U</u> nit
DAB	<u>D</u> igital <u>A</u> udio <u>B</u> roadcasting
DMA	<u>D</u> irect <u>M</u> emory <u>A</u> ccess
DREAM	<u>D</u> istance <u>R</u> outing <u>E</u> ffect <u>A</u> lgorithm for <u>M</u> obility
DS-CDMA	<u>D</u> irect <u>S</u> equence <u>C</u> ode <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
DSCP	<u>D</u> ifferentiated <u>S</u> ervices <u>C</u> ode <u>P</u> oint
DSL	<u>D</u> igital <u>S</u> ubscriber <u>L</u> ine
DVB-T	<u>D</u> igital <u>V</u> ideo <u>B</u> roadcasting- <u>T</u> errestrial
EAN	<u>E</u> mbedded <u>A</u> mbient <u>N</u> etworking
EBCR	<u>E</u> nergy <u>B</u> alancing <u>C</u> ooperative <u>R</u> outing
ECN	<u>E</u> xplicit <u>C</u> ongestion <u>N</u> otification
EEPROM	<u>E</u> lectrically <u>E</u> rasable <u>P</u> rogrammable <u>R</u> ead <u>O</u> nly <u>M</u> emory
EGP	<u>E</u> xterior <u>G</u> ateway <u>P</u> rotocol
EOSC	<u>E</u> volutionary <u>O</u> ptimization <u>S</u> elective <u>C</u> onnectivity
FDDI	<u>F</u> iber <u>D</u> istributed <u>D</u> ata <u>I</u> nterface
FDMA	<u>F</u> requency <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
FDT	<u>F</u> ormal <u>D</u> escription <u>T</u> echniques
FHSS	<u>F</u> requency <u>H</u> opping <u>S</u> pread <u>S</u> pectrum
FIFO	<u>F</u> irst <u>I</u> n <u>F</u> irst <u>O</u> ut

Literaturverzeichnis

Flash-ROM	Flash Read Only Memory
FPGA	Field Programmable Gate Array
FSM	Final State Machines
FSRP	Finnish Software Radio Program
FTP	File Transfer Protocol
GloMoSim	Global Mobile Information Systems Simulation Library
GPSR	Greedy Perimeter Stateless Routing for Wireless Networks
GSM	Global System for Mobile Communications
HAL	Hardware Abstraction Layer
IBMS2	Integrierte Bandbreiteneffiziente Mobile Software-Radio System
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IFB	Interface Block
IFS	Interface Synthesis
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
IH	Interface Handler
IMAP	Internet Message Access Protocol
IP	(Embedded Systems) Intellectual Property
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISM	Industrial, Scientific and Medical Band
ISO	Internationalen Standardisierungs Organisation
ISP	Intelligent Signal Processing
JNI	Java Native Interface
JTAG	Joint Test Action Group
JTRS	Joint Tactical Radio System
L2CAP	Logical Link Control and Adaption Protocol
LAR	Location-Aided Routing
LCU	Link Controller Unit
LEACH	Low-Energy Adaptive Clustering Hierarchy
LLC	Logical Link Control
LMC	Link Management Cache

LMP	<u>L</u> ink <u>M</u> anager <u>P</u> rotocol
LMST	<u>L</u> ocal <u>M</u> inimum <u>S</u> panning <u>T</u> ree
LRU	<u>L</u> east <u>R</u> ecently <u>U</u> sed
MAC	<u>M</u> edia <u>A</u> ccess <u>C</u> ontrol
MAN	<u>M</u> etropolitan <u>A</u> rea <u>N</u> etwork
MANET	<u>M</u> obile <u>A</u> d <u>H</u> oc <u>N</u> etzwerke
MB-OFDM	<u>M</u> ultiband <u>O</u> rthogonal <u>F</u> requency <u>D</u> ivision <u>M</u> ultiplexing
MIMO	<u>M</u> ultiple <u>I</u> nput <u>M</u> ultiple <u>O</u> utput
MPI	<u>M</u> essage <u>P</u> assing <u>I</u> nterface
MW-Node	<u>M</u> odule-based <u>W</u> ireless <u>N</u> ode
NCTUns	<u>N</u> ational <u>C</u> hiao <u>T</u> ung <u>U</u> niversity <u>N</u> etwork <u>S</u> imulator
NFC	<u>N</u> earfield <u>C</u> ommunication
ns2-MIRACLE	.	<u>M</u> ulti- <u>I</u> nte <u>R</u> f <u>A</u> ce <u>C</u> ross <u>L</u> ayer <u>E</u> xtension
OFDM	<u>O</u> rthogonal <u>F</u> requency <u>D</u> ivision <u>M</u> ultiplex
OMNet++	<u>O</u> pen <u>S</u> ource <u>M</u> odule-based <u>N</u> etwork <u>S</u> imulator in C++
OSI	<u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection
OSPF	<u>O</u> pen <u>S</u> horte <u>s</u> t <u>P</u> ath <u>F</u> irst
PDNS	<u>P</u> arallel/ <u>D</u> istributed <u>N</u> S
PH	<u>P</u> rotocol <u>H</u> andler
PLANet	<u>P</u> latform for <u>A</u> mbient <u>N</u> etworking
QoS	<u>Q</u> ualitative of <u>S</u> ervice
RAM	<u>R</u> andom <u>A</u> ccess <u>M</u> emory
RFIS	<u>R</u> adio <u>F</u> requency <u>I</u> dentification
RIP	<u>R</u> outing <u>I</u> nformation <u>P</u> rotocol
RTS/CTS	<u>R</u> eady <u>T</u> o <u>S</u> end / <u>C</u> lear <u>T</u> o <u>S</u> end
S-MAC	<u>S</u> ensor- <u>M</u> edia <u>A</u> ccess <u>C</u> ontrol
SCA	<u>S</u> oftware <u>C</u> ommunications <u>A</u> rchitecture
SDL	<u>S</u> pecication and <u>D</u> escription <u>L</u> anguage
SDMA	<u>S</u> pace <u>D</u> ivision <u>M</u> ultiple <u>A</u> ccess
SDP	<u>S</u> ervice <u>D</u> iscovery <u>P</u> rotocol
SDR	<u>S</u> oftware <u>D</u> efined <u>R</u> adio
SENS	<u>A</u> <u>S</u> ensor, <u>E</u> nvironment and <u>N</u> etwork <u>S</u> imulator
SH	<u>S</u> equence <u>H</u> andler

Literaturverzeichnis

SimANet	Simulation Platform for Ambient Networking
SMTP	Simple Mail Transfer Protocol
SPI	Serial Peripheral Interface
SPIN	Sensor Protocols for Information via Negotiation
SSFNet	Scalable Simulation Framework Network Models
SSH	Secure Shell
T-MAC	Timeout-Media Access Control
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TeNS	The Enhanced Network Simulator
TORA	Temporally Ordered Routing Algorithm
TOS	Type of Service
TRAMA	Traffic-Adaptive Medium Access Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UWB	Ultra Wide Band
VoIP	Voice over IP
WEP	Wired Equivalent Privacy
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
WPAN	Wireless Personal Area Networks
WSN	Wireless Sensor Networks
WUSB	Wireless Universal Serial Bus
xIFB	Extended Interface Block
XML	Extendable Markup Language
ZRP	Zone Routing Protocol

Thesen

Die vorliegende Arbeit stellt mehrere zentrale Thesen auf, welche sowohl mittels Simulationen als auch auf Basis praktischer Messungen in einer entwickelten Prototypplattform evaluiert wurden.

These I:

Die Kopplung handelsüblicher Funkmodule in ein integriertes Kommunikationssystem mit unterschiedlichen, standardisierten Kommunikationsstandards ist effizient umsetzbar.

These II:

Durch die heterogene Topologie kann die Erreichbarkeit aller Elemente in der Netzwerkstruktur gesteigert werden.

These III:

Trotz des simultanen Betriebes mehrerer Funkmodule kann der topologieweit gemittelte Energiebedarf reduziert werden, sofern eine hinreichende Netzlast vorausgesetzt wird.

These IV:

Durch die geeignete Einordnung in den TCP/IP Stack kann das vorgestellte, funkstandardübergreifende Kommunikationssystem ohne Adaptionen an den Host-Systemen genutzt werden.

These V:

Auf Basis eines angepassten Routingverfahrens können die konzeptionellen Vorteile in der geschaffenen, heterogenen Netztopologie genutzt werden.